Efficient and Robust Weighted Least-Squares Cell-Average Gradient Construction Methods For the Simulation of Scramjet Flows

Jeffery A. White, *NASA Langley Research Center, Hampton, VA*
Robert A. Baurle, *NASA Langley Research Center, Hampton, VA*
Hioraki Nishikawa, National Institute of Aerospace, Hampton, VA

ABSTRACT

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 designed to minimize the number of cells used to augment/stabilize the least-squares 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. 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 7.8 turbulent flow of air through a scramjet engine flowpath is computed and compared with experimental data to demonstrate the robustness, accuracy and convergence behavior of the preferred gradient method for a realistic 3-D geometry on a non-hex-dominant grid.

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 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, in an effort to still address the aforementioned computational resource constraints though a multiregion approach, while leveraging the ability of the unstructured-grid approach to reduce the total time to obtain a flow solution, the incorporation of an unstructured-grid spatially-elliptic flow solver capability into the VULCAN-CFD multiregion domain decomposition framework was initiated. This was accomplished 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 1) improve the left hand side as compared to the approximations used by the original LU-SGS and matrix-free SGS schemes using an operator overloaded linearization of a 1st-order advection scheme and a hand linearization of a thin-layer Navier-Stokes diffusion scheme and 2) to couple the partitions during the linear solve subiterative process. Additional point implicit schemes have also been added such as Symmetric Gauss-Seidel (SGS) and Symmetric Successive Over-Relaxation (SSOR). 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. The majority of these modifications were recently described in detail in [9].

The current work seeks to demonstrate how the least-squares cell-average gradient stencil affects the stability and robustness of the unstructured solver on a realistic 3-D Scramjet Inlet geometry. A previous work, [9], described the face neighbor (*fn1*) and the face neighbor of face neighbors (*fn2*) least-squares stencils. A more recent work, [10], described a vertex/node neighbor (*nn*) stencil and the 3-D extension of a new stencil construction method (*symF*), designed to address the short comings of the *fn2* and *nn* stencils [11], that selectively augments the *fn1* stencil with cells from the *nn* stencil. Moreover, in Ref. [10], a robust, stencil-consistent, multidimensional cell-average gradient limiter procedure was described and investigated via the computation of canonical problems that were designed to compare the accuracy, efficiency, robustness and convergence behavior of the cell-average gradient weighted linear least-squares *fn1*, *fn2*, *nn* and the *symF* stencil methodologies. In the current work, numerical experiments are performed for hypersonic turbulent flow over a backward facing step and a turbulent flat plate using the *fn1*, *fn2*, *nn* and *symF* stencil methods and compared to examine robustness and precision. Finally, the inlet portion of the HIFiRE 7 REST scramjet engine shock **t**unnel experiment geometry is computed and compared against experimental data [12].

Results and Discussion

1. 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 [13].

While no cell-average gradient method has been found to be accurate for all arbitrary polygons, with some caveats [14], the weighted linear least-squares method has been found to be a robust method when computing cell-average gradients [14-16] for node-centered and cell-centered 2nd-order finite-volume schemes. Therefore, based on the results in the literature [13-17], the weighted linear 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

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

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:

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

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 overdetermined WLSQ system (2) can be solved in various ways. We chose to use QR factorization via the Householder transformation [18], 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 A by the QR factorization. The solution can be expressed in the following form:

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

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.

1.1. *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 [16,17]. These instabilities can be alleviated by augmenting the stencil to reduce or eliminate the bias [16,17].

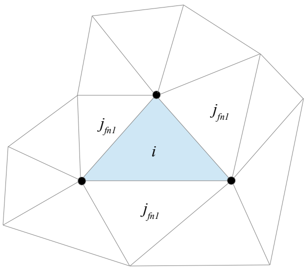


Fig. 1: The *fn1* stencil for computing the WLSQ cell-average gradient on a triangular grid.

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 [16,17]. 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 as shown in Fig. 3. 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 on nonhexahedral grids also can potentially have many more cells in the *nn* stencil than in the *fn2* stencil thereby requiring significantly more storage and operations to compute the gradient.

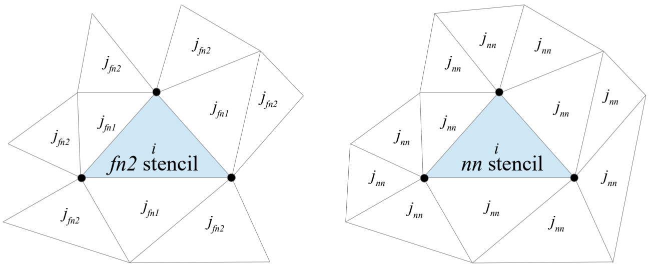


Fig. 2: The *fn2* and *nn* stencils for computing the WLSQ cell-average gradient on a triangular grid.

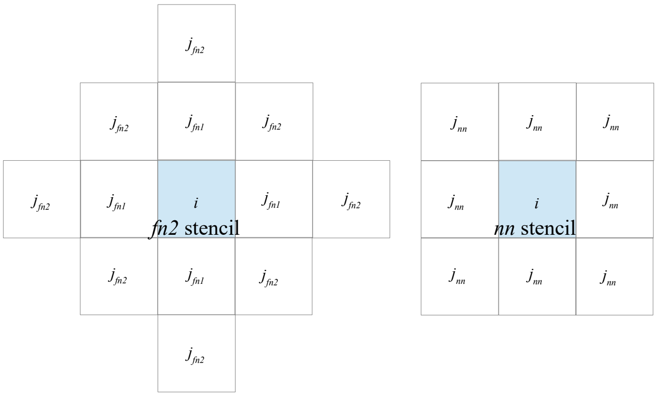
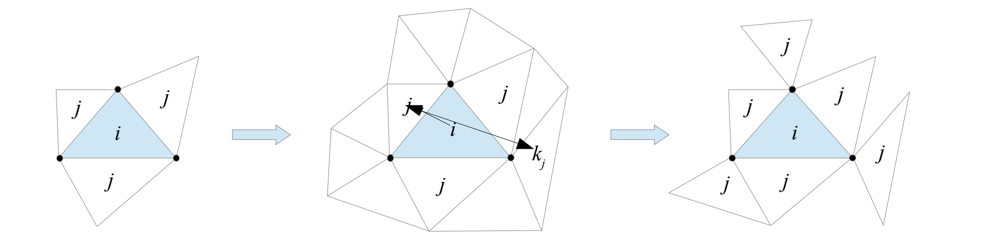

j_{fn2}
j_{fn2} j_{fn1} j_{fn2}
j_{fn2} j_{fn1} i j_{fn1} j_{fn2}
fn2 stencil
j_{fn2} j_{fn1} j_{fn2}
j_{fn2}
j_{nn} j_{nn} j_{nn}
j_{nn} i j_{nn}
nn stencil
j_{nn} j_{nn} j_{nn}

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

It has been shown in Refs. [16,17] that the set of neighbors, $\{g_j\}$, that define the WSLQ stencil, affects the stability of finite-volume solvers. Reference [16] 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. Reference [17] 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 Eq. (6), especially in 3-D [19] as well as decreasing the accuracy of the gradients [11]. Recently, Nishikawa [11] explored ways to construct a gradient stencil that achieves robustness and accuracy of the *fn2* and *nn* stencils with a smaller stencil. In Ref. [11], Nishikawa proposed two augmentation methods that resulted in a robust, efficient and accurate stencil for the 2-D problems examined. 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, the reader is referred to Ref. [11]. 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.

1.2. *Construction of the symmetric augmented fn1 stencil,* i.e., the *sym* stencil => $\{g_j\}_{sym}$

The construction of the symmetric augmented *fn1* stencil, $\{g_i\}_{sym}$, described in Ref. [11], 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 k_{sym} cells, $\{k_i\}_{sym}$, until the stencil, $\{g_i\}_{sym}$, is formed from the union of $\{g_i\}_{fn1}$ *and* $\{k_j\}_{sym}$, As shown in Fig. 4, this process is repeated for each 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 $\{g_i\}_{fn1}$.



***fn1* stencil, $\{g_j\}_{fn1}$**

Find the cells, k_j, symmetric to the *fn1* stencil cells, from the unused cells in, $\{g_j\}_{nn}$

augmented stencil, $\{g_j\}_{sym}$

Fig. 4: The process for constructing the *sym* augmented *fn1* stencil for a grid of triangular cells.

1.3. *Construction of the symmetric F-decreasing augmented fn1 stencil,* i.e., the *symF* stencil => $\{g_j\}_{sym}$

The symmetric *F*-decreasing augmented *fn1* stencil, $\{g_i\}_{symF}$, is constructed using an augmentation of $\{g_i\}_{sym}$, based on the magnitude of the gradient. As described in [11], 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,

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

where

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

The magnitude of the gradient is bounded from below and that lower bound is determined by the measure F as was shown in Ref. [11] and defined as

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

where $\|\cdot\|_F$ 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, illustrated in Fig. 5 for a grid of triangular cells, proceeds 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 stencil, $\{g_i\}_{symF,0}$, or $symF_0$, is the $\{g_i\}_{sym}$ stencil.

4. Let $\{v_i^R\}$ be the subset of the cells in $\{v_j\}$, but not in $\{g_i\}_{symF,0}$, 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):

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

where

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

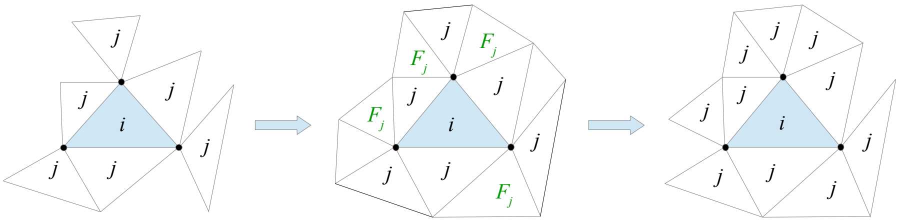
$$d_m = \sqrt{\Delta x_m^2 + \Delta y_m^2 + \Delta z_m^2} \ \text{ and } \ w_j = \frac{1}{d_j^{p(1)}} . \tag{14}$$

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 $\{g_i\}_{symF}$, and set F_0=F, $(A^T A)_0$= $A^T A$ and s_0 =s.

In Ref. [11], 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 were made to the algorithm when extending it to 3-D to simplify its implementation in the VULCAN-CFD [10] code. These modifications were:

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 [11] where it was shown that the $fn1$ stencil had behavior similar to the $symF$ stencil even when the $fn2$ and nn stencils failed.



$\{g_j\}_{sym}$

Find the cells, F_j, that satisfy the $F < K_F F_0$ criteria from the unused cells in $\{g_j\}_{nn}$

augmented stencil, $\{g_j\}_{symF}$

Fig. 5: The process for constructing the *symF* augmented *fn1* stencil for a grid of triangular cells.

2. <u>Inviscid Flux</u>

2.1 *Cell-Face State Variable Reconstruction*

The inviscid fluxes in the unstructured-grid solver are computed using an upwind flux scheme. Currently, either the LDFSS [20] or the HLLC scheme [21] 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 chemical species, static density, cartesian velocity components, vibrational/electronic temperature, static pressure, turbulent kinetic energy, and specific turbulent dissipation rate, respectively.

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 2nd-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

$$q_f^L \ = \ q_i \ + \ \overline{\nabla q_i} \ \cdot \ \vec{r}_{if} \tag{15}$$

$$q_f^R \ = \ q_j \ + \ \overline{\nabla q_j} \ \cdot \ \vec{r}_{jf} \ , \tag{16}$$

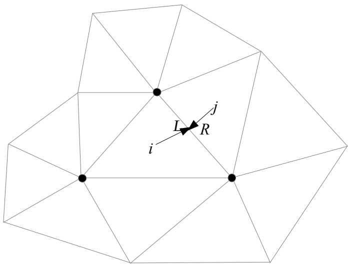


Fig. 6: The 2^{nd}-order reconstruction of the L and R states to the cell face midpoint.

where q_i and q_j are the cell average solution, $\overline{\nabla q_i}$ and $\overline{\nabla q_j}$ are the cell average weighted least squares gradients, of cells i and j, respectively, and $\vec{r}_{if}$ and $\vec{r}_{jf}$ are the vectors shown in Fig. 6. A more general implementation of the scheme above, which is an unstructured-grid interpretation of Fromm's scheme [22], is the higher-order variable extrapolation (or UMUSCL) reconstruction scheme [23]. UMUSCL was implemented to allow to additional control of the dissipation of the scheme. The UMUSCL scheme can be written as

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

$$q_f^R = q_j + \frac{\chi}{2}(q_i - q_j) + (1-\chi)\overline{\nabla q_j} \cdot \vec{r}_{jf}, \tag{18}$$

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

$$\chi \begin{cases} 0, & \text{gives Fromm's scheme} \\ -1, & \text{gives a } 2^{nd}\text{-order fully upwind MUSCL-type scheme} \\ 1/3, & \text{gives a } 3^{rd}\text{-order upwind biased MUSCL-type scheme} \end{cases} \tag{19}$$

2.2. *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 2^{nd}-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 1^{st}-order through the use of some sort of gradient limiter. The gradient limiter has been implemented in two different ways for the UMUSCL scheme; a 1-D "face"-based limiter approach or a multidimensional "stencil"-based limiter approach. Both methods were described in detail in [11] with the modified form of the multidimensional limiter process (*MLP*) of Park and Kim [24] having been found to be the preferred method for general grids. Figure 7 presents the cells involved in the *fn2* and *nn* stencils used to compute the limiter coefficients for a cell *i.* 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

$$\widetilde{q_f^L} = q_i + \Phi_i(q_i)[\frac{\chi}{2}(q_j - q_i) + (1-\chi)\overline{\nabla q_i} \cdot \vec{r}_{if}] \tag{20}$$

$$\widetilde{q_f^R} = q_j + \Phi_j(q_j)[\frac{\chi}{2}(q_i - q_j) + (1-\chi)\overline{\nabla q_j} \cdot \vec{r}_{jf}], \tag{21}$$

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 and

$$q_f = q_i + \frac{\chi}{2}(q_j - q_i) + (1-\chi)\overline{\nabla q_i} \cdot \vec{r}_{if}, \tag{22}$$

where the value of χ is consistent with the value used in Eqs. (17,18).

The primary difference between the implementation of the *MLP* approach and other stencil-based approaches is that the reconstruction of the solution is to the cell nodes instead of to the cell face midpoints. Figure 7 illustrates the stencil cells and nodes that participate in the computation of the *MLP* "node-reconstruction-based" limiter coefficients, for a *fn2* stencil $\Phi_i(q_i^{MLP_{fn2}})$, and an *nn* stencil, $\Phi_i(q_i^{MLP_{nn}})$. The modified *MLP* limiter coefficient of a general stencil for a cell *i*, $\Phi_i(q_i^{MLP_{g_i}})$, described in Ref. [11] is computed using the equation

$$\Phi_i(q_i^{MLP_{g_j}}) \;=\; min(1,\;[\Psi^{g_j}_{i,n}(q_{i,n}), n=1 \rightarrow N_i, nodes_i]), \tag{23}$$

and

$$\Psi^{\{g_j\}}_{i,n}(q_{i,n}) \;=\; \begin{cases} \phi_{i,n}\left(\dfrac{q_i^{max(\{g_j\})} - q_i}{q_{i,n} - q_i}\right), \text{if} \;\; (q_{i,n} - q_i) > 0 \\ \phi_{i,n}\left(\dfrac{q_i^{min(\{g_j\})} - q_i}{q_{i,n} - q_i}\right), \text{if} \;\; (q_{i,n} - q_i) < 0 \\ 1 \qquad\qquad\qquad \text{if} \;\; (q_{i,n} - q_i) = 0 \end{cases}, \tag{24}$$

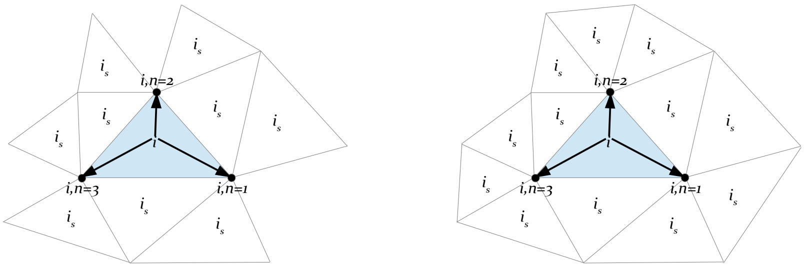
where $\Phi_i(q_i^{MLP_{g_j}})$, is computed in two steps: 1) the cell-node limiter coefficient, $\Psi^{g_j}_{i,n}(q_{i,n})$, is computed at each node, n, that is a vertex of cell i, using Eq. (24), 2) the cell-limiter coefficient is computed as the minimum of those cell-node limiter coefficients using Eq. (23). In Eq. (24), the quantities, $q_i^{max(\{g_j\})}$ and $q_i^{min(\{g_j\})}$, are the maximum and minimum values of the solution of the stencil cells, $\{g_j\}$ 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.,

$$q_{i,n} = q_i + \overline{\nabla q_i} \;\cdot\; \vec{r_{in}}. \tag{25}$$

The node-limiter coefficient in Eq. 24, $\phi_{i,n}$, is computed using a generalization of the form used in Refs. [25,26], which can be written as

$$\phi_{i,n}\left(\frac{\Delta_2}{\Delta_1}\right) \;=\; \frac{L(\Delta_1, \Delta_2)_{(limiter)}}{\Delta_1}, \tag{26}$$

where, $L(\Delta_1, \Delta_2)_{(limiter)}$, can be any 1-D limiter function found in the literature, where Δ_1 and Δ_2 are suitable successive 1-D reconstruction variable differences. Currently, the Sweby [27], van Leer [28], van Albada [29], *MLP-u2* [24], and Koren [30] limiter functions have been implemented in the VULCAN-CFD unstructured-grid solver so as to be consistent as possible with the limiters implemented in the structured-grid solver.



fn2 stencil *nn* stencil

Fig. 7: The *fn2* and *nn* 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 $N_{\text{i}, nodes} = 3$.

The resulting gradient limiter approach of Eq.s (23-26), $\Phi_i(q_i^{MLP_{\{g_j\}}})$, while not a strict implementation of the $\Phi_i(q_i^{MLP_{nn}})$ limiter, has been found to be robust and suitable for use on general 3-D mixed element grids. A slight difference in unstructured limiters, with respect to the structured-grid solver, is the replacement of the Venkatakrishnan limiter [26], with the *MLP-u2* limiter [23]. The *MLP-u2* limiter is a modified form of Venkatakrishnan's limiter. Venkatakrishnan's limiter has the form

$$\phi_{i,n}\left(\frac{\Delta_2}{\Delta_1}\right) \;=\; \left[\frac{1}{\Delta_1}\,\frac{(\Delta_2^2 + \epsilon^2)\Delta_1 + 2\,\Delta_1^2\,\Delta_2}{\Delta_2^2 + 2\,\Delta_1^2 + \Delta_1\,\Delta_2 + \epsilon^2},\right] \tag{27}$$

where

$$\Delta_1 = q_i^{min\,\text{or}\,max} - q_i,\;\; \Delta_2 = \overline{\nabla q_i}\;\cdot \vec{r_{in}}\;\;,\;\; \epsilon^2 = (K_L|\vec{r_{in}}|)^3 \tag{28}$$

and K_L is an 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

$$\epsilon^2 = \frac{K_1}{1+\theta}\,\Delta q_{i,n}^2 \;\;\text{where}\;\; \Delta q_{i,n} = q_{i,n}^{max} - q_{i,n}^{min},\;\; \theta = \frac{\Delta q_{i,n}}{K_2|\vec{r_{in}}|^{K_3}} \;\;\text{and}\;\; K_1 = 5,\;\; K_2 = 5,\;\; K_3 = 1.5. \tag{29}$$

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 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.,

$$\Phi_i(q_i^{MLP_{\{g\}}})=min[\Phi_i(P_i^{MLP_{\{g\}}}),\Phi_i(q_i^{MLP_{\{g\}}})]. \tag{30}$$

3. 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,

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

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

where T_{ve} and T_{tr} are the vibrational/electronic and translational/rotational temperatures, respectively. Three cell-face average gradient schemes have been implemented in the VULCAN-CFD code. Two schemes due to Hasselbacher and Blazek [31] and a third scheme due to Nishikawa [32]. Hasselbacher observed that computing $\overline{\nabla q_f}$ as a simple average of the face neighbor cell-average gradients, i.e.,

$$\overline{\nabla q_f} = \frac{(\overline{\nabla q_i}+\overline{\nabla q_j})}{2}, \tag{31}$$

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. The edge-normal augmented cell-face gradient method, as defined by Hasselbacher, is

$$\widehat{\nabla q_f}^{EN}=\overline{\nabla q_f} - [\overline{\nabla q_f}\cdot\hat{e}_{ij} - \frac{(q_j-q_i)}{|\vec{e}_{ij}|}]\ \hat{e}_{ij}, \tag{32}$$

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

$$\widehat{\nabla q_f}^{FT}=\overline{\nabla q_f} - [\overline{\nabla q_f}\cdot\hat{e}_{ij} - \frac{(q_j-q_i)}{|\vec{e}_{ij}|}](\frac{\hat{n}_f}{\hat{n}_f\cdot\hat{e}_{ij}}). \tag{33}$$

The third method, developed by Nishikawa, considers the Hasselbacher's augmentation term to be a damping term. For a detailed description of the derivation of the Nishikawa's alpha damped cell face gradient method, for a 2^{nd}-order accurate finite volume scheme, the reader is referred to Ref. [32]. In a finite volume context, Nishikawa's alpha-damped cell-face average gradient, $\widehat{\nabla q_f}^{AD}$, has the form:

$$\widehat{\nabla q_f}^{AD}=\overline{\nabla q_f} + \alpha(\frac{\hat{n}_f}{|\vec{e}_{ij}\cdot\hat{n}_f|})\ (q_f^{\ R}-q_f^{\ L}) \tag{34}$$

where α is a damping coefficient, having an optimum value of 4/3 in the context of a 2^{nd}-order accurate finite volume scheme, and $q_f^{\ L}$ and $q_f^{\ R}$ are the left and right reconstruction of the viscous state variables to the cell face. The viscous state variables are reconstructed using Fromm's scheme where

$$q_f^{\ L}=q_i+\overline{\nabla q_i}\cdot\vec{r}_{if} \tag{35}$$

$$q_f^{\ R}=q_j+\overline{\nabla q_i}\cdot\vec{r}_{jf}. \tag{36}$$

The first term in Eq. (34) is the consistent term approximating the face gradient, and the second term is considered to be an adjustable damping term. An extensive comparison of cell face gradient schemes can be found in Jalai [34].

4. Numerical Experiments and Validation

4.1. *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, 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 [35]. The Wilcox (2006) $k-\omega$ 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 HLLC scheme with the higher-order cell-face states constructed using UMUSCL, $\chi=1/3$, with the cell-average gradients limited using the $\Phi_i(q_i^{MLP_{\{g_j\}}})$ gradient limiter and the Park and Kim *MLP-u2* 1-D 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 Ref. [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 15,000 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. 8, 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 9 presents a plot of Mach no. (filled) contours and the static pressure (black line) 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 10 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 11 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.

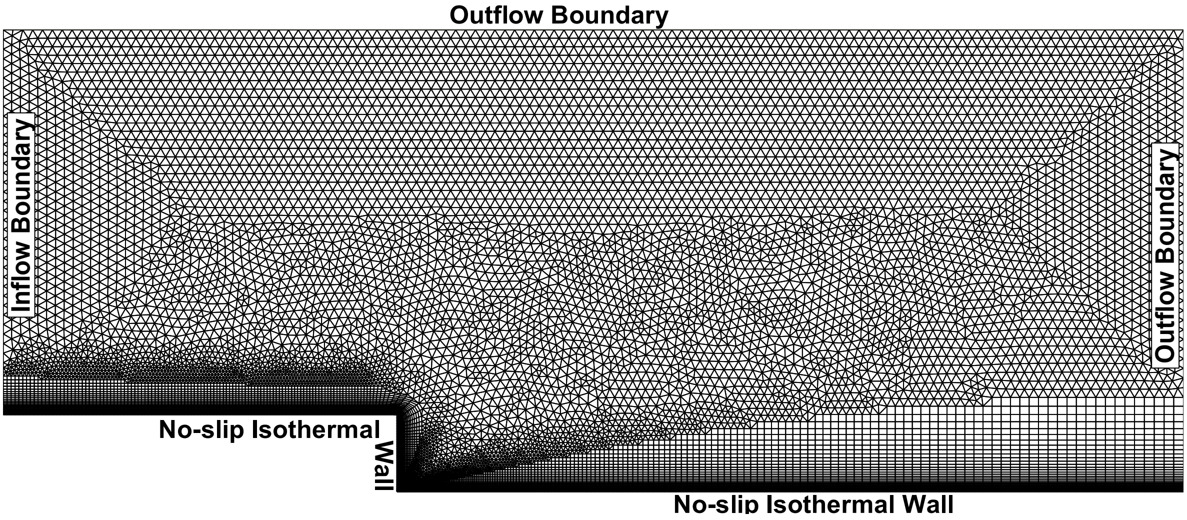


Fig. 8: Computational grid and boundary conditions for hypersonic flow over a 2-D backward facing step.

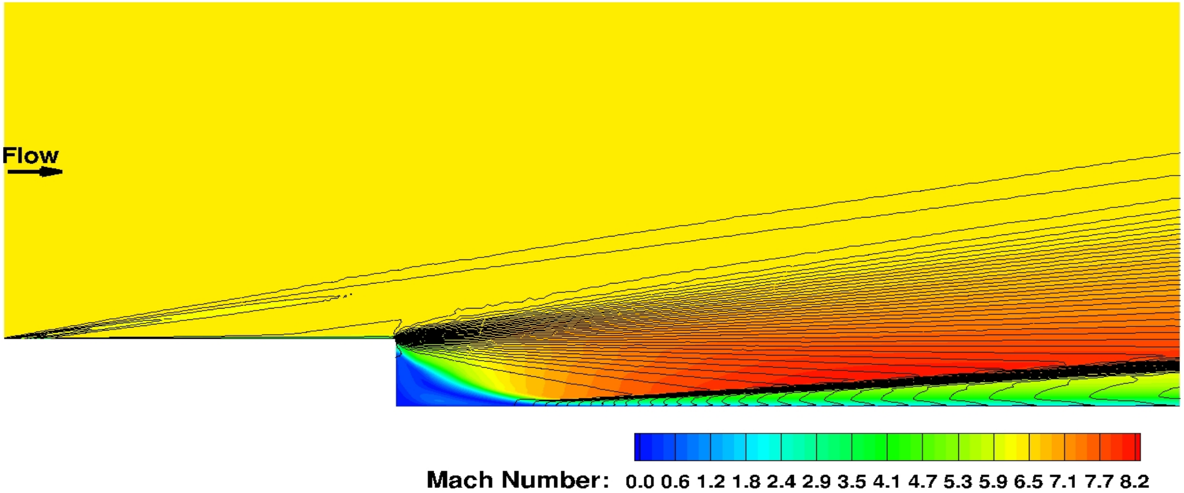


Fig. 9: Mach No. and static pressure contour plot of hypersonic flow over a 2-D backward facing step using the *fn2* stencil.

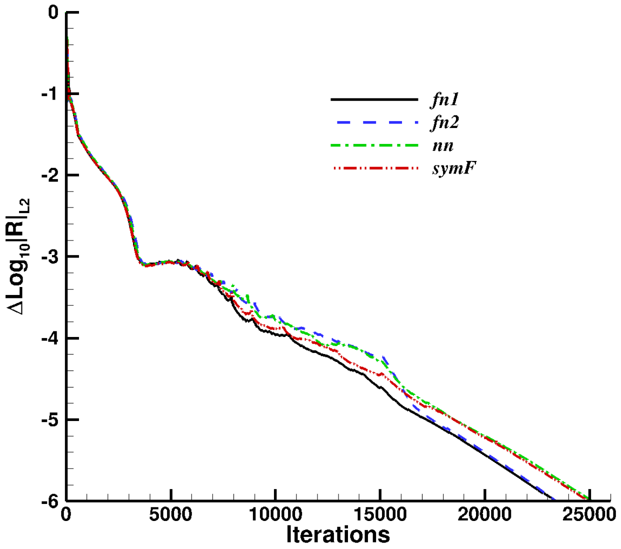


Fig. 10: A comparison of the residual convergence behavior for the computation of hypersonic flow over a 2-D backward facing step using the *fn1*, *fn2*, *nn*, and *symF* stencils.

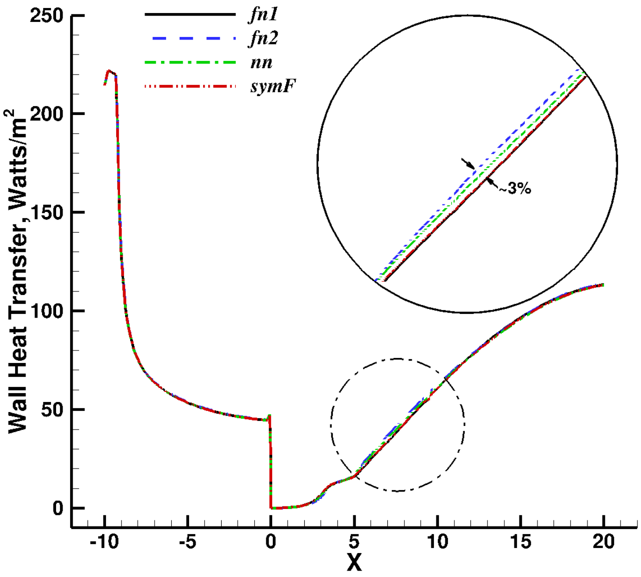


Fig. 11: A comparison of the axial distribution of wall heat transfer for the computation of hypersonic flow over a 2-D backward facing step using the *fn1*, *fn2*, *nn*, and *symF* stencils.

4.2 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 [37] 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, $\chi=0$, with the cell-average gradients limited using the $\Phi_i(q_i^{MLP_{\{g_j\}}})$ gradient limiter and the van Leer 1-D 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 pyramidal 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. 12. 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 3.4 and 6.2 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 13 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 14 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 15 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.

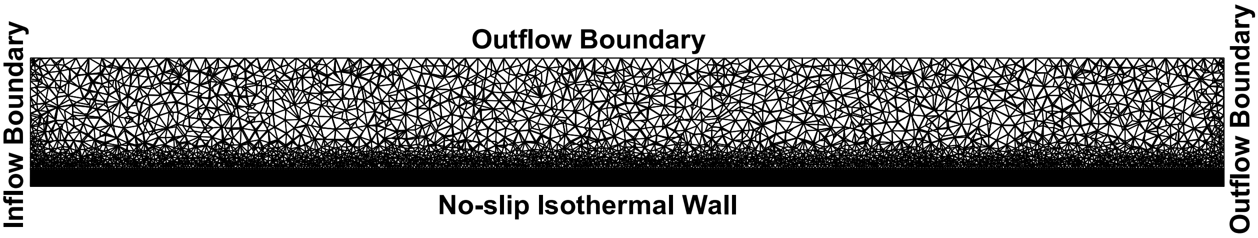


Fig. 12: Computational grid and boundary conditions for hypersonic flow over a flat plate.

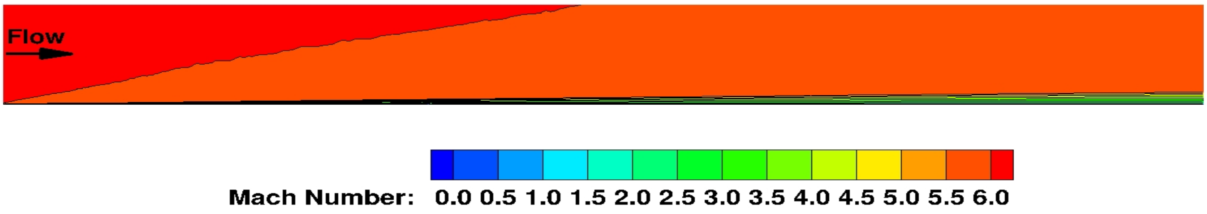


Fig. 13: Mach no. contour plot of hypersonic flow over a 2-D backward facing step using the *nn* stencil.

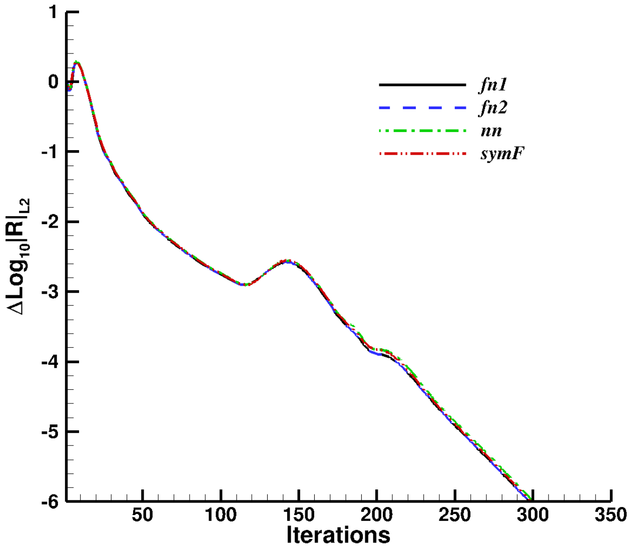


Fig. 14: A comparison of the residual convergence behavior for the computation of hypersonic flow over a 2-D flat plate using the *fn1*, *fn2*, *nn*, and *symF* stencils.

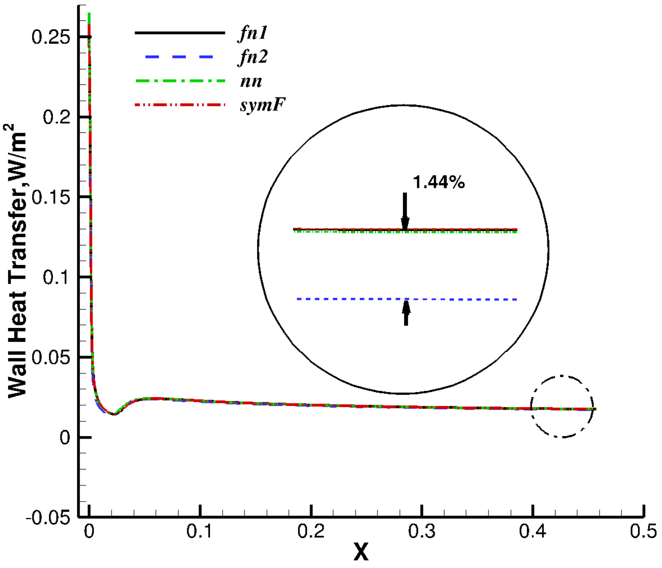


Fig. 15: A comparison of the axial distribution of wall heat transfer for the computation of hypersonic flow over a 2-D flat plate using the *fn1*, *fn2*, *nn*, and *symF* stencils.

4.3 Validation Against HiFiRE 7 REST Scramjet Engine Shock Tunnel Data Using a Grid Containing Tetrahedral, Pyramidal, Prismatic, and Hexahedral Cells.

The University of Queensland experimental test of a 75% scale replica of the HiFiRE 7 REST scramjet engine conducted in the T4 Stalker Tube [12] was simulated previously using the unstructured-grid solver [9]. An early implementation of the *fn2* stencil was used in this prior work and very good comparisons with the inlet centerline experimental data were obtained. In the current work, the unstructured-grid solver was used to compute the zero degree angle-of-attack, tare (no fuel injection) test point, and the bilateral symmetry of the model geometry was exploited to generate a computational mesh for half of the REST scramjet flow path shown in Fig. 16, using Pointwise®. The resulting surface grid was predominantly made up of triangles with quadrilaterals being used along the blunt leading edges and in the internal portion of the flow path. This surface grid was then marched normal to the wall surface into the interior of the computational domain to form a boundary layer grid made up of prisms and hexes. This boundary layer grid then transitioned into pyramids and tetrahedra resulting in a mixed cell type grid. The grid has a total of 44,568,851 cells consisting of 19,198,513 tetrahedral cells, 1,436,197 pyramidal cells, 18,455,646 prismatic, and 5,488,495 hexahedral cells, which were then decomposed into 768 partitions using ParMETIS.

In Ref. [9], the unstructured-grid solver was run with an early implementation of the *fn2* stencil, with the same inflow/reference conditions that Chan et al. used in Ref. [12] to perform their CFD simulation using the VULCAN-CFD structured-grid solver. These conditions were: P_{ref}= 1675.0 Pascals, T_{ref}= 228.0 Kelvin, and velocity, U_{ref}= 2379 m/s. As mentioned above, the zero degree angle-of-attack, tare case, was selected. However, it is important to note that the angel-of-attack convention reported in [12] is relative to the combustor centerline. The angle of attack relative to the x-axis, which runs parallel to the forebody plate, is 6 degrees, as shown in Fig. 17. A thermally perfect, chemically frozen air gas mixture was used to simulate the test gas, which at the given reference conditions yields a Mach number of 7.845 and a unit Reynolds number of Re_{ref}= 4.1x10^6/m. The model surfaces were treated as no-slip, isothermal (300.0 Kelvin) walls, using the Wilcox wall matching formulation [35]. The Menter Baseline two-equation turbulence model [38] was used to compute the Reynolds stresses and Reynolds heat flux (Pr_t=0.9). The inviscid fluxes were computed using the LDFSS scheme with the higher-order cell-face states being reconstructed using the UMUSCL, $\chi=0$, scheme with the cell-average gradients limited using the $\Phi_i(q_i^{MLP_{\{g_j\}}})$ gradient limiter and the Park and Kim *MLP-u2* 1-D limiter function. The viscous fluxes were computed using the face tangent gradient method. The boundary layer trips were not modeled. The governing equations were solved in a fully coupled manner with the matrix based SGS scheme with linear-solve inter-partition coupling and 10 subiterations. The computational domain was initialized to the reference conditions, and a 5-mm thick "initial boundary layer" was constructed by linearly blending the no-slip isothermal wall condition into the interior of the computational domain.

A Mach contour plot on the symmetry plane of the forebody and inlet, superimposed over the computational grid, is presented in Fig. 17. Figure 17 illustrates the small size of the forebody and cowl leading edges relative to the forebody boundary layer thickness. Figure 17 also shows that the shocks are captured without significant oscillations and that the forebody leading edge and forebody compression corner shocks are both captured with a small number of cells even where the grid is predominantly tetrahedral in nature.

Figure 18 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. The computation was run for 1500 iterations using a 1st-order advection scheme to establish the flow. The advection scheme was then switched to the 2nd-order scheme, and the solution was run an additional 1500 iterations. The gradient limiter was then frozen at iteration 3000, which can be seen in the residual plot in Fig. 20, as an abrupt drop in the residual. The solution was then continued for an additional 2000 iterations to make certain that each solution was stable with the frozen limiter. It was deemed "safe" to freeze the limiter at cycle 3000 because the mass flow error, surface integrated heat transfer, as well as the integrated forces and moments had all reached an asymptotic value. This approach resulted in the L_2 of the Residual of the *fn2*, *nn* and *symF* stencil schemes converging approximately 5.0 orders of magnitude relative to the maximum value. However, the behavior of the *fn1* solution residual after the limiter was frozen was significantly different from the *fn2*, *nn* and *symF* solution residuals. Fig. 20 shows that the *fn1* solution increases nearly 2 orders of magnitude relative to the other solutions after iteration 3500. The precise reason for this is not currently known but it is thought that there are instabilities occurring in the *fn1* solution that the gradient limiter is, by virtue of it being frozen, is not able to suppress.

The body side and inlet cowl side wall static pressure at the symmetry plane are presented in Figs. 19 and 20. As in Ref. [9], the comparison of the wall pressure distribution with respect to the experimental wall pressure data is very reasonable and is slightly better than the solution reported in Ref. [12] that was obtained using the VULCAN-CFD code structured-grid solver. Figures 21 and 22 present a comparison of the body and cowl centerline wall heat flux using the four stencils. The *fn2*, *nn* and *symF* stencils give very similar wall heat transfer distribution on the body and cowl side. The *fn1* stencil produced a similar wall heat transfer distribution on the body side. However, on the cowl side, the results were approximately ten percent higher than the *fn2*, *nn* and *symF*

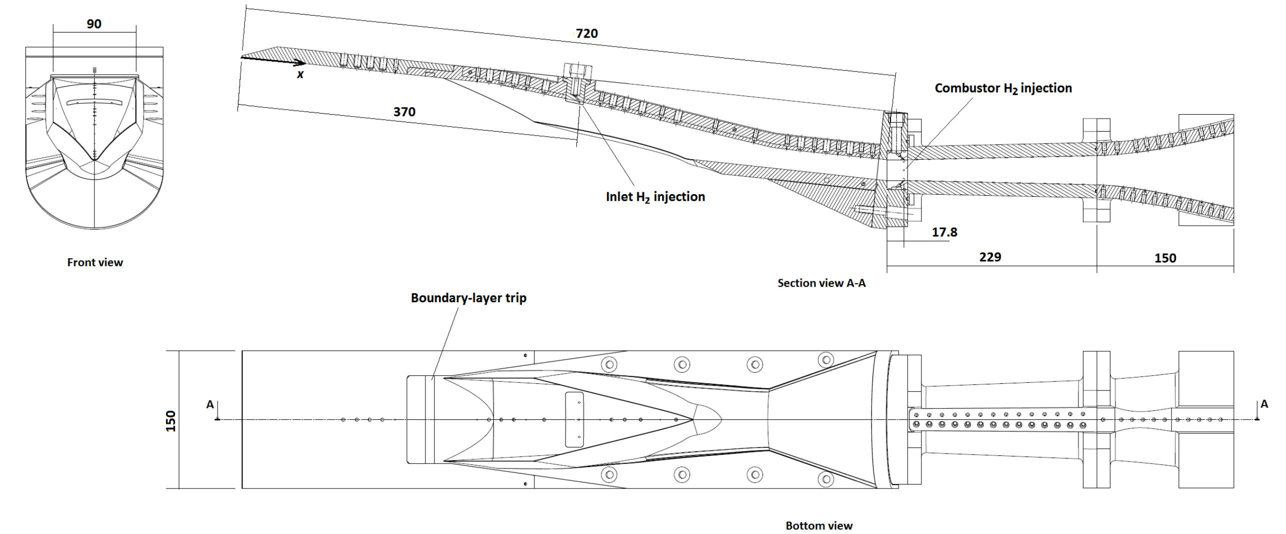


Fig. 16: Schematic of HIFiRE 7 REST scramjet engine experimental model as installed in the University of Queensland T4 Stalker Tube, at 0 degrees angle of attack relative to the combustor centerline, 6 degrees angle-of-attack relative to the forebody plate surface (the labeled x-axis), reprinted with permission from Ref. [12].

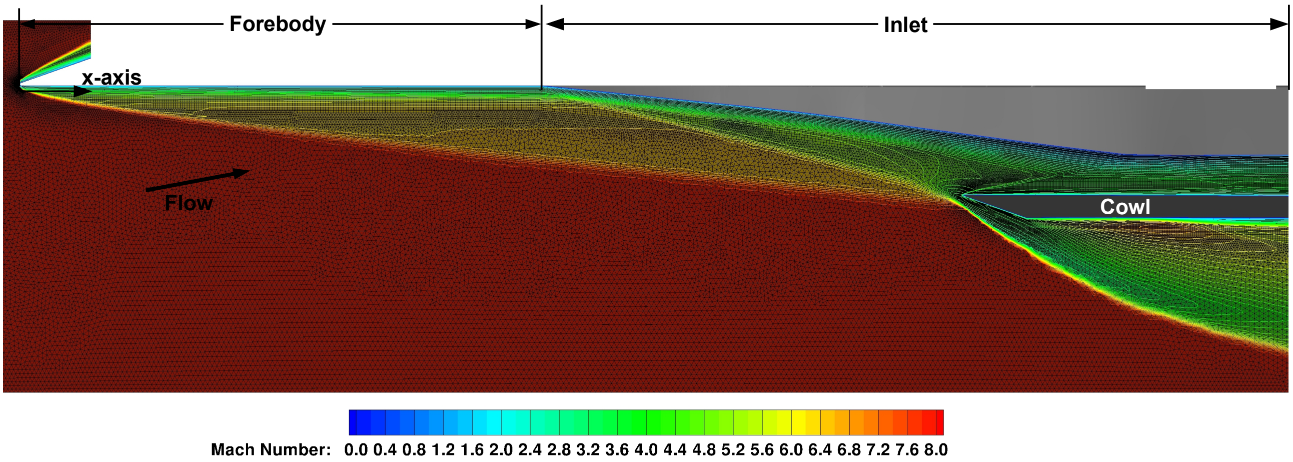


Figure 17: HIFiRE 7 REST scramjet engine forebody/inlet symmetry plane computational Mach contours with unstructured grid superimposed computed using the *fn2* stencil.

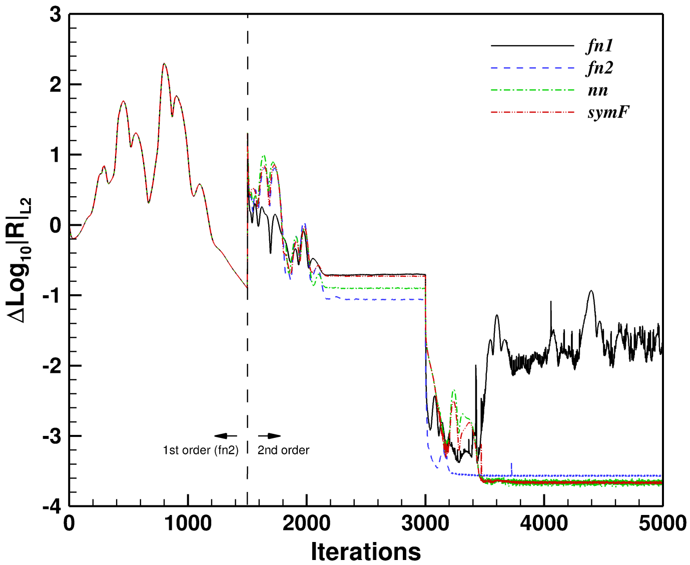


Fig. 18: A comparison of the residual convergence behavior for the HIFiRE 7 REST scramjet engine inlet using the *fn1*, *fn2*, *nn*, and *symF* stencils.

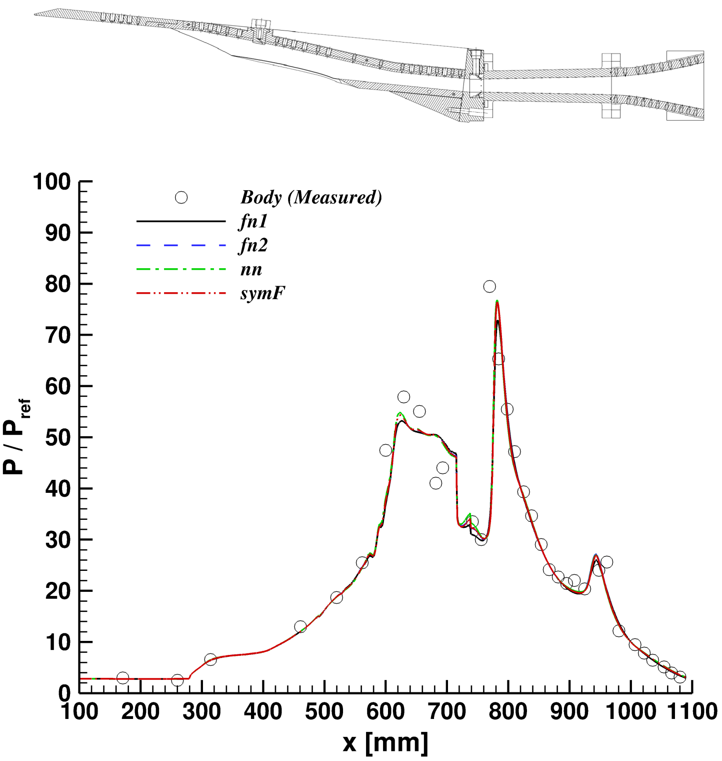


Figure 19: A comparison of the computed HIFiRE 7 REST scramjet engine body wall center-line pressure distributions using the *fn1*, *fn2*, *nn*, and *symF* stencils, with experimental data from Ref. [12].

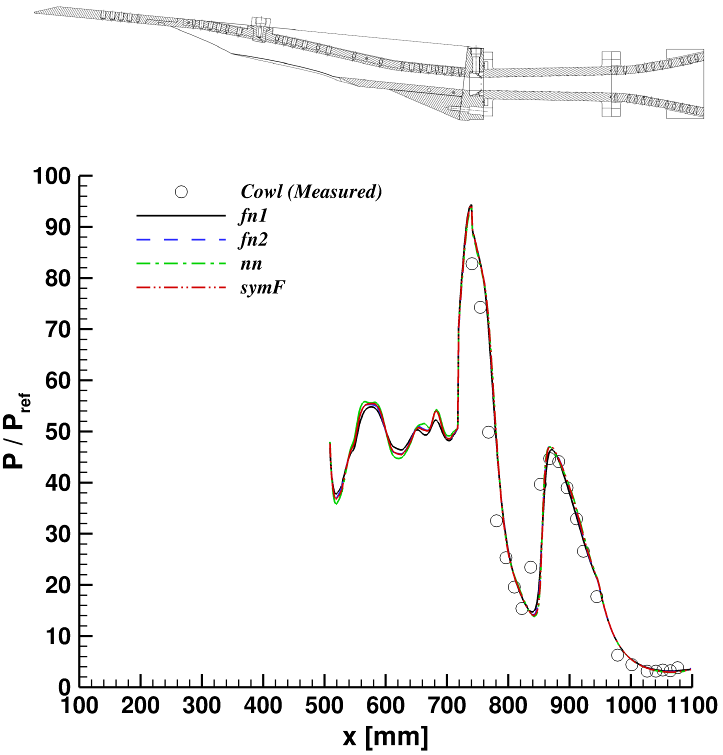


Figure 20: A comparison of the computed HIFiRE 7 REST scramjet engine cowl wall center-line pressure distributions using the *fn1*, *fn2*, *nn*, and *symF* stencils, with experimental data from Ref. [12].

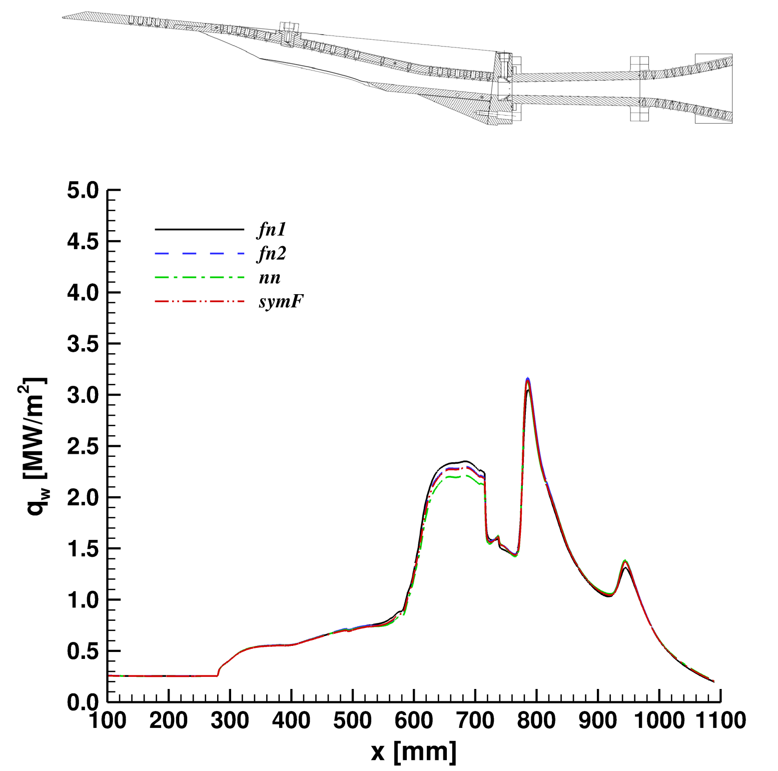


Fig. 21: A comparison of the computed HIFiRE 7 REST scramjet engine body wall center-line axial distribution of wall heat transfer using the *fn1*, *fn2*, *nn*, and *symF* stencils.

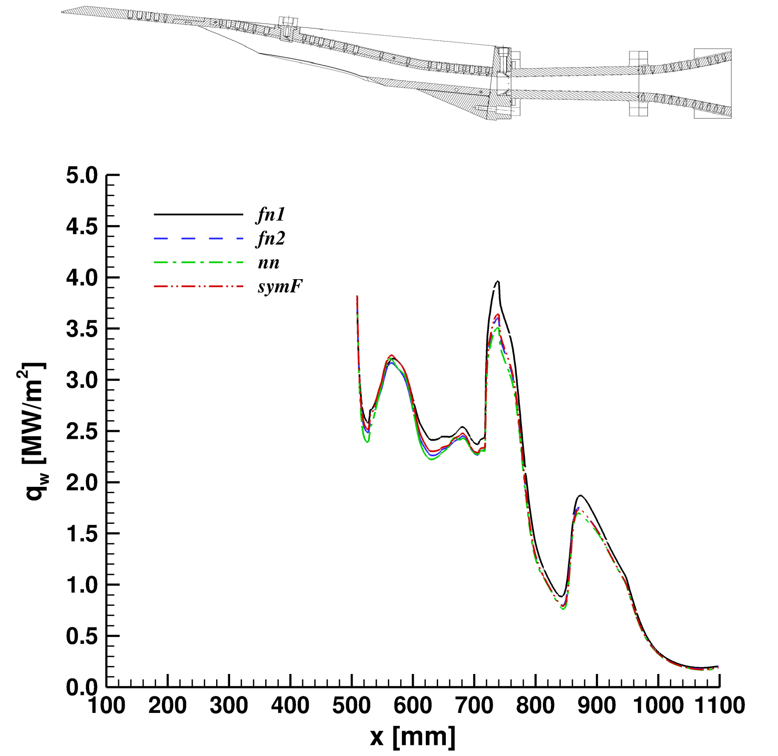


Fig. 22: A comparison of the computed HIFiRE 7 REST scramjet engine cowl wall center-line axial distribution of wall heat transfer using the *fn1*, *fn2*, *nn*, and *symF* stencils.

stencils. In Ref. [10], the *fn1* stencil was found to be unstable for the Mach 8 blunt body grid tested, and Fig. 18 shows that the *fn1* stencil solution convergence behavior was problematic after the limiter was frozen.

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, 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, Table 3 shows stencil statistics for the HiFiRE REST grid. For this case, the mean stencil size of the *fn1* stencil was 4.67. Table 3 shows that the average *symF* stencil is significantly smaller and varies less over the computational domain than the *nn* stencil. However, the table also shows that unlike the backward facing step and flat plate cases the *symF* stencil varies more than the *fn2* stencil, and the relative augmentation cost metric indicates that the *symF* stencil was 1.4 and 5.1 times less expensive than the *fn2* and *nn* stencils, respectively. In all, the mean *symF* stencil size was significantly smaller than the *nn* stencil but not significantly smaller than the *fn2* stencil.

Table 3: Stencil statistics and relative cost of the *fn2*, *nn* and *symF* stencils for the HiFiRE Inlet.

Stencil Augmentation Type	Minimum (min.) Stencil Size	Mean Stencil Size	Maximum (max.) Stencil Size	Stencil Standard Deviation (σ)	Relative Augmentation Cost
fn2	9	17.6	31	3.16	1
nn	15	51.8	129	19.1	3.64
symF	4	14	83	5.48	0.72

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 3-D extension/modification of a Frobenius-norm minimization-based approach recently introduced by Nishikawa. The extension of Nishikawa's *symF* method to 3-D turbulent viscous flow for use in the VULCAN-CFD described in detail in Ref. [10] was summarized. In addition, a modified form of the MLP cell-average gradient limiter method of Park and Kim that can be applied to a general stencil was also summarized here.

The numerical behavior of WLSQ cell-average gradients computed using the *fn1*, *fn2*, *nn* and 3-D *symF* stencil construction methods were examined using three test cases. Turbulent hypersonic flow over a 2-D backward facing step using a 2-D grid, a 2-D flat plate using a 3-D grid and a 3-D inlet geometry from the HiFiRE 7 REST scramjet experiment were computed. Stencil size statistics, convergence behavior and wall quantities of interest, for twelve computations were collected, examined and compared. In addition, in the case of the HiFIRE 7 inlet, comparisons with experimental data were also made. All four stencil types were found to produce solutions that had similar convergence behavior, shock capturing characteristics, and axial distributions of wall quantities of interest for all three flows. In addition, the four stencil types were found to produce solutions that compared well with the inlet body and cowl wall pressure data from the HiFRE 7 REST inlet experiment. Stencil analysis showed that the 3-D *symF* stencil construction method was found to produce stencils having the smallest size and standard deviation on 3-D as well as 2-D unstructured grids. The 3-D *symF* stencil was found to produce stencils that resulted in a WLSQ cost that was only slightly less than the *fn2* stencil WLSQ cost and significantly less than the *nn* stencil WLSQ cost for all test cases. Given that the *symF* stencil was not a clear winner over the *fn2* stencil, further investigation of other newly emergent techniques is warranted. An example of one such approach is a 3-D hybrid grid extension of the face-averaged nodal gradient cell-centered approach recently developed by Nishikawa. This approach, has the potential to reduce the cost of the computation of the least squares gradient by up to a factor 6 for 3-D non-hexahedral grids and by a factor of 2 for hexahedral grids, relative to the *fn2* and *nn* cell-average gradient approaches explored in the current work.

Acknowledgments

This work was supported by the Hypersonic Technology Project, through the Hypersonic Airbreathing Propulsion Branch and the Computational AeroSciences Branch of the NASA Langley Research Center.

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. White, J.A., Nishikawa, H, and Baurle, R.A., **Weighted Least-squares Cell-Average Gradient Construction Methods For The VULCAN-CFD Second-Order Accurate Unstructured Grid Cell-Centered Finite-Volume Solver**, *AIAA Paper 2019-0127*, (Jan. 2019).

11. Nishikawa, H., **Efficient Gradient Stencils for Robust Implicit Finite-Volume Solver Convergence on Distorted Grids**, *Journal of Computational Physics*, Vol. 386, 2019, pp. 486-501.

12. Chan, Y.K., Razzaqi, S. Smart, M.K. and Wise, D., **Freejet Testing of the 75%-scale HIFiRE 7 REST Scramjet Engine**, AIAA-2014-2931, 19th AIAA International Space Planes and Hypersonic Systems and Technologies Conference, (June 2014).

13. 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.

14. 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).

15. 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).

16. 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.

17. 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).

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

19. 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).

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

21. 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.

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

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

24. 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.

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

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

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. 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.

29. 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.

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

31. 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).

32. Nishikawa, H., **Beyond Interface Gradient: A General Principle for Constructing Diffusion Schemes**, *AIAA Paper 2010-5093*, (June 2010).

33. 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.

34. 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.

35. Wilcox, D. C., **Wall Matching, a Rational Alternative to Wall Functions**, *AIAA paper 89-0611*, (Jan. 1989).

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

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

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