Preprint accepted in Journal of Computational Physics (originally submitted in August, 2018).
Download the journal version at https://doi.org/10.1016/j.jcp.2019.02.026

Efficient Gradient Stencils for Robust Implicit Finite-Volume Solver Convergence on Distorted Grids

Hiroaki Nishikawa*
National Institute of Aerospace, Hampton, VA 23666, USA

March 2, 2019

Abstract

Two gradient-stencil augmentation techniques are discussed: symmetric and F-decreasing augmentations, for efficient and robust implicit finite-volume-solver convergence on distorted unstructured grids. The former augments a face-neighbor stencil with extra cells to increase the symmetry of the stencil as much as possible, and the latter adds further extra cells to decrease the reciprocal of the Frobenius norm of a scaled least-squares matrix to minimize the lower bound of the magnitude of the gradient. These techniques are proposed as efficient ways of overcoming a known stability issue with a face-neighbor stencil. It is demonstrated that the F-decreasing augmentation in combination with the symmetric augmentation yields robust and efficient gradient stencils on highly distorted quadrilateral and triangular grids.

1 Introduction

Stability of implicit finite-volume solvers widely used in practical unstructured-grid codes [1, 2, 3, 4, 5, 6] is affected by various algorithmic components, e.g., relaxation schemes, Jacobian construction, pseudo time steps, limiter functions, discretizations, and so on. Among others, a least-squares (LSQ) stencil used to compute gradients is also known to impact the stability of finite-volume solvers. In Ref.[7], Haider *et. al.* studied stability of an explicit cell-centered finite-volume scheme for advection in relation to the extent of gradient stencils. They have shown that the scheme can be unstable on tetrahedral and hybrid grids if the gradients are computed with face-connected neighbors, and concluded that the LSQ gradient stencil should be extended beyond the face neighbors to ensure stability. Later, Zangeneh and Gooch demonstrated also that a larger gradient stencil improved stability [8]. An extended stencil can be constructed by augmenting the face-neighbor stencil with extra cells. A popular approach is to use face neighbors of the face neighbors (see, e.g., Refs.[6, 9]); the resulting stencil is referred to as the face2 stencil. Another is to use the cells sharing the vertex of the cell of interest (see, e.g., Ref.[10]); the resulting stencil is called here the vertex stencil. These augmented stencils are widely used in practical cell-centered finite-volume solvers. However, the stability comes with the additional cost of processing a large number of neighbors in each gradient calculation, especially in the vertex stencil. The number of vertex neighbors is not bounded, and can be unlimitedly large. For example, it can be as large as hundreds if a vertex of a cell is at a polar singularity, from which hundreds of grid lines are originated [11]. If not hundreds, the number of vertex neighbors is known to be around 50 on average in practical unstructured grids [10]. It is, therefore, of great practical interest to construct a gradient stencil as robust as the vertex and face2 stencils but with a minimal number of neighbors.

Very few studies, however, are found in the literature on efficient gradient stencil construction for robust implicit finite-volume-solver convergence on unstructured grids. One approach is the smart augmentation technique in Ref.[12], where for each vertex of a cell, an extra cell closest to the cell centroid is chosen from cells surrounding the vertex and added to the face neighbor stencil. Although it brings some improvements, it only adds as many cells as the number of vertices, which may not be enough to ensure stability. Our experience shows that an implicit solver diverges with the smart augmentation stencil on highly distorted grids. Another approach is discussed in Ref.[13], which selects neighbors to construct a four- or three-point stencil that resembles a structured-grid stencil. However, they report that the stencil exhibits an erratic behavior for different cell types, and conclude that the Green-Gauss method is more reliable. Yet another approach can be found in Ref.[14], which attempts to construct 'body-fitted' gradient stencils based on boundary information. Improvements in accuracy are reported, but the algorithm does not address the stability issue raised in Ref.[7].

*Associate Research Fellow (hiro@nianet.org), 100 Exploration Way, Hampton, VA 23666 USA,

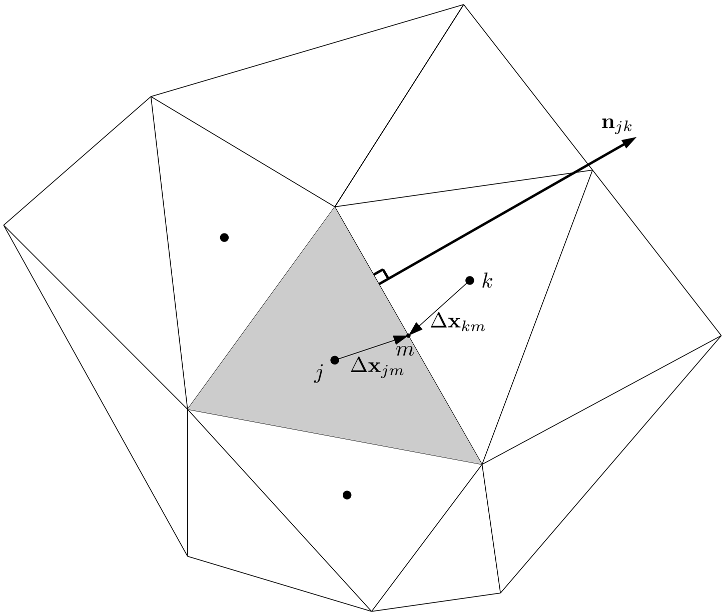


Figure 1: A triangular grid for a cell-centered finite-volume discretization.

The grids used in Ref.[14] are not as distorted as those with which we are concerned, and thus remains to be demonstrated for practical unstructured grids in three dimensions.

This paper is intended to shed light on efficient gradient stencil construction for robust implicit solver convergence on unstructured grids. Focusing on the stencil augmentation, we discuss two novel approaches: symmetric and F-decreasing augmentations. Symmetric augmentation is motivated by the fact that a cell-centered finite-volume scheme with a face-neighbor stencil is unstable on tetrahedral grids but stable on hexahedral grids [7]. It is conjectured that a stencil having a symmetric structure, (i.e., consists of pairs of neighbor cells located symmetrically with respect to the cell of interest), improves the stability of implicit solvers. The F-decreasing augmentation is based on another observation that an implicit finite-volume solver is stable with zero gradients (e.g., first-order scheme or with a limiter function). It implies that a gradient stencil minimizing the magnitude of the gradient improves the stability of implicit solvers. The F-decreasing augmentation attempts to create such a stencil by decreasing the reciprocal of the Frobenius norm of a scaled LSQ matrix that determines the lower bound of the magnitude of the gradient. These augmentations are demonstrated numerically for inviscid problems with highly distorted grids.

This paper focuses on implicit solvers, which are employed in many practical unstructured-grid codes [1, 2, 3, 4, 5, 6]. However, it is equally relevant to explicit solvers because the main focus is on the spatial discretization and the gradient stencil is known to impact the stability of explicit solvers also [7]. Here, we focus on the effect of gradient stencils on iterative convergence rather than gradient accuracy because the ability to obtain solutions is more critically important in practical computations. Accuracy is not relevant unless a solution can be obtained. Nevertheless, we consider LSQ methods that are exact for linear functions (i.e., first-order accurate gradients) on arbitrary grids, and do not consider the Green-Gauss gradient, which is, as is well known, zero-th order accurate on irregular grids. In contrast to Ref.[8], where stencils are identified that cause instability by an eigenvalue analysis of a global residual Jacobian and augment only such stencils, we seek simpler stencil augmentation methods that can be readily applied to existing codes without forming the exact second-order residual Jacobian and computing its eigenvalues.

2 Implicit Finite-Volume Solver

Consider the steady Euler equations:

$$\partial_x \mathbf{f} + \partial_y \mathbf{g} = 0, \tag{1}$$

where $\mathbf{f} = \mathbf{f}(\mathbf{u})$ and $\mathbf{g} = \mathbf{g}(\mathbf{u})$ are the fluxes, and $\mathbf{u}$ is the vector of conservative variables, which is discretized at a cell j (see Figure 1) by a cell-centered finite-volume discretization:

$$\mathbf{Res}_j = \sum_{k \in \{k_j\}} \mathbf{\Phi}_{jk} A_{jk} = 0, \tag{2}$$

where $\{k_j\}$ is a set of face neighbors of the cell j, A_{jk} is the magnitude of the scaled face-normal vector $\mathbf{n}_{jk}$ across j and k: $A_{jk} = |\mathbf{n}_{jk}|$. The numerical flux $\mathbf{\Phi}_{jk}$ is defined by the Roe flux [15]:

$$\mathbf{\Phi}_{jk} = \frac{1}{2}\left[\mathbf{f}_n(\mathbf{u}_L) + \mathbf{f}_n(\mathbf{u}_R)\right] - \frac{1}{2}|\mathbf{A}_n|(\mathbf{u}_R - \mathbf{u}_L), \tag{3}$$

where $\mathbf{f}_n = \mathbf{f}\hat{n}_x + \mathbf{g}\hat{n}_y$, $\hat{\mathbf{n}}_{jk} = \mathbf{n}_{jk}/A_{jk} = (\hat{n}_x, \hat{n}_y)$, $\mathbf{A}_n = \partial \mathbf{f}_n / \partial \mathbf{u}$ evaluated by the Roe-averages [15], and the subscripts L and R denote linearly reconstructed solutions from within the cell j and the neighbor cell k, respectively:

$$\mathbf{u}_L = \mathbf{u}_j + \nabla \mathbf{u}_j \cdot \Delta \mathbf{x}_{jm}, \quad \mathbf{u}_R = \mathbf{u}_k + \nabla \mathbf{u}_k \cdot \Delta \mathbf{x}_{km}, \tag{4}$$

where $\mathbf{u}_j$ and $\mathbf{u}_k$ are the solution values stored at the centroids of the cell j and k, respectively, and $\Delta \mathbf{x}_{jm}$ and $\Delta \mathbf{x}_{km}$ are the vectors pointing from the centroids of j and k, respectively, to the face midpoint. To perform the reconstruction, the gradients, e.g., $\nabla \mathbf{u}_j$ and $\nabla \mathbf{u}_k$, need to be computed at all cells from the current solution values stored at the cell centroids. The LSQ method is widely used for the gradient calculation on unstructured grids; the main subject of this paper is the impact of the LSQ stencil on the implicit finite-volume solver.

The resulting global system of residual equations,

$$0 = \mathbf{Res}(\mathbf{U}_h), \tag{5}$$

where $\mathbf{U}_h$ denotes the global solution vector, is solved by an implicit defect-correction solver, which is representative of practical unstructured-grid solvers [1, 2, 3, 4, 5, 6].:

$$\mathbf{U}_h^{n+1} = \mathbf{U}_h^n + \Delta \mathbf{U}_h, \tag{6}$$

$$\frac{\partial \overline{\mathbf{Res}}}{\partial \mathbf{U}} \Delta \mathbf{U}_h = -\mathbf{Res}(\mathbf{U}_h^n), \tag{7}$$

where n is the iteration counter, and the Jacobian $\partial \overline{\mathbf{Res}} / \partial \mathbf{U}$ is the exact differentiation of the first-order version of the residual $\overline{\mathbf{Res}}$ (i.e., zero LSQ gradients). The linear system (7) is relaxed by the Gauss-Seidel relaxation scheme to a specified tolerance or a fixed number of relaxations. Note that the implicit solver is relevant not only to steady problems but also to unsteady problems, where a global system of residual equations needs to be solved at each time step for implicit time-stepping schemes. The stability of the implicit solver relies on various algorithmic components and parameters as mentioned earlier. Our focus is on the effect of the LSQ gradient stencil. Therefore, we fix all other parameters: 150 Gauss-Seidel relaxations per iteration, no limiters, the Roe flux and its exact Jacobian. A pseudo-time term is added to the Jacobian in a testbed code used here, but the CFL number for a pseudo time is set to be 10^6 by default. The CFL number will be adjusted based on the residual convergence behavior (e.g., reduced by an order of magnitude if the residual increases); it is typically applied only during the initial transient and the CFL number quickly reachess 10^6.

3 Least-Squares Gradient Method

The LSQ 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 suffices to fit a linear polynomial. Suppose we wish to compute the gradient of a solution variable u at a cell j, and have a set $\{g_j\}$ of $N(\geq 2)$ nearby cells (i.e., gradient stencil) available for fitting the linear polynomial:

$$u_k = u_j + \overline{\partial_x u}_j (x_k - x_j) + \overline{\partial_y u}_j (y_k - y_j), \tag{8}$$

where $k \in \{g_j\}$, (x_j, y_j) and (x_k, y_k) denote the centroid coordinates of the cell j and the neighbor k, respectively, and $\overline{\partial_x u}_j$ and $\overline{\partial_y u}_j$ are the derivatives we wish to compute. As the number of neighbors often exceeds two on unstructured grids, the polynomial fit typically leads to an overdetermined problem:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{9}$$

where

$$\mathbf{A} = \left[\begin{array}{cc} w_1(x_1 - x_j) & w_1(y_1 - y_j) \\ \vdots & \vdots \\ w_k(x_k - x_j) & w_k(y_k - y_j) \\ \vdots & \vdots \\ w_N(x_N - x_j) & w_N(y_N - y_j) \end{array} \right], \quad \mathbf{x} = \left[\begin{array}{c} \overline{\partial_x u}_j \\ \overline{\partial_y u}_j \end{array} \right], \quad \mathbf{b} = \left[\begin{array}{c} w_1(u_1 - u_j) \\ \vdots \\ w_k(u_k - u_j) \\ \vdots \\ w_N(u_N - u_j) \end{array} \right], \tag{10}$$

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

$$w_k = \frac{1}{d_k^p}, \quad d_k = \sqrt{(x_k - x_j)^2 + (y_k - y_j)^2}, \tag{11}$$

where p is a parameter, e.g., zero (unweighted LSQ), one (fully weighted LSQ), or any other real value. The overdetermined LSQ system can be solved in various ways. One is the normal equation approach, where we form a 2×2 system by multiplying Equation (9) by the transpose of $\mathbf{A}$, denoted by $\mathbf{A}^T$, from the left:

$$\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b}, \tag{12}$$

and invert it (e.g., via Cholesky decompostion) to obtain the gradient:

$$\mathbf{x} = \left(\mathbf{A}^T \mathbf{A} \right)^{-1} \mathbf{A}^T \mathbf{b}. \tag{13}$$

Another is the QR factorization via Householder transformation [16], which directly solves the overdetermined system (9) as

$$\mathbf{x} = \mathbf{R}^{-1} \mathbf{Q}^T \mathbf{b}, \tag{14}$$

where $\mathbf{Q}$ is the orthonormal matrix and $\mathbf{R}$ is the upper triangular matrix generated from $\mathbf{A}$ by the QR factorization [16]. In either approach, the solution can be expressed in the following form:

$$\left[\begin{array}{c} \overline{\partial_x u}_j \\ \overline{\partial_y u}_j \end{array} \right] = \sum_{k \in \{g_j\}} \left[\begin{array}{c} c_{jk}^x \\ c_{jk}^y \end{array} \right] (u_k - u_j), \tag{15}$$

where c_{jk}^x and c_{jk}^y are the LSQ coefficients to be computed and stored at all cells once for a given stationary grid. It is clear that the cost of the gradient calculation is directly proportional to the number of neighbors involved in the gradient stencil. Note that the LSQ gradient calculation can incorporate boundary solutions or gradients if available from a physical boundary condition [17]. However, we do not consider using boundary values since such is not always possible (e.g., not all solution variables are known at an outflow boundary).

As mentioned in the Introduction, the set of neighbors $\{g_j\}$ affects the stability of finite-volume solvers. In fact, it has been known by practitioners that adding more neighbors tends to stabilize a finite-volume solver. Ref.[7] has formally shown that a finite-volume scheme is unstable with the face-neighbor gradient stencil on tetrahedral and hybrid grids, and adding extra cells to the stencil cures the instability. Later, Ref.[8] also showed that a larger stencil size led to stability. In this regard, the most robust stencil would be the vertex stencil on triangular/tetrahedral grids and the face2 stencil on quadrilateral/hexahedral grids [6], but these stencils substantially increase the cost of the gradient computation and require a large memory for storing the LSQ coefficients, especially in three dimensions [10]. Our objective is to construct a gradient stencil that achieves similar robustness with a fewer number of neighbors. Two augmentations are proposed, which, in combination, would result in a robust and efficient gradient stencil.

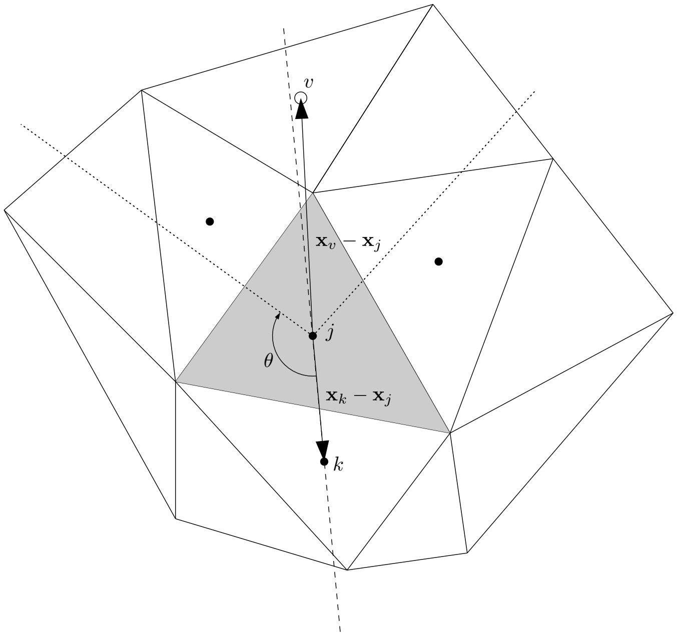
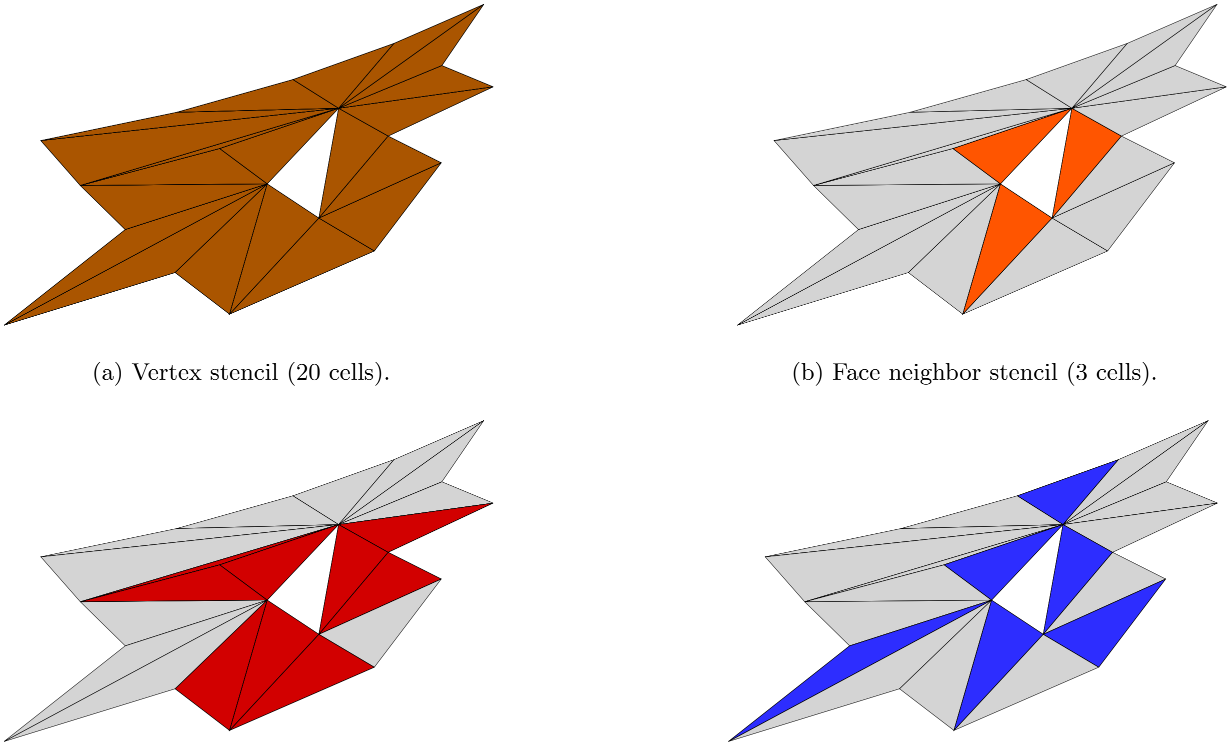


Figure 2: Symmetric stencil construction: a cell is chosen for a face k, from those between the dotted lines defined by the angle θ, whose centroid is located across from the neighbor k and closest to the (dash) line passing through j and k. The dotted line on the right is a reflection of the left one with respect to the dash line.



(a) Vertex stencil (20 cells).

(b) Face neighbor stencil (3 cells).

(c) Face2 stencil (9 cells).

(d) Symmetric stencil (6 cells).

Figure 3: Gradient stencils. Stencil members are colored within the vertex stencil, and so all cells are colored in the vertex stencil.

4 Approaches to LSQ Stencil Augmentation

4.1 Symmetric Augmentation

We begin with the face neighbor stencil $\{k_j\}$ for a cell j, and add to it cells from the union of the vertex and face2 neighbors $\{v_j\}$ that will symmetrize the stencil as much as possible. The symmetric augmentation begins with one of the face neighbors, $k \in \{k_j\}$, and search for a cell located symmetrically with respect to the centroid of the cell j from the face neighbor k. It is performed as follows. First, define the initial symmetric stencil, denoted by $\{sym_j\}$, as the face neighbor stencil:

$$\{sym_j\} \leftarrow \{k_j\}. \tag{16}$$

Then, for each $k \in \{k_j\}$, define a unit vector pointing from j to k:

$$\mathbf{e}_{jk} = \frac{\mathbf{x}_k - \mathbf{x}_j}{|\mathbf{x}_k - \mathbf{x}_j|}, \tag{17}$$

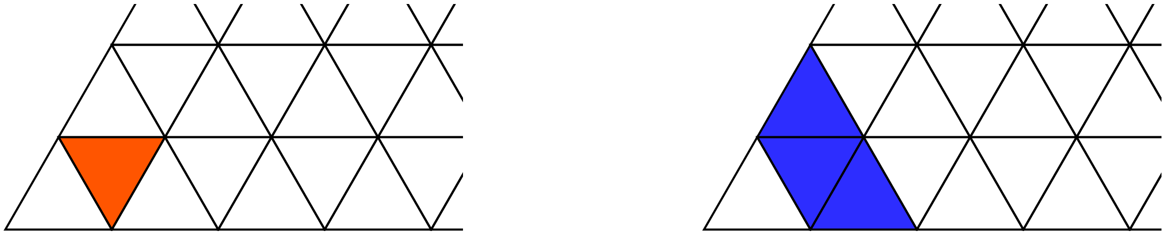
where $\mathbf{x}_j$ is the centroid coordinate vector of the cell j, and $\mathbf{x}_k$ is the centroid coordinate vector of the face neighbor k. See Figure 2. For a boundary face, we define $\mathbf{x}_k$ by the face-midpoint. Then, for all $v \in \{v_j\}$ except those already in the current stencil, compute the direction cosine:

$$d_v = \frac{\mathbf{x}_v - \mathbf{x}_j}{|\mathbf{x}_v - \mathbf{x}_j|} \cdot \mathbf{e}_{jk}, \tag{18}$$

where $-1 \leq d_v \leq 1$, and $\mathbf{x}_v$ is the centroid coordinate vector of the cell v. Define a subset $\{v_j^-\}$ of $\{v_j\}$ satisfying the following condition:

$$d_v < \cos(\theta), \quad \theta = \frac{3\pi}{4}, \tag{19}$$

where $\theta = 3\pi/4$ has been chosen, instead of $\pi/2$, to exclude those located far from the line passing through $\mathbf{x}_j$ and $\mathbf{x}_k$. The subset $\{v_j^-\}$ contains, therefore, a group of cells in the region between the two dotted lines in Figure 2. Note that $d_v < 0$ for all v in $\{v_j^-\}$. Finally, select the one with the minimum d_v, and add it to the stencil $\{sym_j\}$. Performing the same for all faces $\{k_j\}$, we arrive at the final symmetric augmentation stencil $\{sym_j\}$. The algorithm as described is directly applicable to arbitrary grids in two and three dimensions.



(a) Face neighbor stencil.

(b) Symmetric stencil.

Figure 4: Gradient stencils for the corner cell. The face neighbor stencil has only one neighbor, leading to the ill-conditioning of the LSQ matrix (left). The symmetric augmentation adds two extra cells, and resolves the ill-conditioning problem (right). Here, the symmetric stencil turns out to be identical to the vertex and face2 stencils.

The algorithm will find one extra cell per face, but it may fail near a boundary with an empty set $\{v_j^-\}$, indicating that no cells exist on the other side. In such a case, we do not add any cell and proceed to the next face. If a symmetric cell is found for all faces, the resulting stencil will have a symmetric configuration, i.e., the stencil consists of pairs of neighbor cells located symmetrically with respect to the cell of interest, similar to a face-neighbor stencil on a regular quadrilateral grid that leads to stability [7]. An example is shown and compared with other stencils in Figure 3, where all stencils are indicated by colored triangles over the vertex stencil for the cell of interest (white). For this irregular grid (extracted from the grid in Figure 6), the vertex stencil has 20 neighbors as shown in Figure 3(a). On the other hand, the number of cells in the face-neighbor and face2 stencils are limited by the number of faces: three and nine. The symmetric stencil is shown in Figure

3(d). It has an extended stencil with six neighbors (compare it with the face neighbor stencil), addressing the instability issue [7] with a fewer number of neighbors than the vertex and face2 stencils, and having pairs of neighbor cells approximately located in a symmetrical configuration. Note that the use of the boundary face centroid in the symmetric augmentation has the important effect for a cell having only one neighbor that it provides enough neighbors to avoid ill-conditioning of the LSQ problem as illustrated in Figure 4. Finally, it is pointed out that the symmetric augmentation adds two remote cells in each coordinate direction for an orthogonal hexahedral/quadrilateral grid.

Numerical experiments show that the symmetric augmentation greatly improves convergence on distorted grids, but it can still lead to instability on highly-curved grids. This brings the second step: F-decreasing augmentation, which is the subject for the next section.

4.2 F-Decreasing Augmentation

To further augment the symmetric stencil, we introduce another approach based on the magnitude of the gradient. Consider the normal equation:

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}, \tag{20}$$

where

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} \displaystyle\sum_{k\in\{g_j\}} w_k^2 \Delta x_k^2 & \displaystyle\sum_{k\in\{g_j\}} w_k^2 \Delta x_k \Delta y_k \\ \displaystyle\sum_{k\in\{g_j\}} w_k^2 \Delta x_k \Delta y_k & \displaystyle\sum_{k\in\{g_j\}} w_k^2 \Delta y_k^2 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \overline{\partial_x u}_j \\ \overline{\partial_y u}_j \end{bmatrix}, \quad \mathbf{A}^T\mathbf{b} = \begin{bmatrix} \displaystyle\sum_{k\in\{g_j\}} w_k^2 \Delta x_k \Delta u_k \\ \displaystyle\sum_{k\in\{g_j\}} w_k^2 \Delta y_k \Delta u_k \end{bmatrix}, \tag{21}$$

$$\Delta x_k = x_k - x_j, \quad \Delta y_k = y_k - y_j, \quad \Delta u_k = u_k - u_j, \quad k = 1, 2, , \cdots, N. \tag{22}$$

Let us rescale the system without changing the solution $\mathbf{x}$:

$$s^{-1}\mathbf{A}^T\mathbf{A}\mathbf{x} = \tilde{\mathbf{b}}, \tag{23}$$

where

$$\tilde{\mathbf{b}} = s^{-1}\mathbf{A}^T\mathbf{b}, \quad s = \sum_{k\in\{g_j\}} w_k^2 d_k, \tag{24}$$

so that $\tilde{\mathbf{b}} = O(\Delta u)$, where Δu is a typical solution variation over the stencil. It is easy to show that the L_2 norm $||\cdot||_2$ of the vector $s^{-1}\mathbf{A}^T\mathbf{A}\mathbf{x}$ is bounded as follows:

$$||s^{-1}\mathbf{A}^T\mathbf{A}\mathbf{x}||_2^2 = \sum_{i=1}^{2} |\boldsymbol{\ell}_i^t \mathbf{x}|^2 \leq \sum_{i=1}^{2} ||\boldsymbol{\ell}_i||_2^2 ||\mathbf{x}||_2^2 = ||s^{-1}\mathbf{A}^T\mathbf{A}||_F^2 ||\mathbf{x}||_2^2, \tag{25}$$

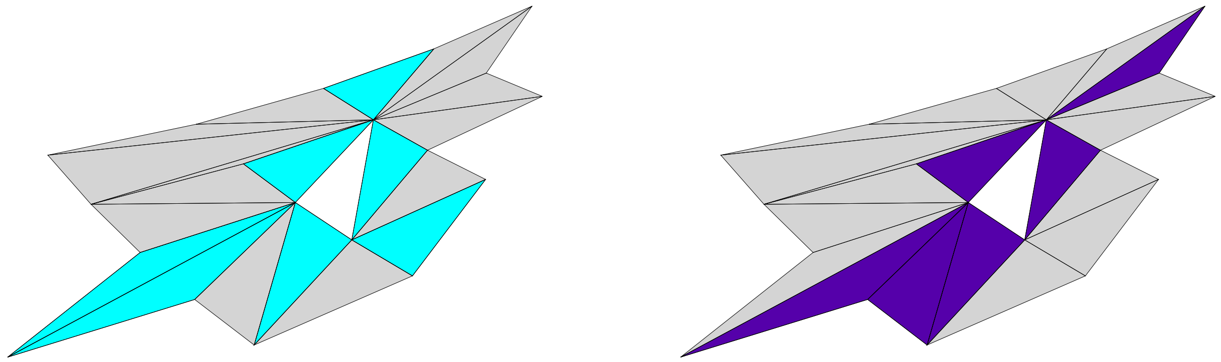
where $\boldsymbol{\ell}_i^t$ denotes the i-th row of the matrix $s^{-1}\mathbf{A}^T\mathbf{A}$, the inequality is due to the Cauchy-Bunyakovsky-Schwarz inequality applied to each i, and $||\cdot||_F$ denotes the Frobenius norm defined by $||\mathbf{B}||_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |b_{i,j}|^2}$ for an arbitrary $m\times n$ matrix $\mathbf{B}$. Substituting Equation (23) into the left hand side, we derive

$$||\mathbf{x}||_2 \geq \frac{||\tilde{\mathbf{b}}||_2}{||s^{-1}\mathbf{A}^T\mathbf{A}||_F} = \frac{||\tilde{\mathbf{b}}||_2}{s^{-1}||\mathbf{A}^T\mathbf{A}||_F} = \left(\frac{s}{||\mathbf{A}^T\mathbf{A}||_F}\right) ||\tilde{\mathbf{b}}||_2. \tag{26}$$

Therefore, the magnitude of the gradient is bounded from below, and the lower bound is determined by the measure F defined by

$$F = \frac{s}{||\mathbf{A}^T\mathbf{A}||_F}. \tag{27}$$

for a given nonzero solution variation $||\tilde{\mathbf{b}}||_2$. Note that the measure F only determines the lower bound, and thus it does not necessarily predict the magnitude of the gradient accurately. This is also because the magnitude of the vector $\tilde{\mathbf{b}}$ can also vary with the stencil although it remains of $O(\Delta u)$. Nevertheless, the measure F can be easily and efficiently computed for a given stencil, and has been found to serve well as a guide for stencil



(a) Symmetric/F-decreasing stencil $\{symF_j\}$ (7 cells).

(b) Face/F-decreasing stencil $\{\text{faceF}_j\}$ (6 cells).

Figure 5: F-decreasing augmentation applied to the symmetric and face-neighbor stencils. The F-decreasing augmentation has added one extra cell to the symmetric stencil, and three extra cells to the face-neighbor stencil.

augmentation. Our proposal is to augment a gradient stencil with cells that will decrease the measure F, i.e., potentially reducing the magnitude of the gradient and thus giving an effect of stabilizing the implicit solver.

In this note, the augmentation based on F is referred to as the F-decreasing augmentation. It is an independent algorithm that can be applied to any gradient stencil. Here, we apply it to the symmetric stencil to generate a further improved stencil $\{symF_j\}$:

1. Construct the symmetric stencil $\{sym_j\}$ as described in Section 4.1.
2. Compute $\mathbf{A}^T\mathbf{A}$ and s with $\{g_j\} = \{sym_j\}$, and then F .
3. Let $F_0 = F$, $(\mathbf{A}^T\mathbf{A})_0 = \mathbf{A}^T\mathbf{A}$, $s_0 = s$, and $\{symF_j\} = \{sym_j\}$.
4. Let $\{v_j^R\}$ be a subset of cells in $\{v_j\}$ not in $\{symF_j\}$, and N_R be the number of cells in $\{v_j^R\}$.
5. If $N_R = 0$, no further augmentation is possible. Stop.
6. For $m = 1$ to N_R, perform the following:

   (1) Compute $\mathbf{A}^T\mathbf{A}$ and s by adding the contribution of the m-th cell at (x_m, y_m):

   $$\mathbf{A}^T\mathbf{A} = (\mathbf{A}^T\mathbf{A})_0 + \begin{bmatrix} w_m^2 \Delta x_m^2 & w_m^2 \Delta x_m \Delta y_m \\ w_m^2 \Delta x_m \Delta y_m & w_m^2 \Delta y_m^2 \end{bmatrix}, \quad s = s_0 + w_m^2 d_m, \tag{28}$$

   where

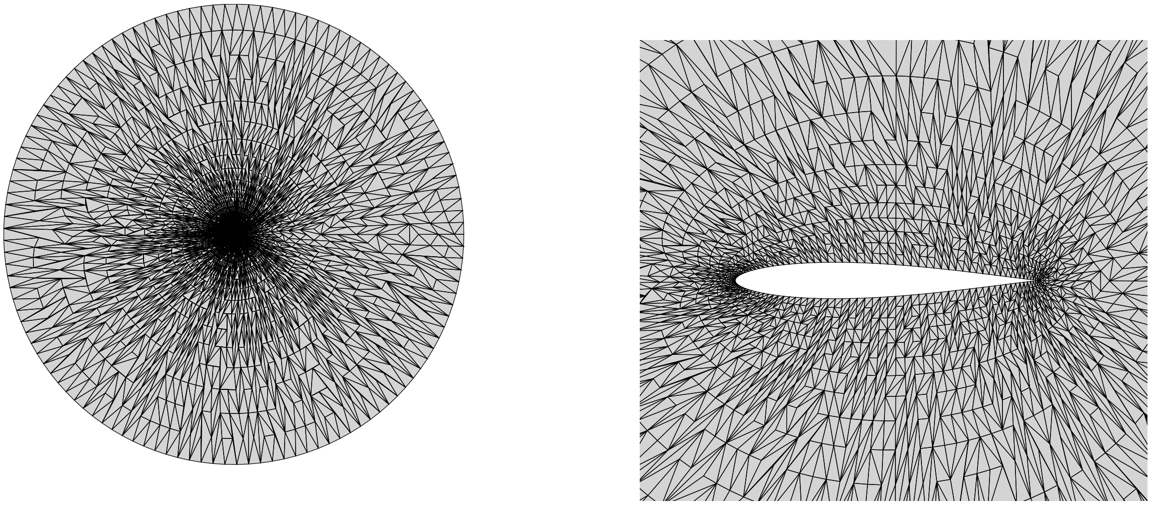
   $$\Delta x_m = x_m - x_j, \quad \Delta y_m = y_m - y_j, \quad d_m = \sqrt{\Delta x_m^2 + \Delta y_m^2}, \quad w_m = \frac{1}{d_m^p}. \tag{29}$$

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

   (3) If $F < KF_0$ add the m-th cell to $\{symF_j\}$ and set $F_0 = F$, $(\mathbf{A}^T\mathbf{A})_0 = \mathbf{A}^T\mathbf{A}$, and $s_0 = s$.

The set $\{v_j\}$ is the union of the vertex and face2 stencils, but a larger set may be used if needed. For the factor K, we set $K = 0.85$ in order to accept only significant reductions. The algorithm is efficient as it does not require any matrix inversion, and directly applicable to arbitrary grids in two and three dimensions. Note that the algorithm is dependent on the order of $\{v_j^R\}$ applied in Step 6, but finding the best from all possible orderings can take a significant amount of effort and thus the effect of ordering is not studied in this work.

Figure 5(a) shows a stencil obtained by the F-decreasing augmentation applied to the symmetric stencil in the previous example. It can be seen that one additional cell has been added to the symmetric stencil, decreasing the measure F from 34.0 to 27.6. Compare it with Figure 3(d). The total number of neighbors is 7, and it is still significantly smaller than that of the vertex stencil. Shown in Figure 5(b) is a stencil obtained by applying the



(a) Entire grid with the airfoil located at the center.

(b) Close view.

Figure 6: Irregular triangular grid for the Joukowsky airfoil case.

F-decreasing augmentation to the face-neighbor stencil, which is denoted by faceF. In this case, the F-decreasing augmentation has added three extra cells, decreasing the measure F from 67.5 to 25.4. Compare it with Figure 3(b). As will be shown later, there are cases, where the *symF* stencil successfully stabilizes the implicit solver, but the faceF stencil fails to do so. The F-augmentation is not intended to address all problems, and therefore is best used in combination with the symmetric augmentation rather than the face-neighbor stencil.

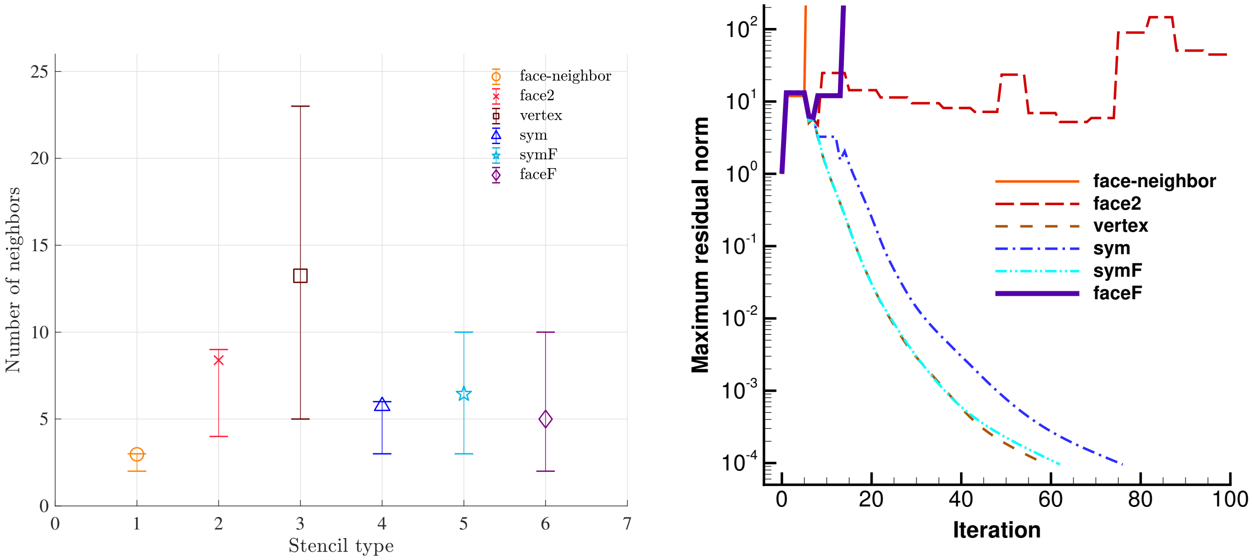
Ref.[12] states that a Frobenius norm is used as an estimate of the condition number of a LSQ system to further augment the smart-augmentation stencil for the purpose of improving gradient accuracy. It is not clear if they used $||s^{-1}\mathbf{A}^T\mathbf{A}||_F$ (because the equation is not shown in Ref.[12]), but it is interesting that they added extra cells to reduce it whereas we add cells to increase it (and thereby decrease the lower bound of the gradient magnitude). Apparently, the former just attempts to correct ill-conditioned LSQ systems, not to improve gradient accuracy in the general sense due to the lack of a rigorous proof for the relationship between the level of the gradient error and a Frobenius norm. On the other hand, our objective is clear and mathematically justified: to decrease the lower bound of the gradient magnitude. It is important to note that Equation (26) simply states that a larger Frobenius norm will decrease the lower bound for a given $||\tilde{\mathbf{b}}||_2$. It does not guarantee that the magnitude of the gradient is actually reduced, nor does it indicate anything about accuracy. Typically, gradient errors increase for a larger stencil, and such could be the case for F-augmented stencils; however we focus on iterative convergence rather than accuracy for the reason mentioned in the Introduction. It is emphasized that we only propose it as a practical measure to guide a stencil augmentation, i.e., a means to add extra cells to a given initial stencil.

5 Numerical Experiments

The *sym* and *symF* stencils are compared with the vertex, face2, faceF, and face-neighbor stencils for inviscid problems on highly irregular grids. Our focus is here on iterative convergence, which is critically important as there will be no point of speaking about accuracy if numerical solutions cannot be obtained.

5.1 Highly irregular triangular grid

Gradient stencils are compared for an inviscid flow over a Joukowsky airfoil. A triangular grid was generated over a domain defined by the airfoil of a unit chord and the outer boundary located at a distance of 50. See Figure 6. Initially, a smooth quadrilateral grid was generated. It was then subdivided into triangles by random diagonal insertions, and then further randomized by diagonal swapping. The resulting grid has 4235 nodes and 8228 triangles (121 nodes on the airfoil) with the minimum, maximum, and average cell areas being 1.97E-06, 17.3, and 0.95, respectively. The skewness measure defined by the dot product of the unit face normal vector



(a) Number of neighbors for various gradient stencils: 1=face-neighbor, 2=face2, 3=vertex, 4=sym, 5=symF, 6=faceF.

(b) Residual versus iteration.

Figure 7: Results for the Joukowsky airfoil case.

and the unit vector pointing from a cell centroid to that of a face neighbor is 0.67 on average with the minimum 0.027. Such a severely-distorted grid is typically avoided in practice, but it represents highly irregular nature that can easily arise locally on unstructured grids for complex geometries in three dimensions, and thus it is of great practical interest to see if an algorithm is robust enough to deal with such severe grid irregularity.

Figure 7(a) compares the number of neighbors for different gradient stencils. As can be expected, the vertex stencil has the largest large number of neighbors. The next largest stencil is the face2 stencil followed by the *symF* and faceF stencils. Sample stencils are shown in Figures 3 and 5.

To investigate the impact of gradient stencils on the implicit solver convergence, we consider a subsonic flow of Mach number 0.3 at an angle of attack 1.25 degrees. A slip boundary condition is applied to the airfoil, and a free stream condition is applied to the outer boundary. These boundary conditions are imposed weakly through the numerical flux. For the gradient computation, we use only neighbor cells and thus skip boundary faces. The implicit solver was used to reduce the maximum L_1 residual norm among the equation set by four orders of magnitude. All solver parameters are fixed as mentioned in Section 2. Also, the power p to the inverse distance weights is fixed as $p = 0.1$ for all cases in order to avoid the stability problem with $p = 1$ known for implicit inviscid flow solvers [18]. The LSQ coefficients are computed by the QR factorization, which is known to be more stable (in solving the LSQ problem) than the normal equation approach that suffers from a large squared condition number.

The convergence results are shown in Figure 7(b). As can be seen, the solver diverged with the face-neighbor, face2, and faceF stencils. The instability was found to emerge in the large-gradient region around the leading edge, where a suction peak arises. On the other hand, the solver converged successfully with the vertex, *sym*, and *symF* stencils with the same CFL number adjustment for the vertex and *symF*: 10^6, 10^5, 10^4, 10^3, 10^2, 10^2, and 10^6 for the first seven iterations, and then 10^6 for the rest of the iteration. The *sym* stencil required another cycle of the same adjustment. The results indicate, as desired, that the implicit solver converges with a significantly fewer number of neighbors for the *sym* and *symF* stencils. Although it took slightly more iterations to converge, the *sym* stencil is very efficient since it uses only about a half of the vertex neighbors. The convergence is very similar for the vertex and *symF* stencils; the *symF* stencil also serves as an efficient stencil with a fewer number of neighbors than the vertex stencil. Finally, the instability observed for the faceF stencil indicates that the F-decreasing augmentation alone is not sufficient to stabilize the solver. It diverges even with $K = 1$, i.e., all neighbors are added that reduce the measure F by any amount.

No significant speed-up in computing time is, however, observed by the *sym* and *symF* stencils since the gradient calculation is only a fraction of the entire implicit solver algorithm that is largely dominated by the residual and Jacobian calculations. Nevertheless, one saves a significant amount of memory with the *sym*, and *symF* stencils, and each gradient calculation is indeed made cheaper than with the vertex stencil. Saving in computing time and memory for the implicit solver would be more significant on three-dimensional grids, where the *sym*, and *symF* stencils are expected to be substantially smaller than the vertex and face2 stencils.

5.2 Curved grids

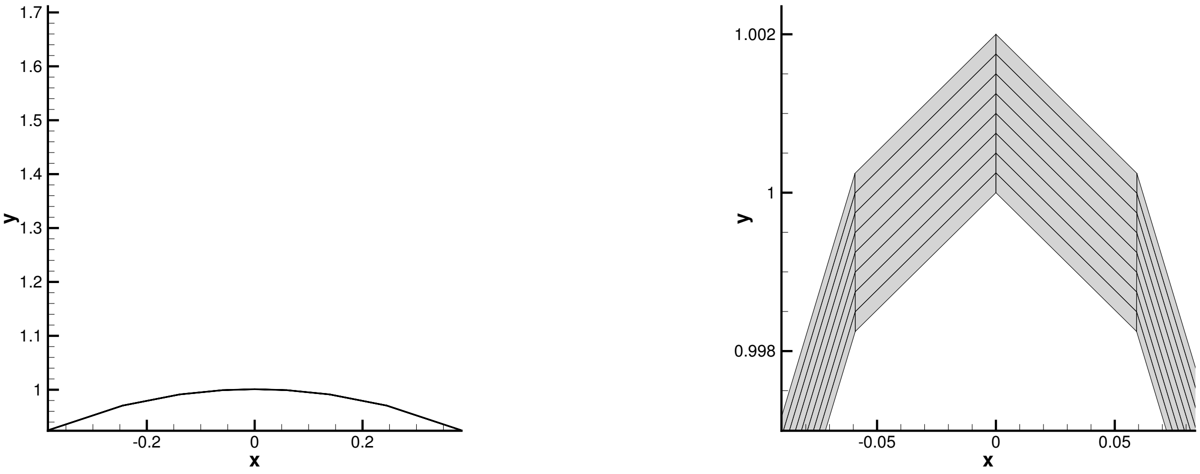
To further demonstrate the impact of the *symF* stencil on the solver stability, we consider curved high-aspect-ratio grids, which are typical in high-Reynolds-number turbulent-flow simulations and known to introduce severe stability issues [18, 19, 20, 21]. Curvature introduces an additional stability challenge to high-aspect-ratio grids as discussed in Ref.[18]; therefore, if the solver is stable on curved high-aspect-ratio grids, it is likely to be stable on non-curved high-aspect-ratio grids. We create a curved domain from a unit square domain $(\xi, \eta) \in [0,1]\times[0,1]$ by the following mapping:

$$x = (0.002\eta + 1)\cos(\theta), \quad y = (0.002\eta + 1)\sin(\theta), \quad \theta = \frac{1}{2}(\pi + \theta_{ext}) - \xi\theta_{ext}, \tag{30}$$

where $\theta_{ext} = \pi/4$, and solve the Euler equations with source terms derived such that the following function is the exact solution for all variables:

$$\sin\left(100\pi\sqrt{x^2+y^2} + \pi/6\right) + \frac{6}{5}. \tag{31}$$

The domain spans over the horizontal axis approximately from $x = -0.4$ to $x = 0.4$, and has the thickness about 0.002 (see Figure 8(a)). The solution varies predominantly in the radial direction, modeling a typical boundary layer solution. The combination of the curvature, the high-aspect-ratio, and a large solution variation in the radial direction presents a severe difficulty in gradient accuracy and solver stability [18, 19, 20, 21]. The boundary condition is imposed weakly through the numerical flux with the right state specified by the exact solution. For the gradient computation, we use only neighbor cells and skip boundary faces unless otherwise stated. As in the previous case, the implicit solver is used again to reduce the maximum residual norm by four orders of magnitude with the same set of parameters. However, in this case, we keep CFL$= 10^6$ from the first iteration to the last iteration. For this type of grid, an approximate mapping technique that maps a local stencil to a body-fitted coordinate system is known to alleviate the problems [18]. However, it requires a distance function computation and is not simple to implement; here we investigate whether stencil augmentations resolve convergence issues.

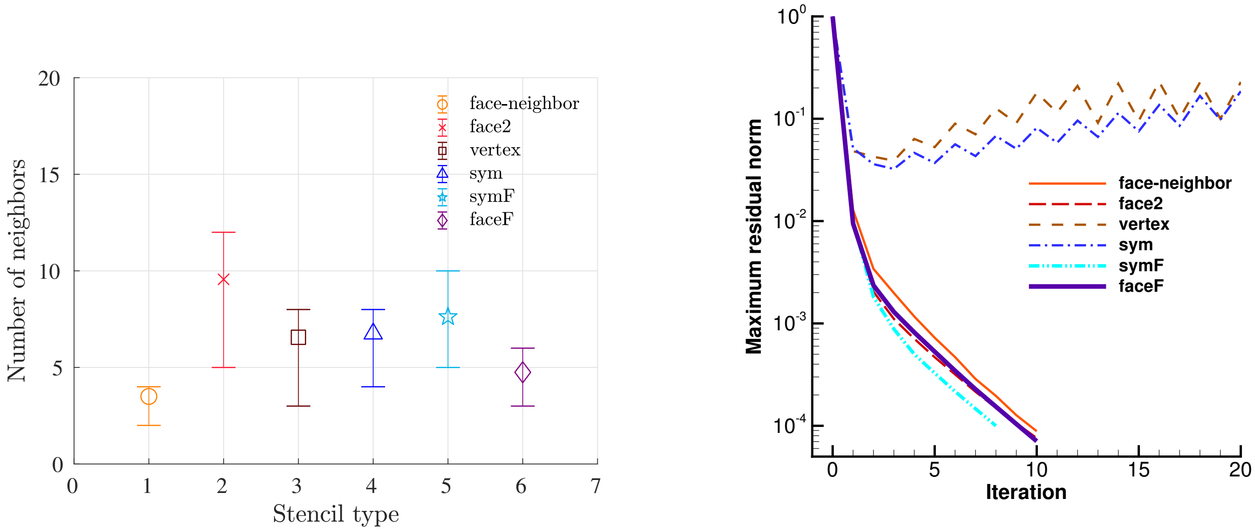


(a) Entire grid.

(b) Close view in scaled coordinates.

Figure 8: A curved quadrilateral grid.

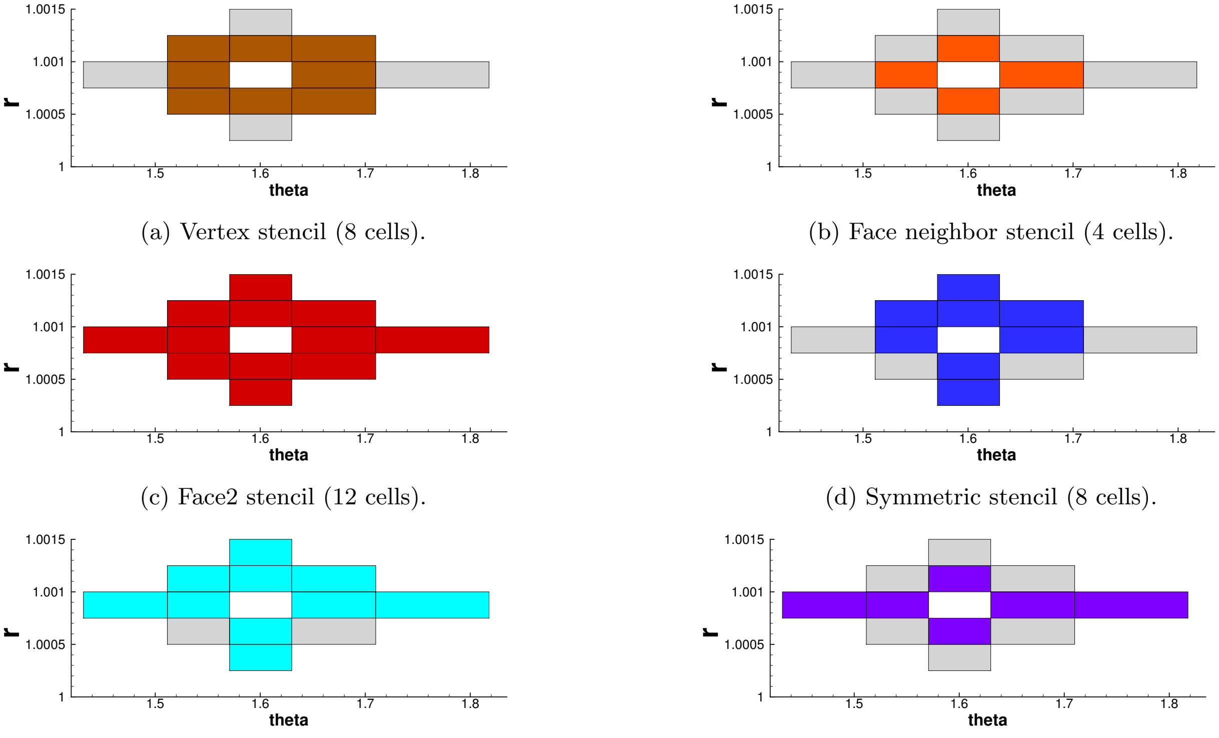
We begin with a quadrilateral grid with 81 nodes and 64 cells as shown in Figure 8. The grid spacing is uniform in the radial direction, but not in the circumferential direction, where the grid is gently stretched towards the vertical line at $x = 0$. Figure 9(a) shows the comparison of the number of neighbors for different



(a) Number of neighbors for various gradient stencils: 1=face-neighbor, 2=face2, 3=vertex, 4=sym, 5=symF, 6=faceF.

(b) Residual versus iteration.

Figure 9: Results for the curved quadrilateral grid case.



(a) Vertex stencil (8 cells).

(b) Face neighbor stencil (4 cells).

(c) Face2 stencil (12 cells).

(d) Symmetric stencil (8 cells).

(e) symF stencil (10 cells).

(f) faceF (6 cells).

Figure 10: Gradient stencils for the curved quadrilateral grid shown in the polar coordinates. Stencil members are colored and shown over the face2 stencil, and so all cells are colored in the face2 stencil.

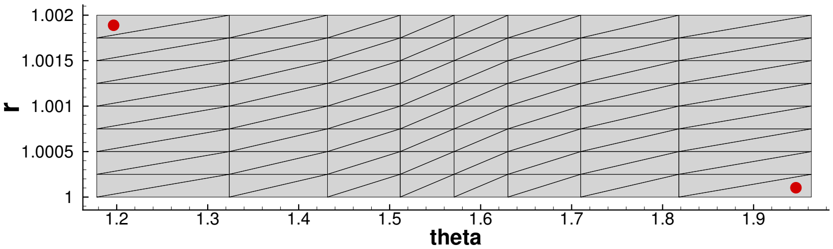


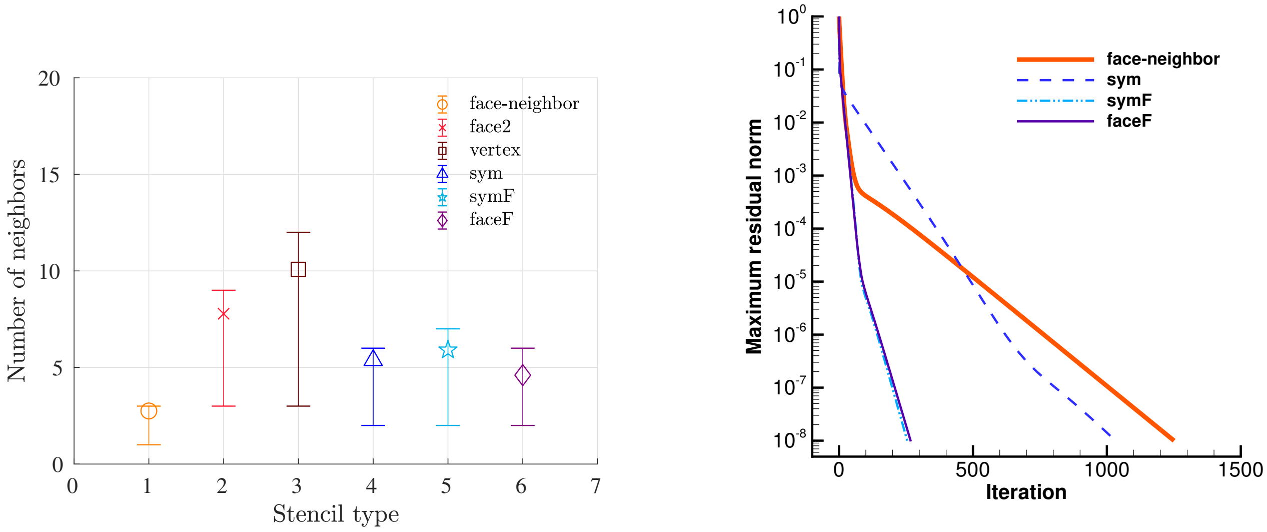
Figure 11: The triangular grid with regular connectivity shown in the polar coordinate system.

gradient stencils. For quadrilateral grids, the face2 stencil is the largest as is well known [6, 9]. The next largest is the *symF* stencil, and then the *sym*, vertex, faceF, and face-neighbor stencils, in order of decreasing size. It is important to note that faceF and *symF* have different sets of neighbors, and thus the F-decreasing augmentation depends on the initial stencil as expected. In fact, there is no reason that the F-decreasing augmented stencil must contain a symmetric stencil.

The convergence results are shown in Figure 9(b). The implicit solver diverged with the vertex and symmetric stencils, and converged with other stencils. The fact that it converges with the face neighbor stencil indicates that no stencil augmentation is needed for this problem. The results show, nevertheless, that the F-decreasing augmentation generated gradient stencils smaller than the largest and leads to stable iterative convergence in this particular case. To gain some insight, let us compare the gradient stencils. Figure 10 shows the gradient stencils taken from a representative interior cell located near the center of the grid, displayed in the polar coordinates (r, θ). Comparing the vertex stencil and the face neighbor stencil, we observe that four corner cells included in the vertex stencil introduce a large solution variation and seem to have destabilized the implicit solver (face 2 and *symF* also include the corner cells but have an additional feature as discussed further below). The same is observed for the symmetric stencil. Note that the symmetric augmentation picked up the two corner cells instead of the left and right most cells in the θ direction due to the curved nature of the grid combined with the non-uniform spacing in θ. The grid is rectangular in the (r, θ) space because the nodes have been generated as such, but the cell centroids are not necessarily mapped onto a rectangular configuration if the grid spacing in θ is not uniform. In fact, we found that a similar stencil was obtained even by applying the symmetric augmentation in the (r, θ) space. The fact that the face2 stencil leads to stable convergence indicates that the problem is overcome by adding these left and right most cells. This is exactly what has been done by the F-decreasing augmentation. See Figures 10(e) and 10(f).

Next, we consider a triangular grid generated from the quadrilateral grid by inserting diagonals in the same direction (i.e., between the bottom left node and the top right node in a quadrilateral). See Figure 11. The resulting grid has 128 triangles in total. Figure 12(a) shows the comparison of the number of neighbors for different gradient stencils. The vertex stencil is the largest stencil, followed by the face2, the *symF*, *sym*, faceF, and face-neighbor stencils, in order of decreasing size. For this grid, the discretization failed with the face-neighbor stencil due to ill-conditioned LSQ matrices at cells having only one neighbor cell. There exist two such cells at the upper left and lower right corners of the domain as indicated by the red filled circles in Figure 11. To avoid the problem, we added manually the midpoints of their boundary faces to the gradient stencils and use the exact solution to obtain the solution values at these points. The faceF stencil does not suffer as the F-decreasing augmentation added an extra cell to these face-neighbor stencils. In the *symF* stencil, the problem is already avoided by the symmetric stencil. Iterative convergence results are shown in Figure 12(b). The implicit solver converged with the *sym*, *symF*, faceF, and somewhat surprisingly with the face-neighbor stencil, but diverged with the vertex and face2 stencils. The results indicate that a larger number of neighbors does not necessarily stabilize the implicit solver, which presents a counter example to the studies in Refs.[7, 8]. On the other hand, the symmetric and F-decreasing augmentations have successfully added extra cells to stabilize the implicit solver.

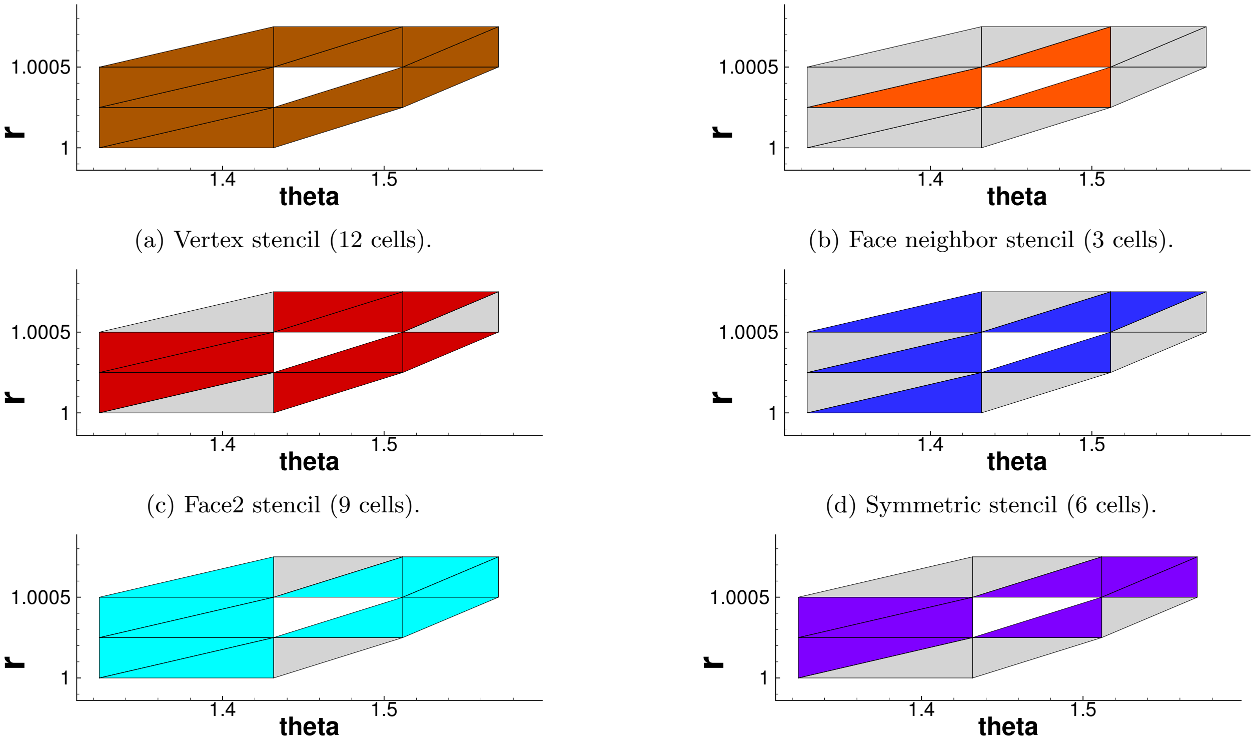
The convergence with the face-neighbor and symmetric stencils is, however, very slow compared with other stencils. Figure 13 shows the interior gradient stencils taken at a location close to the point $(1.46, 1.0004)$ in the grid. The success with the face-neighbor stencil may be due to its symmetric configuration: the top and bottom neighbors are aligned vertically, and the left and bottom ones are also nearly aligned in the θ direction. Note that due to the curvature and the non-uniform spacing in θ, again, the symmetric stencil is not precisely symmetric in the (r, θ) space. The *symF* and faceF are different, but both successfully stabilized the implicit solver.



(a) Number of neighbors for various gradient stencils: 1=face-neighbor, 2=face2, 3=vertex, 4=sym, 5=symF, 6=faceF.

(b) Residual versus iteration.

Figure 12: Results for the curved triangular grid case (regular connectivity).



(a) Vertex stencil (12 cells).

(b) Face neighbor stencil (3 cells).

(c) Face2 stencil (9 cells).

(d) Symmetric stencil (6 cells).

(e) symF stencil (10 cells).

(f) faceF (7 cells).

Figure 13: Gradient stencils for the curved triangular grid having regular connectivity, shown in the polar coordinates. Stencil members are colored within the vertex stencil, and so all cells are colored in the vertex stencil.

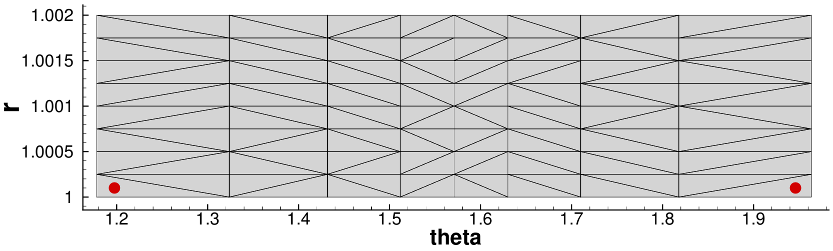
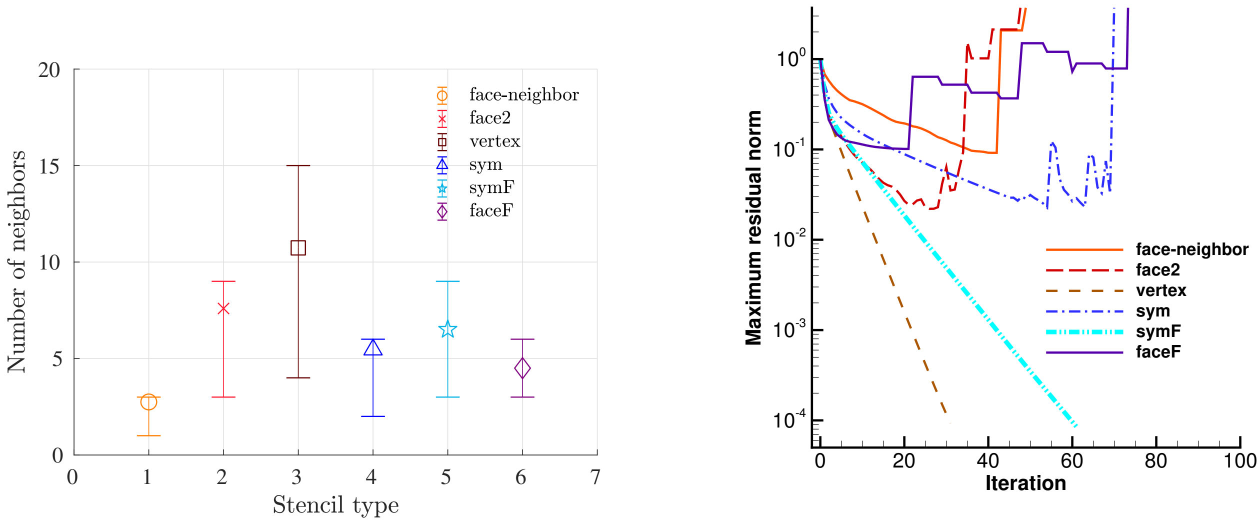


Figure 14: The triangular grid with irregular connectivity shown in the polar coordinate system.



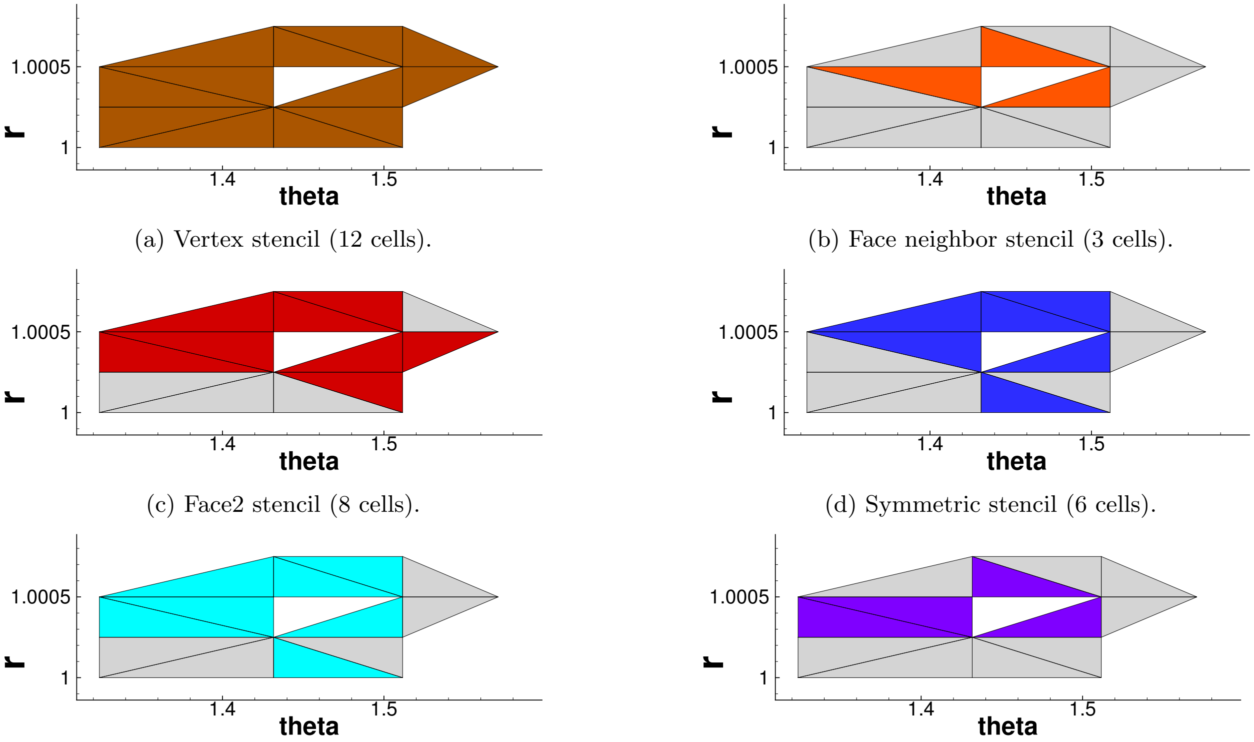
(a) Number of neighbors for various gradient stencils: 1=face, 2=face2, 3=vertex, 4=sym, 5=symF, 6=faceF.

(b) Residual versus iteration.

Figure 15: Results for the curved triangular grid case (irregular connectivity).

Finally, we consider an irregular triangular grid generated by inserting diagonals with random directions in the quadrilateral grid (see Figure 14). The grid lines in the radial direction are kept straight, which is a typical feature across a boundary layer in a practical unstructured grid. A striking difference from the previous regular triangulation is that the number of cells around a node is no longer bounded. As expected, random diagonal insertion has increased the size of the vertex stencil significantly as shown in Figure 15(a). The next largest is the face2 stencil, and then the *symF*, *sym*, faceF, and face-neighbor stencils, in order of decreasing size. In this grid, there exist two cells having only one neighbor at the lower left and lower right corners of the domain as indicated by the red filled circles in Figure 14; and we again employed the boundary face midpoints to avoid the ill-conditioning of the LSQ matrix for the face neighbor stencil. The faceF stencil has no problem since the F-decreasing augmentation added, again, an extra cell to these stencils (the extra cell is the one that appears in Figure 16(f) but not in Figure 16(b)). Figure 15(b) shows the iterative convergence results. The implicit solver diverged with the face, face2, faceF, and *sym* stencils, and converged with the vertex and *symF* stencils. As can be seen in Figure 16, where the interior gradient stencils are taken again at a location close to the point $(1.46, 1.0004)$ in the grid, the face-neighbor stencil does not have a symmetric configuration because of the random diagonal insertion. The *sym* stencil again does not precisely symmetrize the stencil in the (r, θ) space. The results demonstrate that as intended, the F-decreasing augmentation improved the *sym* stencil, and led to convergence with significantly fewer numbers of neighbors than the vertex stencil. As in the airfoil case, the faceF stencil fails to stabilize the implicit solver. It confirms again that the F-decreasing augmentation alone is not sufficient.

For the converged solutions, we compared the discretization and gradient errors, and found that the errors were very similar: the L_1 errors in the x-velocity component u are 5.552E-02 (vertex) and 5.967E-02 (*symF*),



(a) Vertex stencil (12 cells).

(b) Face neighbor stencil (3 cells).

(c) Face2 stencil (8 cells).

(d) Symmetric stencil (6 cells).

(e) symF stencil (7 cells).

(f) faceF (4 cells).

Figure 16: Gradient stencils for the curved triangular grid having irregular connectivity, shown in the polar coordinates. Stencil members are colored within the vertex stencil, and so all cells are colored in the vertex stencil.

the L_1 errors in the gradient are 4.057E+01 (vertex) and 3.952E+01 (*symF*) for $\partial_x u$, and 2.308E+02 (vertex) and 2.261E+02 (*symF*) for $\partial_y u$. Very similar error results have been obtained for other variables and for other types of grids. Further discussion and investigation on gradient accuracy are left as future work.

6 Concluding Remarks

We have discussed two gradient-stencil augmentation methods for robust implicit finite-volume-solver convergence: symmetric and F-decreasing augmentations. The symmetric augmentation constructs a gradient stencil with a symmetric property similar to a face-neighbor stencil in a quadrilateral grid. It is an attempt to develop an efficient remedy to address the stability issue in a cell-centered finite-volume scheme with the face-neighbor gradient stencil [7]. The F-decreasing augmentation adds extra cells to an existing stencil that decrease the reciprocal of the Frobenius norm of a scaled LSQ matrix, which determines the lower bound of the gradient magnitude. It is based on the observation that implicit finite-volume solvers are more stable with smaller gradients. Numerical experiments demonstrate that the symmetric stencil indeed helps the implicit solver converge on a highly irregular triangular grid with a significantly fewer number of neighbors than the vertex stencil. For curved grids with various element types: quadrilateral, regular triangles, and random triangles, the symmetric stencil is not sufficient, and the F-decreasing augmentation has been found critically important for stabilizing the implicit solver. Other conventional stencils are not entirely successful for these curved grids, leading to divergence. Only the *symF* stencil, which is constructed by the F-decreasing augmentation applied to the symmetric stencil, successfully stabilized the implicit solver for all the test cases. It is noted, however, that all the conclusions are not based on a rigorous mathematical proof that guarantees stability. This paper was intended to shed light on efficient gradient stencil construction and propose practical guides for a stencil augmentation.

For a practical unstructured grid, the study suggests that one would employ the symmetric and F-decreasing augmentation near viscous surfaces, and the symmetric augmentation elsewhere. Ideally, we would like to switch from the latter to the former when a strong curvature effect is detected, but developing a reliable geometry quantity to determine a highly-curved region for general three-dimensional unstructured grids remains an open area of research. Although the proposed techniques have been shown to bring significant improvements, the resulting gradient stencils are not necessarily optimal, i.e., not the smallest stencil that stabilizes the implicit solver. To attempt such an optimization, a technique would need to be developed that removes cells from a stencil without losing robust convergence. Another topic left for future work is the investigation of grid-resolution effects, i.e., whether the stability behavior changes on a finer grid where the gradient is appropriately resolved.

The proposed augmentation techniques are independent of the grid connectivity unlike other popular stencils (e.g., face2 and vertex), and can be directly applied to a general point cloud in two and three dimensions. These techniques may find their applications in high-order reconstruction-based finite-volume methods [22] and second/third-order node-centered edge-based methods [23] for generating efficient gradient stencils. Finally, it is emphasized that the present study has focused on only one of many algorithmic aspects affecting implicit solver stability. Existing techniques such as an approximate mapping method [18], mesh optimizations [8], and Jacobian-Free Newton Krylov methods [5], continue to be useful, and they may turn out to be even more effective when combined with these improved gradient stencils.

Acknowledgments

This work was supported by the Hypersonic Technology Project, through the Hypersonic Airbreathing Propulsion Branch of the NASA Langley Research Center, under Contract No. 80LARC17C0004. The author would like to thank Jeffery A. White (NASA Langley Research Center) for valuable comments and discussions.

References

[1] FUN3D online manual. `http://fun3d.larc.nasa.gov`.

[2] Description of the DLR TAU Code. `http://tau.dlr.de/code-description/`.

[3] scFlow, Software Cradle, MSC Software Company. `http://www.cradle-cfd.com/products/scflow`, visited on 10/30/2018.

[4] Y. Nakashima, N. Watanabe, and H. Nishikawa. Development of an effective implicit solver for general-purpose unstructured CFD software. In *The 28th Computational Fluid Dynamics Symposium*, C08-1, Tokyo, Japan, 2014.

[5] Mohagna J. Pandya, Boris Diskin, James L. Thomas, and Neal T. Frink. Improved convergence and robustness of USM3D solutions on mixed element grids. *AIAA J.*, 54(9):2589–2610, September 2016.

[6] Jeffery A. White, Robert Baurle, Bradley J. Passe, Seth C. Spiegel, and Hiroaki Nishikawa. Geometrically flexible and efficient flow analysis of high speed vehicles via domain decomposition, part 1, unstructured-grid solver for high speed flows. In *JANNAF 48th Combustion 36th Airbreathing Propulsion, 36th Exhaust Plume and Signatures, 30th Propulsion Systems Hazards, Joint Subcommittee Meeting, Programmatic and Industrial Base Meeting*, Newport News, VA, 2017.

[7] F. Haider, J.-P. Croisille, and B. Courbet. Stability analysis of the cell centered finite-volume MUSCL method on unstructured grids. *Numer. Math.*, 113(4):555–600, 2009.

[8] Reza Zangeneh and Carl Ollivier-Gooch. Mesh optimization to improve the stability of finite-volume methods on unstructured meshes. *Comput. Fluids*, 156:590–601, 2017.

[9] Chandan B. Sejekan and Carl F. Ollivier-Gooch. Improving finite-volume diffusive fluxes through better reconstruction. *Comput. Fluids*, 139:216–232, 2016.

[10] Andrew W. Cary, Andrew J. Dorgan, and Mori Mani. Towards accurate flow predictions using unstructured meshes. In *Proc. of 39th AIAA Fluid Dynamics Conference*, AIAA Paper 2009-3650, San Antonio, 2009.

[11] Hiroaki Nishikawa, Boris Diskin, James L. Thomas, and D. H. Hammond. Recent advances in agglomerated multigrid. In *51st AIAA Aerospace Sciences Meeting*, AIAA Paper 2013-863, Grapevine, Texas, 2013.

[12] A. Schwöppe and B. Diskin. Accuracy of the cell-centered grid metric in the DLR TAU-code. In A. Dillmann, G. Heller, H. P. Kreplin, W. Nitsche W., and I. Peltzer, editors, *New Results in Numerical and Experimental Fluid Mechanics VIII. Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Volume 121*, pages 429–437. Springer, 2013.

[13] Emre Sozer, Christoph Brehm, and Cetin C. Kiris. Gradient calculation methods on arbitrary polyhedral unstructured meshes for cell-centered cfd solvers. In *Proc. of 52nd AIAA Aerospace Sciences Meeting*, AIAA Paper 2014-1440, National Harbor, Maryland, 2014.

[14] M. Xionga, X. Denga, X. Gaoa, Y. Dongb, C. Xua, and Z. Wanga. A novel stencil selection method for the gradient reconstruction on unstructured grid based on OpenFOAM. *Comput. Fluids*, 172:426–442, 2018.

[15] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43:357–372, 1981.

[16] Gilbert Strang. *Linear Algebra and Its Applications.* Academic Press, second edition, 1980.

[17] A. Haselbacher. On constrained reconstruction operators. In *Proc. of 44th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2006-1274, Reno, Nevada, 2006.

[18] B. Diskin and J. L. Thomas. Comparison of node-centered and cell-centered unstructured finite-volume discretizations: Inviscid fluxes. *AIAA J.*, 49(4):836–854, 2011.

[19] D. J. Mavriplis. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. In *Proc. of 16th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2003-3986, Orlando, Florida, 2003.

[20] B. Diskin and J. L. Thomas. Accuracy of gradient reconstruction on grids with high aspect ratio. *NIA Report No. 2008-12*, 2008.

[21] E. Shima, K. Kitamura, and T. Haga. Green-Gauss/weighted-least-squares hybrid gradient reconstruction for arbitrary polyhedra unstructured grids. *AIAA J.*, 51(11):2740–2747, 2013.

[22] Alireza Jalali and Carl Ollivier-Gooch. Higher-order unstructured finite volume RANS solution of turbulent compressible flows. *Comput. Fluids*, 143:32–47, 2017.

[23] H. Nishikawa and Y. Liu. Accuracy-preserving source term quadrature for third-order edge-based discretization. *J. Comput. Phys.*, 344:595–622, 2017.