AIAA 2025-0074, AIAA SCITECH 2025 Forum

A Numerically Verified Parameter-free Third Order Accurate Reconstruction Method for Cell-Centered Finite Volume Solvers on Arbitrarily Unstructured Grids

Shuang Z. Tu* and Chao Jiang†
Jackson State University, Jackson, MS, 39217, USA

This paper presents the implementation of a parameter-free third-order reconstruction method for cell-centered finite volume solvers on unstructured grids. The reconstruction is based on nodal gradients obtained using the least squares approach from solutions at adjacent cell centers. The cell and face gradients are computed by simple arithmetic averaging of vertex gradients, while the face values are obtained through quadratic interpolation. Importantly, the current reconstruction method does not require explicit second derivatives, and its stencil remains as compact as that used in traditional linear reconstruction methods. The third-order accuracy of the left and right states at the face values, along with the second-order accuracy of the face gradients, is numerically verified on various unstructured grids. This verified third-order accuracy is a crucial condition for ensuring the overall accuracy of the finite volume solver.

I. Introduction

The finite volume method (FVM) in Computational Fluid Dynamics (CFD) emphasizes the conservation of physical quantities by integrating the governing partial differential equations over discrete control volumes. The integral form of the conservation laws ensures that fluxes entering and exiting each control volume are accurately accounted for. In a cell-centered finite volume method, the unknowns are stored at the centers of the cells, and only the cell averages at these centers are directly solved for. To update these cell averages, it is necessary to evaluate the advective and diffusive fluxes between adjacent cells (cf. Fig. 1). In FVM, a numerical advective flux function requires the left and right states of the solution at the cell interface as inputs, while a diffusive flux function needs the solution gradient at the interface. Since only the cell averages are directly available from the solution process, the left and right states and the solution gradient at the interface must be "reconstructed" using the known cell-center solutions and the underlying mesh geometry.

To achieve sufficient accuracy for practical applications, reconstruction must be implemented carefully to ensure that the solutions at cell interfaces reach at least second-order accuracy (i.e., linear reconstruction, as commonly used in most commercial CFD packages). To obtain accuracy higher than second-order, three primary approaches are typically used in the reconstruction process. The first is the MUSCL (Monotonic Upstream-centered Scheme for Conservation Laws) method. The second involves constructing high-degree polynomial fits to interpolate the solution across each

*Professor, AIAA Associate Fellow, (shuang.z.tu@jsums.edu).

†PhD candidate, (j00744382@students.jsums.edu).

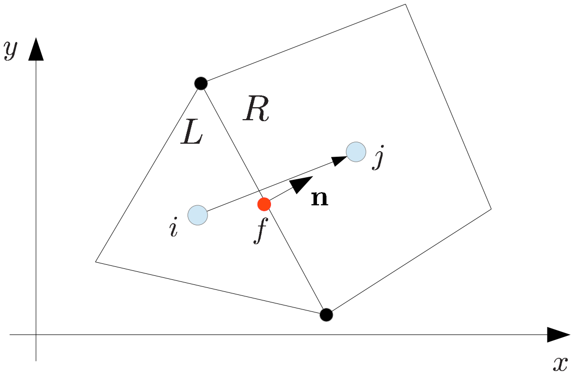


Figure 1: Flux calculation in a cell-centered finite volume solver.

control volume. The third approach utilizes the WENO (Weighted Essentially Non-Oscillatory) method. For structured meshes, all of these methods can achieve third-order accuracy relatively easily. However, many practical CFD solvers employ unstructured meshes to discretize complex computational domains, and achieving consistent third-order accuracy on arbitrary grids is not a trivial task.

The MUSCL approach uses a parameter to achieve third-order accuracy. A detailed study of the U-MUSCL scheme, first introduced in,[1] was conducted by Padway and Nishikawa.[7] Their research concluded that the U-MUSCL scheme can achieve third-order accuracy for linear problems on regular grids. However, for unstructured grids and/or nonlinear problems, the U-MUSCL scheme cannot maintain third-order accuracy. The approach based on high-degree polynomials, such as the k-exact method,[6] typically requires a larger stencil compared to linear reconstruction in order to achieve higher than second-order accuracy. The WENO approach, on the other hand, combines multiple potential reconstruction stencils weighted by smoothness indicators,[3,4] which also results in a non-compact stencil. In practice, the k-exact reconstruction is often combined with the WENO method to enhance stability near discontinuities.

In the author's view, a successful third-order reconstruction strategy must fulfill the following two fundamental criteria:

- **Suitability for Arbitrary Grids**: The strategy should ideally be parameter-free and applicable to any grid type.
- **Compact Reconstruction Stencil**: The stencil should only involve immediate neighboring cells, which is crucial for parallel computing based on mesh partitioning.

To ensure global third-order accuracy in a cell-centered finite volume (FV) solver for advection-diffusion-type equations, such as the Navier-Stokes equations, it is essential that the reconstruction meets the following conditions:

- The left and right states of the solution at the cell interfaces must be third-order accurate.
- The gradient of the solution at the cell interfaces must be second-order accurate.

This paper presents our implementation of a parameter-free third-order accurate reconstruction for multi-dimensional, arbitrarily unstructured grids. Our approach closely follows the reconstruction strategy for 1-D grids, which has been verified to achieve third-order accuracy. We focus on numerically verifying the claimed order of accuracy for reconstructing cell-interface values on arbitrarily unstructured meshes.

This paper is organized as follows. Section II provides a review of the reconstruction method for 1-D grids, offering three interpretations of the same reconstruction formulation. Section III describes various reconstruction strategies for multi-dimensional grids that closely mirror the 1-D methods. In Section IV, numerical tests are carried out to reveal a truly third order accurate reconstruction on arbitrarily unstructured grids. Finally, section V offers conclusions and further discussion.

II. Review of Third Oder Reconstruction in 1-D Grids

We begin by reviewing the reconstruction process in one dimension. For a cell-centered finite volume solver to achieve overall third-order accuracy, it is essential that the solution at the cell interface be reconstructed to third-order accuracy. Starting with the known solution at the cell centers, the reconstructed solution gradient at the cell centers, the solution at the face centers, and the solution gradient at the face centers are the key quantities used to assess the order of accuracy.

Considering the 1-D reconstruction stencil as shown in Fig. 2. As can be seen, this is the most compact 3-point stencil. The whole-number indices represent cell centers and the half-number indices represent vertices (or faces in 1-D). Assuming the grid is evenly spaced with $\Delta x = x_{i+1} - x_i$. Given the solutions at the cell centers, i.e. u_{i-1}, u_i and u_{i+1}, the goal is to reconstruct the solution gradient at cell i.

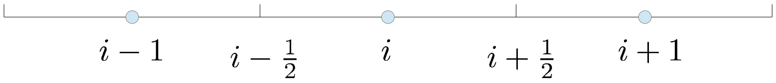


Figure 2: Reconstruction stencil in 1-D. Cell centers are represented by circles.

- *Method 1*: use central differencing to obtain $(u_x)_i$ directly, namely,

$$(u_x)_i = \frac{u_{i+1} - u_{i-1}}{2\Delta x} \tag{1}$$

which is clearly a second-order accurate approximation of the first derivative.

- *Method 2*: first use simple linear interpolation to find the solution at vertices (or faces for 1-D grids) as follows

$$u_{i-1/2} = \frac{1}{2}(u_{i-1} + u_i) \text{ and } u_{i+1/2} = \frac{1}{2}(u_i + u_{i+1}) \tag{2}$$

then $(u_x)_i$ can be obtained via the following central differencing

$$(u_x)_i = \frac{u_{i+1/2} - u_{i-1/2}}{\Delta x} = \frac{\frac{1}{2}(u_i + u_{i+1}) - \frac{1}{2}(u_{i-1} + u_i)}{\Delta x} = \frac{u_{i+1} - u_{i-1}}{2\Delta x} \tag{3}$$

which is the same as the second-order approximation (1).

- *Method 3*: first use central differencing to obtain the solution gradient at vertices (aka faces for 1-D grids) as follows

$$(u_x)_{i-1/2} = \frac{u_i - u_{i-1}}{\Delta x} \text{ and } (u_x)_{i+1/2} = \frac{u_{i+1} - u_i}{\Delta x} \tag{4}$$

then $(u_x)_i$ can be obtained via linear interpolation of solution gradients

$$(u_x)_i = \frac{1}{2}\left((u_x)_{i-1/2} + (u_x)_{i+1/2}\right) = \frac{1}{2}\left(\frac{u_i - u_{i-1}}{\Delta x} + \frac{u_{i+1} - u_i}{\Delta x}\right) = \frac{u_{i+1} - u_{i-1}}{2\Delta x} \tag{5}$$

which is also the same as the second-order approximation (1).

As observed, all three methods lead to the same second order accurate approximation for the solution gradient at the cell center (cf. Eqs. (1), (3) and (5)). While it may seem trivial to use Eq. (3) or Eq. (5) instead of Eq. (1) in the calculation of the solution gradient at the cell center, this observation offers valuable insight. It serves as an important foundation for calculating similar quantities on an unstructured multi-dimensional grid, as will be discussed in Sec. III.

Now we proceed to compute the left state of the solution at face $i + \frac{1}{2}$. The most commonly used approach is to use the linear reconstruction via

$$u^L_{i+1/2} = u_i + (u_x)_i \left(\frac{\Delta x}{2}\right) = u_i + \frac{u_{i+1} - u_{i-1}}{2\Delta x}\left(\frac{\Delta x}{2}\right) = u_i + \frac{1}{4}(u_{i+1} - u_{i-1}) \tag{6}$$

Without increasing the size of the stencil, one can achieve a third-order reconstruction as follows. First the second derivative of the solution at cell i is approximated using the solution gradients at vertices (4) via

$$(u_{xx})_i = \frac{(u_x)_{i+1/2} - (u_x)_{i-1/2}}{\Delta x} = \frac{\frac{u_{i+1}-u_i}{\Delta x} - \frac{u_i - u_{i-1}}{\Delta x}}{\Delta x} = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2}$$

then the solution at the face can be computed using quadratic Taylor expansions via

$$\begin{aligned} u^L_{i+1/2} &= u_i + (u_x)_i\frac{\Delta x}{2} + \frac{1}{2}(u_{xx})_i\left(\frac{\Delta x}{2}\right)^2 = u_i + \frac{u_{i+1} - u_{i-1}}{2\Delta x}\left(\frac{\Delta x}{2}\right) + \frac{1}{2}\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2}\left(\frac{\Delta x}{2}\right)^2 \\ &= u_i + \frac{1}{4}(u_{i+1} - u_{i-1}) + \frac{u_{i+1} - 2u_i + u_{i-1}}{8} = \frac{1}{8}(3u_{i+1} + 6u_i - u_{i-1}) \end{aligned} \tag{7}$$

which happens to be the QUICK scheme[2] that is a third-order accurate approximation.

Note that Eq. (7) can also be obtained via

$$u^L_{i+1/2} = u_i + \frac{(u_x)_i + (u_x)_{i+1/2}}{2}\frac{\Delta x}{2} \tag{8}$$

where $(u_x)_i$ and $(u_x)_{i+1/2}$ are given by Eq. (5) and Eq. (4), respectively. The advantage of using Eq. (8) is that it does not require the explicit computation of the second derivative of the solution.

Though the 1-D test for verifying the order of accuracy of above reconstruction process is trivial. It is included here for completeness. We start with exact solutions at cell centers. The following analytical solution that describes a raised cosine function in the domain $x \in [-1, 1]$

$$u(x) = \begin{cases} 1 + \frac{1}{2}\left[1 + \cos\left(\frac{4\pi x}{3}\right)\right] & |x| \le 0.75 \\ 0 & \text{elsewhere} \end{cases} \tag{9}$$

is used as an example. A series of meshes obtained via isotropic subdivision are used in the convergence study process. The l_1 norm of the reconstruction error is computed as follows

$$\epsilon = \frac{1}{N}\sum_{i=1}^{N} |u^h_i - u^{\text{exact}}_i| \tag{10}$$

where N is the number of entities (cells or faces) across the entire mesh. The numerical order of accuracy is computed according to

$$\text{order} = \frac{\log(\epsilon_i/\epsilon_{i-1})}{\log(0.5)} \quad i = 1, 2, \cdots \tag{11}$$

where the subscript 'i' stands for the refinement level of the meshes with ϵ_0 representing the error from the coarsest mesh.

Table 1 shows the error norms and corresponding order of accuracy of the reconstructed solution gradient at cell centers, reconstructed gradient at faces and the reconstructed solution at faces. As can be seen, the quadratic reconstruction for the solution at faces using Eq. (7) achieves third-order accuracy.

Table 1: Accuracy of the solution reconstruction in 1-D grids.

Number	$(u_x)_c$ from Eq. (1) or (3) or (5)		$(u_x)_f$ from Eq. (4)		u_f^L from Eq. (7)	
of cells	error	order	error	order	error	order
6	4.392568e-01	-	8.811850e-02	-	5.087634e-02	-
12	1.113321e-01	1.98	2.442422e-02	1.85	6.364477e-03	3.00
24	2.915719e-02	1.93	6.145001e-03	1.99	8.214035e-04	2.95
48	7.295913e-03	2.00	1.566612e-03	1.97	1.033307e-04	2.99
96	1.824393e-03	2.00	3.944622e-04	1.99	1.296400e-05	2.99
192	4.561242e-04	2.00	9.890699e-05	2.00	1.623628e-06	3.00

III. Reconstruction in Multi-dimensions

We aim to adapt the three methods (cf. Eqs. (1), (3) and (5)) used for 1-D grids to construct the solution gradient at cell centers on multi-dimensional grids. The reconstruction procedure begins with the solutions at the cell centers, i.e. u_c.

It can be shown that the least squares method is equivalent to central differencing on structured grids where the grid lines align with the coordinate axes. However, for unstructured grids, standard central differencing cannot be applied. In such cases, the least squares method is used whenever central differencing needs to be approximated.

III.A. Gradient Reconstruction at Cell Centers in Multi-dimensions

III.A.1. Method 1*: Least Square Construction Using the Solutions at Cell Centers*

In this method, to mimic Eq. (1), the solutions gradient at the center of a cell center is obtained via the least square procedure by minimizing the following objective function

$$S = \sum_{i=1}^{n_f} \left(u_c + \nabla u_c \cdot \Delta \mathbf{x}_{c,i} - u_{c,i}\right)^2 \tag{12}$$

where n_f is the number of faces this cell has. $u_{c,i}$ is the solution at the ith neighboring cell.

Another slightly different version is to minimize the objective function

$$S = \sum_{i=1}^{n_f} \left(\nabla u_c \cdot \mathbf{n}_{c,i} - \left(\frac{u_{c,i} - u_c}{||\Delta \mathbf{x}||_{c,i}}\right)\right)^2 = \sum_{i=1}^{n_f} \frac{1}{||\Delta \mathbf{x}||^2_{c,i}} \left(u_c + \nabla u_c \cdot \Delta \mathbf{x}_{c,i} - u_{c,i}\right)^2 . \tag{13}$$

Here $\Delta \mathbf{x}_{c,i} = \mathbf{x}_{c,i} - \mathbf{x}_c$ is the displacement vector connecting the centroids of the two neighboring cells, and $\mathbf{n}_{c,i} = \Delta \mathbf{x}_{c,i} / ||\Delta \mathbf{x}_{c,i}||$. The latter can be considered as an inverse distance weighted least square method. For unstructured grids, it can be shown that the inverse distance weighted least square method yields more accurate results.

For example, minimizing (13) leads to the following system for a 2-D mesh

$$\begin{bmatrix} \sum_i^{n_f} \frac{\Delta x_i^2}{||\Delta \mathbf{x}||_{c,i}} & \sum_i^{n_f} \frac{\Delta x_i \Delta y_i}{||\Delta \mathbf{x}||_{c,i}} \\ \sum_i^{n_f} \frac{\Delta x_i \Delta y_i}{||\Delta \mathbf{x}||_{c,i}} & \sum_i^{n_f} \frac{\Delta y_i^2}{||\Delta \mathbf{x}||_{c,i}} \end{bmatrix} \begin{bmatrix} (u_x)_c \\ (u_y)_c \end{bmatrix} = \begin{bmatrix} \sum_i^{n_f} \frac{\Delta x_i}{||\Delta \mathbf{x}||_{c,i}} (u_{c,i} - u_c) \\ \sum_i^{n_f} \frac{\Delta y_i}{||\Delta \mathbf{x}||_{c,i}} (u_{c,i} - u_c) \end{bmatrix} \tag{14}$$

III.A.2. Method *2: Least Square Construction Using the Solutions at Vertices*

Since the solution is known only at the cell centers, an interpolation method is required to estimate the solution at the mesh vertices. In this approach, we obtain the solution at a vertex by using the inverse distance weighted average of the solutions at its neighboring cell centers, i.e.

$$u_v = \frac{\sum_{i=1}^{n_c} \frac{u_{c,i}}{r_i}}{\sum_{i=1}^{n_c} \frac{1}{r_i}} \tag{15}$$

where $r_i = |\mathbf{x}_v - \mathbf{x}_{c,i}|$ and n_c is the number of surrounding cells of the vertex under consideration. $\mathbf{x}_{c,i}$ and $u_{c,i}$ are the location and solution at the ith neighboring cell, respectively. Such averaging ensures that u_v will not exceed the range defined by solutions at its surrounding cell centers and is second-order accurate. In our earlier work,[9] we stated that inverse distance weighted interpolation is the most robust method when the vertex lies outside the convex hull formed by its surrounding cell centers, which is a possible situation when high aspect ratio cells are used near the walls with large local surface curvature.

To mimic Eq. (3), the least square method is then used to minimize the following inverse distance weighted objective function

$$S = \sum_{i=1}^{n_v} \left(\nabla u_c \cdot \mathbf{n}_{v,i} - \left(\frac{u_{v,i} - u_c}{||\Delta \mathbf{x}||_{v,i}} \right) \right)^2 = \sum_{i=1}^{n_v} \frac{1}{||\Delta \mathbf{x}||_{v,i}^2} \left(u_c + \nabla u_c \cdot \Delta \mathbf{x}_{v,i} - u_{v,i} \right)^2 \tag{16}$$

over each cell. Here $\Delta \mathbf{x}_{v,i} = \mathbf{x}_{v,i} - \mathbf{x}_c$, $\mathbf{n}_{v,i} = \Delta \mathbf{x}_{v,i} / ||\Delta \mathbf{x}_{v,i}||$ and n_v is the number of vertices forming the cell. The least square method needs to solve an nsd × nsd system for each variable. nsd is the number of spatial dimensions. For example, minimizing (16) leads to the following system for a 2-D mesh

$$\begin{bmatrix} \sum_i^{n_v} \frac{\Delta x_i^2}{||\Delta \mathbf{x}||_{v,i}} & \sum_i^{n_f} \frac{\Delta x_i \Delta y_i}{||\Delta \mathbf{x}||_{v,i}} \\ \sum_i^{n_v} \frac{\Delta x_i \Delta y_i}{||\Delta \mathbf{x}||_{v,i}} & \sum_i^{n_f} \frac{\Delta y_i^2}{||\Delta \mathbf{x}||_{v,i}} \end{bmatrix} \begin{bmatrix} (u_x)_c \\ (u_y)_c \end{bmatrix} = \begin{bmatrix} \sum_i^{n_v} \frac{\Delta x_i}{||\Delta \mathbf{x}||_{v,i}} (u_{v,i} - u_c) \\ \sum_i^{n_v} \frac{\Delta y_i}{||\Delta \mathbf{x}||_{v,i}} (u_{v,i} - u_c) \end{bmatrix} \tag{17}$$

Alternatively, one can also use the Green-Gauss theorem to compute the solution gradient inside the cell as follows:

$$\nabla u_c = \frac{1}{|\Omega_c|} \sum_{i=1}^{n_f} u_{f,i} A_i \mathbf{n}_i \tag{18}$$

where n_f is the number of faces surrounding the cell under consideration and $u_{f,i}$ is the interpolated solution (simple arithmetic average of the solution at vertices) at the ith face. A_i and $\mathbf{n}_i$ are the area and the outward unit normal of the ith face, respectively. $|\Omega_c|$ is the volume of the cell under consideration. The Green-Gauss theorem based method is more efficient since it does not require to solve an equation system.

III.A.3. Method *3: Interpolation Using the Solution Gradient at Vertices*

In 1-D, the solution gradient at vertices can be straightforwardly obtained via Eq. (4). For multi-dimensional unstructured grids, the computation of the solution gradient at vertices is more involved. Since the most natural analogy to central differencing in multi-dimensions is the least squares method, the following objective function is minimized

$$S = \sum_{i=1}^{n_c} \left(\nabla u_v \cdot \mathbf{n}_{c,i} - \left(\frac{u_{c,i} - u_v}{||\Delta \mathbf{x}||_{c,i}} \right) \right)^2 = \sum_{i=1}^{n_v} \frac{1}{||\Delta \mathbf{x}||_{c,i}^2} \left(u_v + \nabla u_v \cdot \Delta \mathbf{x}_{c,i} - u_{c,i} \right)^2 \tag{19}$$

to find the solution gradient at each vertex. Here $\Delta \mathbf{x}_{c,i} = \mathbf{x}_{c,i} - \mathbf{x}_v$ and $\mathbf{n}_{c,i} = \Delta \mathbf{x}_{c,i} / ||\Delta \mathbf{x}_{c,i}||$. Considering a 2-D example, the least squares method results in the following equation system for the solution and its gradient at a vertex

$$\begin{bmatrix} \sum_i^{n_c} \frac{1}{||\Delta \mathbf{x}||_{c,i}} & \sum_i^{n_c} \frac{\Delta x_i}{||\Delta \mathbf{x}||_{c,i}} & \sum_i^{n_c} \frac{\Delta x_i}{||\Delta \mathbf{x}||_{c,i}} \\ \sum_i^{n_c} \frac{\Delta x_i}{||\Delta \mathbf{x}||_{c,i}} & \sum_i^{n_c} \frac{\Delta x_i^2}{||\Delta \mathbf{x}||_{c,i}} & \sum_i^{n_c} \frac{\Delta x_i \Delta y_i}{||\Delta \mathbf{x}||_{c,i}} \\ \sum_i^{n_c} \frac{\Delta y_i}{||\Delta \mathbf{x}||_{c,i}} & \sum_i^{n_c} \frac{\Delta x_i \Delta y_i}{||\Delta \mathbf{x}||_{c,i}} & \sum_i^{n_c} \frac{\Delta y_i^2}{||\Delta \mathbf{x}||_{c,i}} \end{bmatrix} \begin{bmatrix} u_v \\ (u_x)_v \\ (u_y)_v \end{bmatrix} = \begin{bmatrix} \sum_i^{n_c} \frac{1}{||\Delta \mathbf{x}||_{c,i}} u_{c,i} \\ \sum_i^{n_c} \frac{\Delta x_i}{||\Delta \mathbf{x}||_{c,i}} u_{c,i} \\ \sum_i^{n_c} \frac{\Delta y_i}{||\Delta \mathbf{x}||_{c,i}} u_{c,i} \end{bmatrix} \tag{20}$$

Once the solution gradients at vertices become available, the solution gradient at the center of a cell can be obtained via simple averaging, namely,

$$\nabla u_c = \frac{1}{n_v} \sum_{i=1}^{n_v} (\nabla u_v)_i \tag{21}$$

where n_v is the number of vertices forming this cell.

III.B. Gradient Reconstruction at Face Centers in Multi-dimensions

III.B.1. Method 1*: Based on the Solution Gradient at Vertices*

If the solution gradients at vertices are available via the least square approach (19), the solution gradient at a face center can be obtained via the following simple arithmetic averaging

$$\nabla u_f = \frac{1}{n_v} \sum_{i=1}^{n_v} (\nabla u_v)_i . \tag{22}$$

where n_v is the number of vertices forming this face.

III.B.2. Method 2*: Based on the Solution Gradient at Cell Centers*

If the solution gradients at vertices are not available, then the solution gradient at a face center can be obtained via

$$\nabla u_f = \frac{u_j - u_i}{\Delta \mathbf{r}_{ij} \cdot \mathbf{n}} \mathbf{n} + \left(\overline{\nabla u_f} - \frac{\overline{\nabla u_f} \cdot \Delta \mathbf{r}_{ij}}{\Delta \mathbf{r}_{ij} \cdot \mathbf{n}} \mathbf{n} \right) \tag{23}$$

or

$$\nabla u_f = \frac{u_j - u_i}{||\Delta \mathbf{r}_{ij}||} \mathbf{n}_{ij} + \left(\overline{\nabla u_f} - \frac{\overline{\nabla u_f} \cdot \Delta \mathbf{r}_{ij}}{||\Delta \mathbf{r}_{ij}||} \mathbf{n}_{ij} \right) \tag{24}$$

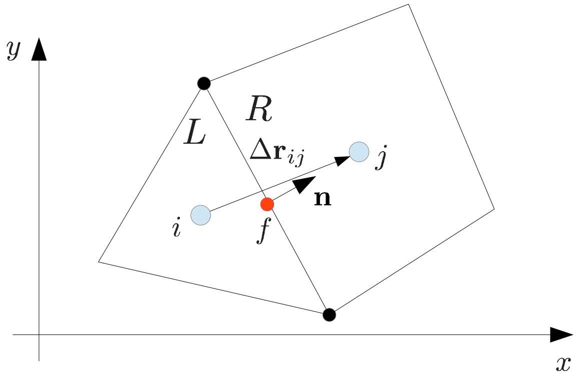


Figure 3: Stencil to compute the solution gradient at the face.

where

$$\overline{\nabla u_f} = \frac{1}{2}\left(\nabla u_i + \nabla u_j\right)$$

and $\Delta\mathbf{r}_{ij}$ is the displacement vector connecting cell centroids i and j, $\mathbf{n}$ is the unit normal of face k and $\mathbf{n}_{ij} = \Delta\mathbf{r}_{ij}/||\Delta\mathbf{r}_{ij}||$.

Note that in one-dimension or multi-dimensions with grid lines aligned with coordinate axes, the second term on the right hand side of Eq. (23) and Eq. (24) is zero, thus recovering Eq. (4).

III.C. Solution Reconstruction at Face Centers in Multi-dimensions

Once the solution gradients at cell centers are available, the solution at a face can be easily computed. In a typical data structure for unstructured meshes, a face connects to its two neighboring cells.

III.C.1. Linear Reconstruction

The left and right states of the solution at the face (cf. Fig. 1) can be calculated via

$$u_f^L = u_{c,i} + (\nabla u_c)_i \cdot (\mathbf{x}_f - \mathbf{x}_{c,i}), \text{ and} \tag{25a}$$

$$u_f^R = u_{c,j} + (\nabla u_c)_j \cdot (\mathbf{x}_f - \mathbf{x}_{c,j}) \tag{25b}$$

respectively. Here, $(\nabla u_c)_i$ and $(\nabla u_c)_j$ are the solution gradients at the ith and jth cell, respectively.

III.C.2. Quadratic Reconstruction

To derive the quadratic reconstruction formula, first note that the linear reconstruction Eq. (25a) can be rewritten as follows

$$u_f^L = u_{c,i} + (\nabla u_c)_i \cdot \left(\frac{\mathbf{x}_f - \mathbf{x}_{c,i}}{||\mathbf{x}_f - \mathbf{x}_{c,i}||}\right) ||\mathbf{x}_f - \mathbf{x}_{c,i}|| = u_{c,i} + \frac{\mathrm{d}u}{\mathrm{d}l}||\mathbf{x}_f - \mathbf{x}_{c,i}||$$

where

$$\frac{\mathrm{d}u}{\mathrm{d}l} = \nabla u_{c,i} \cdot \mathbf{n}_l \tag{26}$$

with

$$\mathbf{n}_l = \frac{\mathbf{x}_f - \mathbf{x}_{c,i}}{||\mathbf{x}_f - \mathbf{x}_{c,i}||}$$

being the unit direction vector of the line segment that starts at $\mathbf{x}_{c,i}$ and ends at $\mathbf{x}_f$. $\frac{\mathrm{d}u}{\mathrm{d}l}$ is simply the directional gradient along this line segment.

A quadratic reconstruction can be achieved via

$$u_f^L = u_{c,i} + \frac{\mathrm{d}u}{\mathrm{d}l}||\mathbf{x}_f - \mathbf{x}_{c,i}|| + \frac{1}{2}\frac{\mathrm{d}^2u}{\mathrm{d}l^2}||\mathbf{x}_f - \mathbf{x}_{c,i}||^2 \tag{27}$$

where the second derivative of the solution along the line segment can be obtained via the known solution gradient at the face, i.e.

$$\frac{\mathrm{d}u}{\mathrm{d}l} + \frac{\mathrm{d}^2u}{\mathrm{d}l^2}||\mathbf{x}_f - \mathbf{x}_{c,i}|| = \nabla u_f \cdot \mathbf{n}_l.$$

Simple algebraic manipulation leads to

$$\frac{\mathrm{d}^2u}{\mathrm{d}l^2} = \frac{(\nabla u_f - (\nabla u_c)_i) \cdot \mathbf{n}_l}{||\mathbf{x}_f - \mathbf{x}_{c,i}||} = \frac{(\nabla u_f - (\nabla u_c)_i) \cdot (\mathbf{x}_f - \mathbf{x}_{c,i})}{||\mathbf{x}_f - \mathbf{x}_{c,i}||^2} \tag{28}$$

which can be put back to Eq. (27) to yield the following form of the quadratic reconstruction

$$\begin{aligned} u_f^L &= u_{c,i} + (\nabla u_c)_i \cdot (\mathbf{x}_f - \mathbf{x}_{c,i}) + \frac{1}{2}(\nabla u_f - (\nabla u_c)_i) \cdot (\mathbf{x}_f - \mathbf{x}_{c,i}) \\ &= u_{c,i} + \frac{1}{2}(\nabla u_f + (\nabla u_c)_i) \cdot (\mathbf{x}_f - \mathbf{x}_{c,i}) \end{aligned} \tag{29}$$

The right state of the solution at the face can be computed similarly, i.e.

$$u_f^R = u_{c,j} + \frac{1}{2}(\nabla u_f + (\nabla u_c)_j) \cdot (\mathbf{x}_f - \mathbf{x}_{c,j}) \tag{30}$$

As can be seen in Eqs. (29) and (30), the quadratic reconstruction does not require the explicit knowledge about the second derivatives of the solution. In addition, Eqs. (29) and (30) are similar to Eq. (8) for 1-D grids.

IV. Numerical Tests

In this section, we conduct a series of tests to evaluate the reconstruction accuracy of various reconstruction strategies. Our focus is on the reconstruction of the solution gradient at cell centers, as well as the solution and its gradient at face centers. We assess the reconstruction error and the numerical order of convergence for each strategy on different types of meshes. The following three reconstruction strategies are tested, with each method designed to mimic the corresponding 1-D reconstruction discussed in Sec. II.

- *Method 1*:
  - Compute the solution gradient at a cell center via the least square approach utilizing the known solutions at immediate cell neighbors, i.e. via Eq. (14).
  - Compute the solution gradient at a face center using Eq. (24).
  - Compute the left state of the solution at a face center using Eq. (29).
- *Method 2*:
  - Obtain the solutions at vertices via inverse distance weighted averaging of the solutions at immediate cell neighbors, i.e. via Eq. (15).

  - Compute the solution gradient at a cell center according to either the least square approach (17) or the Green-Gauss approach (18) utilizing the available solutions at vertices.
  - Compute the solution gradient at a face center using Eq. (24).
  - Compute the left state of the solution at a face center using Eq. (29).
- *Method 3*:
  - Obtain the solution and its gradient at vertices via the least square approach (20).
  - Compute the solution gradient at a cell center using the simple average of the solution gradients at neighboring vertices, i.e. via (21).
  - Compute the solution gradient at a face center using the simple average of the solution gradients at neighboring vertices, i.e. via Eq. (22).
  - Compute the left state of the solution at a face center using Eq. (29).

Note that the reconstruction is implemented in parallel using the mesh partitioning technique (cf. Fig. 4) and the Message Passing Interface (MPI) library. Thanks to the compact stencil in the reconstruction, the inter-processor communication involves only nodes, faces and elements on the partition boundaries (cf. Fig. 5). This compactness makes it trivial to attain high parallelizability using MPI for fixed-topology meshes. Very efficient non-blocking MPI functions can be called to set up the inter-processor "gather" and "scatter" routines in the pre-processing stage.[8]

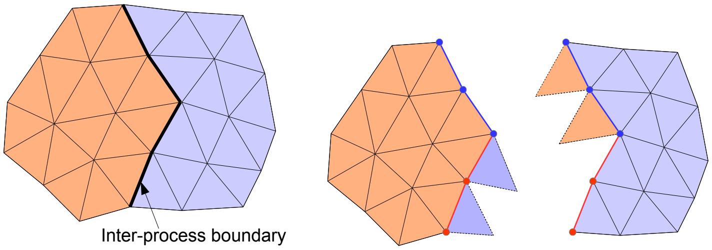


Figure 4: Mesh partitioning.

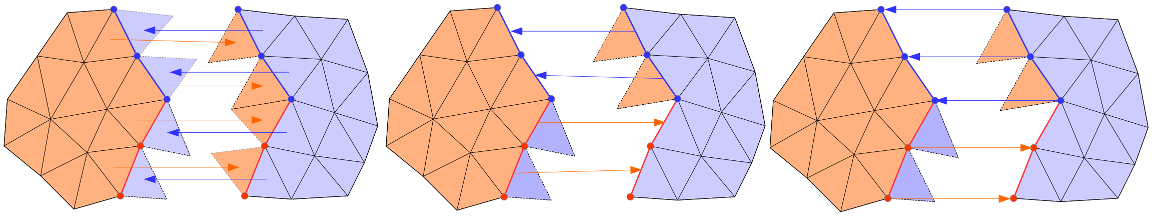


Figure 5: Inter-process communication. Left: element communication. Middle: face communication. Right: vertex communication.

IV.A. 2-D Tests

The following 2-D analytical field in the computational domain $[-1,1]\times[-1,1]$ is used to demonstrate the numerical order of accuracy of different reconstruction strategies on various meshes.

$$u(x,y)=\begin{cases}1+\frac{1}{4}\left[1+\cos\left(\frac{4\pi x}{3}\right)\right]\left[1+\cos\left(\frac{4\pi y}{3}\right)\right] & |x|\le 0.75,\text{ and }|y|\le 0.75\\ 0 & \text{elsewhere}\end{cases}\tag{31}$$

Figure 6 shows various meshes used in the tests. Each type of mesh is isotropically subdivided repeatedly to obtain a series of refined meshes. The l_1 reconstruction error is computed according to Eq. (10) and the numerical order of convergence is computed according to Eq. (11).

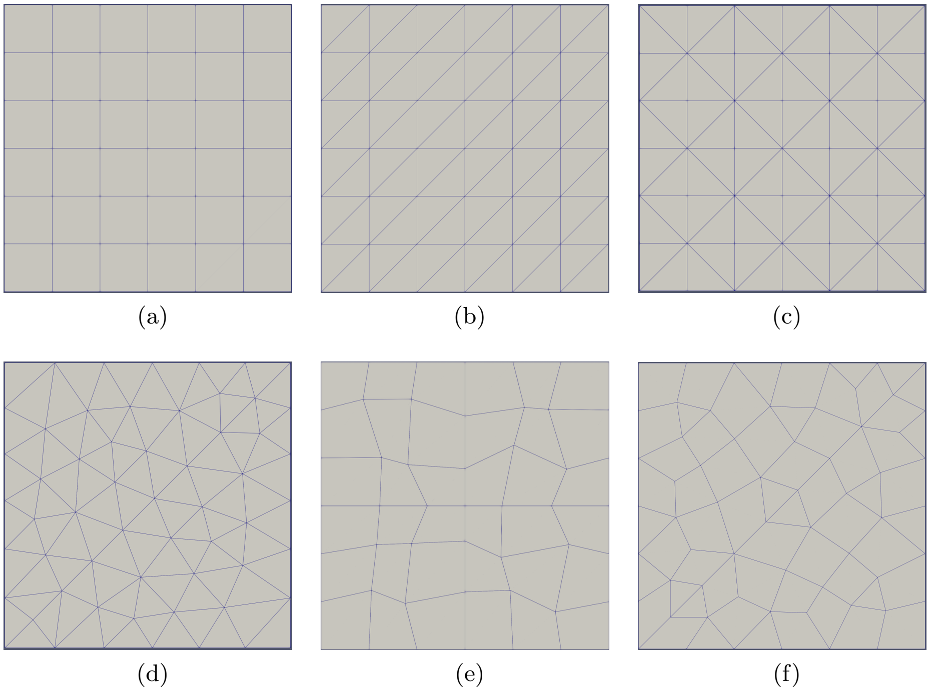


Figure 6: 2-D meshes used in the tests. (a) Rectangular mesh containing 36 cells (6 × 6). (b) Structured triangular mesh of type 1 containing 72 cells (tri6-type1). (c) Structured triangular mesh of type 2 containing 72 cells (tri6-type2). (d) Unstructured triangular mesh containing 90 cells (tri6-type3). (e) Perturbed quadrilateral mesh containing 36 cells (quad-6). (f) Hybrid quadrilateral/triangular mesh containing 40 quadrilaterals and 18 triangles.

Tables 2, 3 and 4 show the reconstruction error and order of accuracy for each of the three reconstruction methods on rectangular meshes (cf. Fig. 6a). As can be seen, all methods are second order accurate in reconstructing the solution gradient at cell centers and face centers, and third order accurate in reconstructing the solution at face centers. Since 2-D rectangular grids are tensor products of 1-D grids, the order of accuracy of each reconstruction method is expected to be the same as the 1-D counterpart.

Table 2: Accuracy of reconstruction on rectangular grids via Method 1.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
6 × 6	1.59E-01	-	9.43E-02	-	2.08E-02	-
12 × 12	5.14E-02	1.63	3.08E-02	1.62	2.96E-03	2.81
24 × 24	1.00E-02	2.36	8.12E-03	1.92	3.57E-04	3.05
48 × 48	2.59E-03	1.95	2.13E-03	1.93	4.51E-05	2.98
96 × 96	6.53E-04	1.99	5.41E-04	1.98	5.62E-06	3.00

Tables 5, 6 and 7. show the reconstruction order of accuracy for each of the three reconstruction

Table 3: Accuracy of reconstruction on rectangular grids via Method 2.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
6×6	1.91E-01	-	1.22E-01	-	2.40E-02	-
12×12	7.08E-02	1.43	4.10E-02	1.58	3.78E-03	2.67
24×24	1.52E-02	2.22	1.11E-02	1.89	4.60E-04	3.04
48×48	3.94E-03	1.94	2.92E-03	1.92	5.82E-05	2.98
96×96	9.94E-04	1.99	7.44E-04	1.97	7.25E-06	3.00

Table 4: Accuracy of reconstruction on rectangular grids via Method 3.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
6×6	1.91E-01	-	1.27E-01	-	3.00E-02	-
12×12	7.08E-02	1.43	4.17E-02	1.61	4.80E-03	2.64
24×24	1.52E-02	2.22	1.13E-02	1.89	5.91E-04	3.02
48×48	3.94E-03	1.94	2.94E-03	1.94	7.57E-05	2.97
96×96	9.94E-04	1.99	7.47E-04	1.98	9.43E-06	3.00

methods on the structured triangular meshes of type I (cf. Fig 6b). As can be seen, Methods 1 and 2 are both first order accurate in reconstructing the solution gradient at cell centers and second order accurate in reconstructing the solution gradient at face centers and second order accurate in reconstructing the solution at face centers. However, Method 3 is second order accurate in reconstructing the solution gradient at both cell centers and face centers and third order accurate in reconstructing the solution at face centers.

Table 5: Accuracy of reconstruction on structured triangular grids of type I via Method 1.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
tri6-type1	1.33E-01	-	7.57E-02	-	9.34E-03	-
tri12-type1	5.55E-02	1.26	2.62E-02	1.53	1.71E-03	2.45
tri24-type1	2.63E-02	1.08	6.23E-03	2.07	3.45E-04	2.31
tri48-type1	1.31E-02	1.01	1.65E-03	1.92	8.20E-05	2.07
tri96-type1	6.54E-03	1.00	4.24E-04	1.96	2.01E-05	2.03

Tables 8, 9 and 10. show the reconstruction order of accuracy for each of the three reconstruction methods on the structured triangular meshes of type II (cf. Fig 6c). As shown, Methods 1 and 2 both first order accurate in reconstructing the solution gradient at both cell centers and face centers and second order accurate in reconstructing the solution at face centers. However, Method 3 is second order accurate in reconstructing the solution gradient at both cell centers and face centers and third order accurate in reconstructing the solution at face centers.

Tables 11, 12 and 13. show the reconstruction order of accuracy for each of the three reconstruction methods on the unstructured triangular meshes (cf. Fig 6d). As can be seen, Method 1 is first

Table 6: Accuracy of reconstruction on structured triangular grids of type I via Method 2.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
tri6-type1	1.91E-01	-	1.08E-01	-	1.36E-02	-
tri12-type1	8.45E-02	1.18	3.96E-02	1.45	2.50E-03	2.44
tri24-type1	3.99E-02	1.08	9.87E-03	2.00	5.12E-04	2.29
tri48-type1	1.95E-02	1.03	2.63E-03	1.91	1.16E-04	2.14
tri96-type1	9.66E-03	1.01	6.77E-04	1.96	2.78E-05	2.06

Table 7: Accuracy of reconstruction on structured triangular grids of type I via Method 3.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
tri6-type1	1.91E-01	-	1.47E-01	-	2.08E-02	-
tri12-type1	5.93E-02	1.69	5.14E-02	1.51	3.31E-03	2.65
tri24-type1	1.45E-02	2.04	1.27E-02	2.01	4.06E-04	3.03
tri48-type1	3.81E-03	1.92	3.36E-03	1.92	5.34E-05	2.93
tri96-type1	9.73E-04	1.97	8.62E-04	1.96	6.84E-06	2.96

Table 8: Accuracy of reconstruction on structured triangular grids of type II via Method 1.

Method	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
tri6-type2	1.45E-01	-	1.09E-01	-	1.49E-02	-
tri12-type2	6.43E-02	1.17	5.32E-02	1.03	2.88E-03	2.37
tri24-type2	3.12E-02	1.04	2.36E-02	1.17	6.80E-04	2.08
tri48-type2	1.56E-02	1.00	1.14E-02	1.05	1.68E-04	2.02
tri96-type2	7.81E-03	1.00	5.56E-03	1.03	4.16E-05	2.01

Table 9: Accuracy of reconstruction on structured triangular grids of type II via Method 2.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
tri6-type2	2.19E-01	-	1.27E-01	-	1.87E-02	-
tri12-type2	1.03E-01	1.09	4.76E-02	1.41	3.35E-03	2.48
tri24-type2	5.16E-02	0.99	1.60E-02	1.58	6.92E-04	2.28
tri48-type2	2.57E-02	1.01	6.76E-03	1.24	1.57E-04	2.14
tri96-type2	1.28E-02	1.00	3.06E-03	1.14	3.77E-05	2.06

Table 10: Accuracy of reconstruction on structured triangular grids of type II via Method 3.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
tri6-type2	1.94E-01	-	1.48E-01	-	2.15E-02	-
tri12-type2	5.82E-02	1.74	4.99E-02	1.56	3.22E-03	2.74
tri24-type2	1.41E-02	2.05	1.24E-02	2.00	3.90E-04	3.05
tri48-type2	3.68E-03	1.93	3.27E-03	1.93	5.05E-05	2.95
tri96-type2	9.35E-04	1.98	8.34E-04	1.97	6.42E-06	2.97

order accurate in reconstructing the solution gradient at cell centers and second order accurate at reconstructing solution gradients at face centers. Method 2 is first order accurate in reconstructing the solution gradient at both cell centers and face centers. Both Method 1 and Method 2 yield second order accuracy in reconstructing the solution at face centers. However, Method 3 is second order accurate in reconstructing the solution gradient at both cell centers and face centers and third order accurate in reconstructing the solution at face centers.

Table 11: Accuracy of reconstruction on unstructured triangular grids via Method 1.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
tri6-type3	1.03E-01	-	6.79E-02	-	6.87E-03	-
tri12-type3	4.78E-02	1.11	2.10E-02	1.69	1.37E-03	2.33
tri24-type3	2.31E-02	1.05	5.79E-03	1.86	2.97E-04	2.20
tri48-type3	1.15E-02	1.01	1.68E-03	1.78	7.09E-05	2.07
tri96-type3	5.72E-03	1.01	4.56E-04	1.88	1.73E-05	2.03

Table 12: Accuracy of reconstruction on unstructured triangular grids via Method 2.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
tri6-type3	1.38E-01	-	8.12E-02	-	9.73E-03	-
tri12-type3	5.90E-02	1.23	2.70E-02	1.59	1.67E-03	2.55
tri24-type3	2.85E-02	1.05	1.01E-02	1.42	3.63E-04	2.20
tri48-type3	1.49E-02	0.94	4.65E-03	1.11	9.05E-05	2.00
tri96-type3	7.97E-03	0.90	2.17E-03	1.10	2.38E-05	1.93

Tables 14, 15 and 16. show the reconstruction order of accuracy for each of the three reconstruction methods on the perturbed quadrilateral meshes (cf. Fig 6e). As can be seen, Method 1 is second order accurate in reconstructing the solution gradient at both cell centers and face centers and third order accurate in reconstructing the solution at face centers. Method 2 tends to be first order accurate in reconstructing the solution gradient at both cell centers and face centers and second order accurate in reconstructing the solution at face centers. However, Method 3 is second order accurate in reconstructing the solution gradient at both cell centers and face centers and third order accurate in reconstructing the solution at face centers.

Table 13: Accuracy of reconstruction on unstructured triangular grids via Method 3.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
tri6-type3	1.21E-01	-	1.05E-01	-	1.26E-02	-
tri12-type3	3.83E-02	1.66	3.39E-02	1.63	1.90E-03	2.73
tri24-type3	1.02E-02	1.92	9.28E-03	1.87	2.40E-04	2.99
tri48-type3	2.76E-03	1.88	2.56E-03	1.86	3.33E-05	2.85
tri96-type3	7.42E-04	1.89	6.97E-04	1.88	4.51E-06	2.88

Table 14: Accuracy of reconstruction on perturbed quadrilateral grids via Method 1.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
quad-6	1.63E-01	-	1.10E-01	-	2.15E-02	-
quad-12	4.81E-02	1.76	3.69E-02	1.58	3.00E-03	2.84
quad-24	1.61E-02	1.58	1.25E-02	1.56	4.67E-04	2.68
quad-48	4.34E-03	1.89	3.56E-03	1.81	6.90E-05	2.76
quad-96	1.21E-03	1.85	9.73E-04	1.87	9.59E-06	2.85

Table 15: Accuracy of reconstruction on perturbed quadrilateral grids via Method 2.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
quad-6	2.24E-01	-	1.36E-01	-	2.71E-02	-
quad-12	7.50E-02	1.58	4.58E-02	1.57	4.36E-03	2.63
quad-24	3.12E-02	1.27	1.95E-02	1.23	8.09E-04	2.43
quad-48	1.20E-02	1.37	7.50E-03	1.38	1.47E-04	2.46
quad-96	5.15E-03	1.22	3.11E-03	1.27	2.89E-05	2.34

Table 16: Accuracy of reconstruction on perturbed quadrilateral grids via Method 3.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
quad-6	2.23E-01	-	1.44E-01	-	3.31E-02	-
quad-12	6.55E-02	1.77	4.46E-02	1.69	5.19E-03	2.67
quad-24	2.02E-02	1.69	1.44E-02	1.63	7.69E-04	2.75
quad-48	5.22E-03	1.96	4.01E-03	1.85	1.07E-04	2.84
quad-96	1.42E-03	1.88	1.08E-03	1.89	1.44E-05	2.89

Tables 17, 18 and 19. show the reconstruction order of accuracy for each of the three reconstruction methods on the hybrid quadrilateral/triangular meshes (cf. Fig 6f). As can be seen, the orders of accuracy are similar to those on the perturbed quadrilateral meshes.

Table 17: Accuracy of reconstruction on unstructured hybrid quadrilateral/triangular grids via Method 1.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
hybrid-6	1.24E-01	-	1.02E-01	-	1.39E-02	-
hybrid-12	4.13E-02	1.59	3.55E-02	1.52	2.44E-03	2.51
hybrid-24	1.32E-02	1.64	1.09E-02	1.71	3.63E-04	2.75
hybrid-48	4.49E-03	1.56	3.22E-03	1.75	5.35E-05	2.76
hybrid-96	1.52E-03	1.57	8.93E-04	1.85	7.77E-06	2.78

Table 18: Accuracy of reconstruction on unstructured hybrid quadrilateral/triangular grids via Method 2.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
hybrid-6	1.18E-01	-	9.93E-02	-	1.25E-02	-
hybrid-12	4.56E-02	1.37	3.71E-02	1.42	2.37E-03	2.40
hybrid-24	2.02E-02	1.18	1.52E-02	1.29	4.52E-04	2.39
hybrid-48	9.79E-03	1.04	6.52E-03	1.22	9.44E-05	2.26
hybrid-96	4.78E-03	1.03	2.91E-03	1.16	2.05E-05	2.21

Table 19: Accuracy of reconstruction on unstructured hybrid quadrilateral/triangular grids via Method 3.

Mesh	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
hybrid-6	1.35E-01	-	9.34E-02	-	1.50E-02	-
hybrid-12	4.07E-02	1.74	3.10E-02	1.59	2.82E-03	2.41
hybrid-24	1.16E-02	1.81	9.42E-03	1.72	4.39E-04	2.68
hybrid-48	3.33E-03	1.80	2.79E-03	1.75	6.35E-05	2.79
hybrid-96	9.08E-04	1.88	7.84E-04	1.83	8.85E-06	2.84

Based on the test results, it can be concluded that only Method 3 performs consistently on all types of meshes. Specifically, it is second-order accurate in reconstructing solution gradients at both the cell centers and face centers, and third-order accurate in reconstructing solutions at the face centers.

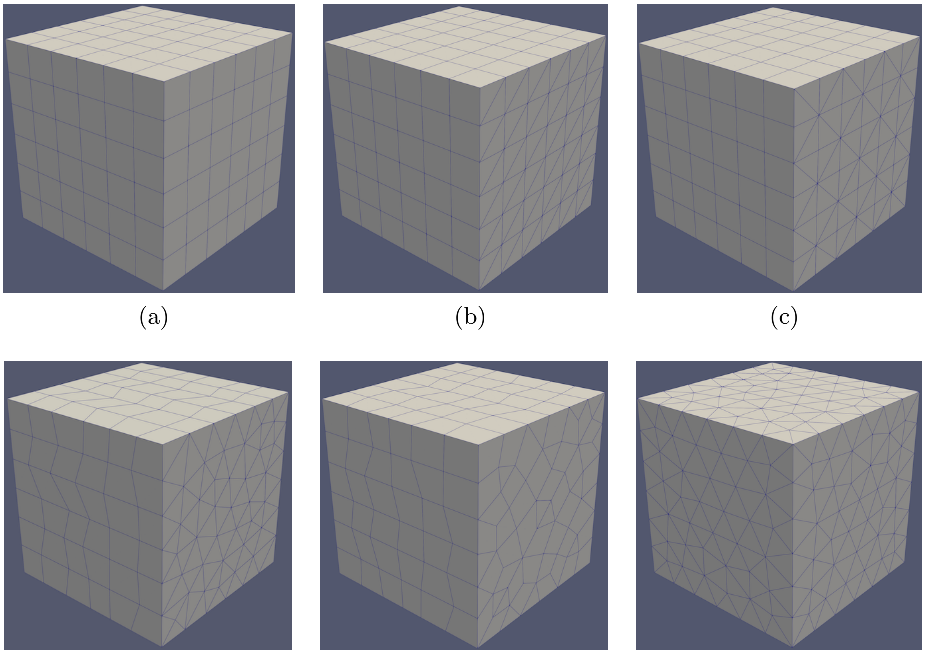
IV.B. 3-D Tests

We also provide 3-D tests. The following analytical solution is used for this purpose

$$u(x,y,z) = \begin{cases} 1 + \frac{1}{8}\left[1+\cos\left(\frac{4\pi x}{3}\right)\right]\left[1+\cos\left(\frac{4\pi y}{3}\right)\right]\left[1+\cos\left(\frac{4\pi z}{3}\right)\right] & |x| \leq 0.75,\ |y| \leq 0.75 \text{ and } |z| \leq 0.75 \\ 0 & \text{elsewhere} \end{cases} \tag{32}$$

where the computational domain is $[-1,1] \times [-1,1] \times [-1,1]$.

Since it was concluded from the previous 2-D tests that only Method 3 behaves consistently across all types of meshes, Method 3 will be the sole method used in the current 3-D tests. Figure 7 shows the various 3-D meshes used in this test. Each type of mesh is isotropically subdivided repeatedly to obtain a series of refined meshes. Note that when the tetrahedral mesh (cf. Fig. 7f) is subdivided, each tetrahedral element is subdivided into four child tetrahedra and one octahedron to preserve the mesh quality of the subsequent meshes.



(a) (b) (c)

(d) (e) (f)

Figure 7: 3-D meshes used in the tests. (a) hexahedral mesh. (b) structured prismatic mesh of Type I. (c) structured prismatic mesh of Type II. (d) unstructured prismatic mesh. (e) hybrid unstructured prismatic mesh. and (f) unstructured tetrahedral mesh.

Tables 20-25 shows the reconstruction order of accuracy of Method 3 on each of the meshes shown in Fig. 7. As can be seen, Method 3 exhibits consistent orders of accuracy in all kinds of meshes, i.e. second order accurate in reconstructing solution gradients at cell centers and face centers and third order accurate in reconstructing solution at face centers.

The results from the above 3-D tests further reinforce the conclusion drawn from the 2-D tests: the reconstruction process based on Method 3 consistently achieves second-order accuracy in reconstructing the gradients at both the cell centers and face centers, and third-order accuracy in reconstructing the solutions at the face centers.

Table 20: Accuracy of reconstruction of Method 3 on structured hexahedral grids.

ncell	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
hex-6	8.87E-02	-	6.28E-02	-	1.38E-02	-
hex-12	3.27E-02	1.44	2.38E-02	1.40	2.38E-03	2.54
hex-24	7.38E-03	2.15	6.03E-03	1.98	2.98E-04	3.00
hex-48	1.92E-03	1.95	1.59E-03	1.93	3.80E-05	2.97
hex-96	4.84E-04	1.99	4.04E-04	1.97	4.74E-06	3.00

Table 21: Accuracy of reconstruction of Method 3 on structured prismatic grids of Type I.

ncell	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
pri-6-type1	8.36E-02	-	6.83E-02	-	1.10E-02	-
pri-12-type1	2.85E-02	1.55	2.45E-02	1.48	1.86E-03	2.57
pri-24type1	7.15E-03	1.99	6.28E-03	1.96	2.34E-04	2.98
pri-48type1	1.87E-03	1.93	1.67E-03	1.92	3.03E-05	2.95
pri-96type1	4.77E-04	1.97	4.27E-04	1.96	3.83E-06	2.98

Table 22: Accuracy of reconstruction of Method 3 on structured prismatic grids of Type II.

ncell	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
pri-6-type2	8.52E-02	-	6.83E-02	-	1.13E-02	-
pri-12-type2	2.82E-02	1.60	2.42E-02	1.50	1.86E-03	2.60
pri-24-type2	6.98E-03	2.02	6.13E-03	1.98	2.31E-04	3.01
pri-48-type2	1.83E-03	1.93	1.62E-03	1.92	2.97E-05	2.96
pri-96-type2	4.63E-04	1.98	4.15E-04	1.97	3.74E-06	2.99

Table 23: Accuracy of reconstruction of Method 3 on prismatic grids of Type III.

ncell	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
pri-6-type3	6.41E-02	-	5.46E-02	-	9.07E-03	-
pri-12-type3	2.19E-02	1.55	1.84E-02	1.57	1.49E-03	2.61
pri-24-type3	5.96E-03	1.88	5.22E-03	1.82	2.05E-04	2.86
pri-48-type3	1.61E-03	1.89	1.44E-03	1.86	2.80E-05	2.87
pri-96-type3	4.25E-04	1.92	3.86E-04	1.90	3.73E-06	2.91

Table 24: Accuracy of reconstruction of Method 3 on prismatic grids of Type IV.

ncell	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
pri-6-type4	6.38E-02	-	5.09E-02	-	9.19E-03	-
pri-12-type4	2.12E-02	1.59	1.74E-02	1.55	1.68E-03	2.45
pri-24-type4	6.03E-03	1.82	5.21E-03	1.74	2.51E-04	2.74
pri-48-type4	1.70E-03	1.82	1.51E-03	1.79	3.48E-05	2.85
pri-96-type4	4.61E-04	1.89	4.17E-04	1.85	4.74E-06	2.88

Table 25: Accuracy of reconstruction of Method 3 on unstructured tetrahedral meshes.

ncell	$(u_x)_c$		$(u_x)_f$		u_f^L	
	error	order	error	order	error	order
tet-6	4.72E-02	-	4.16E-02	-	5.26E-03	-
tet-12	1.62E-02	1.54	1.44E-02	1.53	1.03E-03	2.35
tet-24	4.70E-03	1.79	4.30E-03	1.74	1.90E-04	2.44
tet-48	1.38E-03	1.76	1.30E-03	1.73	3.02E-05	2.66
tet-96	4.03E-04	1.78	3.85E-04	1.75	4.47E-06	2.76

V. Conclusions and Future Work

In this paper, we numerically verified a third-order accurate reconstruction method for arbitrary 2-D and 3-D unstructured grids. The multi-dimensional reconstruction follows a similar approach to that used for 1-D grids. To summarize, the reconstruction is implemented as follows:

- Start with the known solutions at cell centers.
- Obtain the solution and its gradient at vertices via the least square approach using the known solutions at surrounding cell centers (19).
- Obtain the solution gradient at the cell center via simple arithmetic averaging of the solution gradient at vertices (21).
- Obtain the solution gradient at the face center via simple arithmetic averaging of the solution gradient at vertices (22).
- Obtain the solution at the face center via the quadratic fits (29) and (30) using the solution gradients at both the cell center and the face center.

The current third-order reconstruction method utilizes a compact stencil and does not require explicit knowledge of the second derivatives of the solution. The primary computational cost lies in determining the nodal gradients using the least squares method, which can be easily implemented in a parallel finite volume (FV) solver based on mesh partitioning.

Although the third-order accuracy of the reconstruction has been numerically verified — a necessary condition for the overall accuracy of a finite volume solver — the total accuracy in an actual FV solver requires further validation. In an FV solver, the overall accuracy depends not only on the interface flux computation but also on the spatial and temporal integration methods. High-order FV solvers typically use quadrature rules for integrating the inter-cell fluxes. For a third-order FV solver, third-order accuracy can be achieved with a single quadrature point at the face center, along with a curvature correction, as demonstrated in.[5] The author intends to implement this third-order accurate reconstruction in their FV solvers.

Several issues warrant further investigation, including: (1) the impact of equation nonlinearity on the accuracy of the reconstruction; (2) the need for special attention to reconstruction near domain boundaries, as inaccuracies there can degrade the overall accuracy; and (3) how the choice of cell averages or point values affects the reconstruction.

Acknowledgments

This work is partially supported by U.S. Department of Energy Office of Science Award No. DE-SC0023125, NSF Award No. 2219542 and NASA Award No. 80NSSC21M0332.

References

[1] Clarence Burg. Higher order variable extrapolation for unstructured finite volume RANS flow solvers. In *17th AIAA Computational Fluid Dynamics Conference, AIAA 2005-4999*, June 2005.

[2] B.P. Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer Methods in Applied Mechanics and Engineering*, 19(1):59–98, 1979.

[3] Xu-Dong Liu, Stanley Osher, and Tony Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200–212, 1994.

[4] Yuan Liu and Yong-Tao Zhang. A robust reconstruction for unstructured WENO schemes. *Journal of Scientific Computing*, 54(2):603–621, Feb 2013.

[5] Hiroaki Nishikawa and Jeffery A. White. An efficient quadratic interpolation scheme for a third-order cell-centered finite-volume method on tetrahedral grids. *Journal of Computational Physics*, 490:112324, 2023.

[6] Carl Ollivier-Gooch and Michael Van Altena. A high-order-accurate unstructured mesh finite-volume scheme for the advection-diffusion equation. *Journal of Computational Physics*, 181(2):729–752, 2002.

[7] Emmett Padway and Hiroaki Nishikawa. Resolving confusion over third order accuracy of U-MUSCL. *AIAA Journal*, 60(3), 2022.

[8] S. Tu, S. Aliabadi, A. Johnson, and M Watts. A robust parallel implicit finite volume solver for high-speed compressible flows, January 2005. 43rd AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper-2005-1396.

[9] S. Tu, S. Aliabadi, R. Patel, and M. Watts. An implementation of the Spalart-Allmaras DES model in an implicit unstructured hybrid finite volume/element solver for incompressible turbulent flow. *Int. J. Numer. Meth. Fluids*, 59(9):1051–1062, 2008.