

Resolving Confusion Over Third Order Accuracy of U-MUSCL

Emmett Padway* and Hiroaki Nishikawa†

National Institute of Aerospace, Hampton, VA 23666, USA

In this paper, we discuss the U-MUSCL reconstruction scheme – an unstructured-grid extension of Van Leer's κ -scheme – proposed by Burg for the edge-based discretization [AIAA Paper 2005-4999]. This technique has been widely used in practical unstructured-grid fluid-dynamics solvers but with confusions: e.g., third-order accuracy with $\kappa = 1/2$ or $\kappa = 1/3$. This paper clarifies some of these confusions: e.g., the U-MUSCL scheme can be third-order accurate in the point-valued solution with $\kappa = 1/3$ on regular grids for linear equations, it can be third-order with $\kappa = 1/2$ as the QUICK scheme in one dimension. In conclusion, the U-MUSCL scheme cannot be third-order accurate for nonlinear equations (except a very special case) but can be a very accurate second-order scheme. Two techniques are discussed, which render the U-MUSCL scheme genuinely third-order on a regular grid: an efficient flux-reconstruction with $\kappa = 1/3$ and a special source term quadrature formula for $\kappa = 1/2$.

1. Introduction

Despite decades of progress in high-order methods motivated by a desire for increase in accuracy, practical unstructured-grid computational fluid dynamics (CFD) solvers are still dominated by lower-order methods that can be written in the flux-balance form given below. A residual at a node/cell j is given by

$$\text{Res}_j = \sum_{k \in \{k_j\}} \Phi_{jk} A_{jk}, \quad (1.1)$$

where $\{k_j\}$ is a set of neighbors, A_{jk} is a face area, and Φ_{jk} is a numerical flux. Typically, such a flux-balance method is second-order accurate at best on arbitrary grids. Higher-order accuracy will require significantly more complex algorithms (e.g., high-order flux quadrature) and these are rarely employed in practical codes. Under certain constraints, much simpler high-order methods are available. One such example is the third-order edge-based method, which achieves third-order accuracy on arbitrary grids but only with simplex elements [1–3]. Another example is the unstructured MUSCL (U-MUSCL) scheme [4], which is claimed to be third-order on regular grids. These schemes are efficient and simple since they are based on a single flux evaluation per edge, and can be easily incorporated in existing unstructured-grid codes by modifying the solution/flux reconstruction part and the source discretization [3]. The U-MUSCL scheme is particularly simple to implement and is widely employed in many practical codes [4–9]. However, there still exist confusions about the U-MUSCL scheme: it is third-order accurate with $\kappa = 1/2$ [4] or $\kappa = 1/3$ [10], $\kappa = -1/6$ [11]. Also, as discussed in Ref.[12], in fact, the U-MUSCL scheme cannot be high-order for nonlinear problems in multi-dimensions. Therefore, in many CFD simulations, third-order accuracy is not actually achieved even on a regular grid. This paper clarifies the confusions and discusses correct ways to achieve high-order accuracy on regular grids with the U-MUSCL reconstruction scheme.

The U-MUSCL reconstruction scheme, widely employed in practical unstructured-grid CFD codes today, was first proposed by Burg for the edge-based discretization in Ref.[4] as an unstructured-grid extension of Van Leer's κ -scheme, [13, 14]. For a one-dimensional uniform grid, the kappa scheme leads to a third-order accurate finite-volume advection scheme with $\kappa = 1/3$. Because of this special property, $\kappa = 1/3$ has been a popular choice in MUSCL finite-volume schemes. On the other hand, when Burg proposed its extension to unstructured grids in Ref.[4], he showed that third-order accuracy was obtained with $\kappa = 1/2$ in the node-centered edge-based method for a one-dimensional nonlinear system of conservation laws with integrated source terms. He then demonstrated superior performance with $\kappa = 1/2$ for practical CFD simulations. Since then, there has been a confusion about the value of κ for third-order accuracy; one of the main contributions of this paper is to clarify the choice of κ for third-order accuracy.

* Research Scholar (emmett.padway@nianet.org), 100 Exploration Way, Hampton, VA 23666 USA, Member AIAA

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

Recently, in a series of papers [12,15,16], we provided clarifications of the U-MUSCL type schemes. Ref.[15] provides a detailed truncation error analysis of a finite-volume scheme with cell-averaged solutions (i.e., the MUSCL scheme) for a nonlinear conservation law and proves that third-order accuracy is achieved with $\kappa = 1/3$. Ref.[16] demonstrates that a finite-volume scheme with point-valued solutions (e.g., the QUICK scheme) is third-order accurate with $\kappa = 1/2$. Ref.[12] shows that the U-MUSCL scheme achieves third-order accuracy with $\kappa = 1/3$ on a regular grid in multi-dimensions as a conservative finite-difference scheme but only for linear equations; a particular class of exact solutions effectively linearizes nonlinear equations and allows third-order accuracy to be observed. In this paper, we will follow these previous papers and discuss how the U-MUSCL scheme can be third-order in one dimension and how it cannot be third-order in higher dimensions and for general nonlinear equations.

Although not genuinely third-order accurate, the U-MUSCL scheme does improve solutions with $\kappa = 1/2$ or $\kappa = 1/3$ as demonstrated in many test cases (see, e.g., Refs [4, 6]). However, genuine third-order accuracy would be desirable for high-fidelity simulations using highly refined grids because second-order schemes are significantly less efficient on such grids. To this end, we discuss two techniques to construct U-MUSCL-type schemes that can be genuinely third-order accurate on regular grids in multi-dimensions. One is the flux reconstruction as demonstrated with an efficient chain-ruled-based formula in Ref.[12]. The other is a special source term quadrature that eliminates the leading second-order error for $\kappa = 1/2$. To the best of the authors' knowledge, this particular $\kappa = 1/2$ scheme is a newly discovered scheme. These schemes can be extended to fourth- and/or fifth-order, but in this paper, we will focus on third-order accuracy, which is sufficient to resolve confusions. Higher-order schemes will be discussed in a subsequent paper.

Additional clarifications are provided: (1) the solution jump across a face between the left and right solutions reconstructed by U-MUSCL vanishes for quadratic functions on arbitrary grids for any value of κ , (2) the U-MUSCL solution reconstruction is exact for quadratic function with $\kappa = 1/2$ on arbitrary grids provided the gradients are computed by a quadratic least-squares (LSQ) method, and (3) the U-MUSCL reconstruction scheme is directly applicable to cell-centered methods but second-order accuracy may be lost if the face centroid is not exactly located at halfway between two adjacent centroids.

The paper is organized as follows. Section 2 describes the edge-based discretization with the U-MUSCL scheme for the Euler equations. Section 3 discusses the clarification of the U-MUSCL scheme. Section 4 describes a genuinely third-order scheme based on efficient flux reconstruction. Section 5 presents the special third-order U-MUSCL scheme with $\kappa = 1/2$. Section 6 discusses the quadratic exactness of the U-MUSCL reconstruction formula and vanishing solution jumps with quadratically exact gradients. Section 7 discusses the use of U-MUSCL in cell-centered methods. Section 8 presents numerical results. Finally, the paper concludes with remarks in Section 9.

2. Edge-Based Discretization with U-MUSCL Reconstruction Scheme

2.1. Edge-based discretization with U-MUSCL

Consider the Euler equations:

$$\partial_t \mathbf{u} + \operatorname{div} \mathcal{F} = \mathbf{s}, \quad (2.1)$$

where \mathbf{s} is a source (or forcing) term vector, and

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{bmatrix}, \quad \mathcal{F} = \begin{bmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + p \mathbf{I} \\ \rho \mathbf{v} H \end{bmatrix}, \quad (2.2)$$

where ρ is the density, p is the pressure, \mathbf{v} is the velocity vector, \otimes denotes the dyadic product, \mathbf{I} is the identity matrix, and $E = H - p/\rho = p/(\gamma - 1) + \mathbf{v}^2/2$ is the specific total energy ($\gamma = 1.4$).

Following Burg [4], we store the solution values at nodes and discretize the system by the edge-based discretization method (see Figure 1):

$$\frac{d\mathbf{u}_j}{dt} + \frac{1}{V_j} \mathbf{Res}_j = \mathbf{s}_j, \quad (2.3)$$

where V_j denotes the dual volume around the node j and the residual \mathbf{Res}_j is defined by the edge-based quadra-

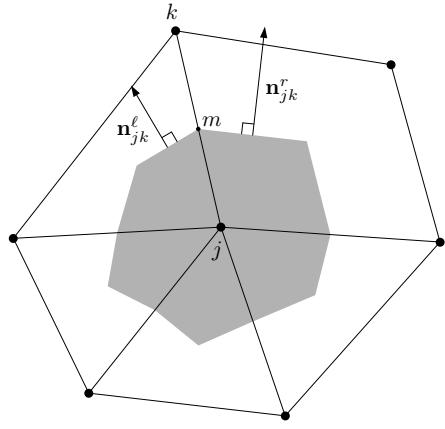


Figure 1: Stencil for the edge-based discretization. The directed-area vector across the nodes j and k , denoted by \mathbf{n}_{jk} , is defined by $\mathbf{n}_{jk} = \mathbf{n}_{jk}^l + \mathbf{n}_{jk}^r$.

ture formula, first introduced in equation 1.1,

$$\mathbf{Res}_j = \sum_{k \in \{k_j\}} \Phi_{jk} A_{jk}, \quad (2.4)$$

where $\{k_j\}$ is a set of edge-connected neighbor nodes, $\Phi_{jk} = \Phi_{jk}(\mathbf{w}_L, \mathbf{w}_R, \mathbf{n}_{jk})$ is a numerical flux evaluated at the edge midpoint in the direction of the directed-area vector \mathbf{n}_{jk} (see Figure 1), \mathbf{w} denotes a vector of primitive variables $\mathbf{w} = (\rho, u, v, p)$, \mathbf{w}_L and \mathbf{w}_R are primitive variables reconstructed from the nodes j and k , respectively, and A_{jk} is the magnitude of the directed-area vector associated with the edge defined by the nodes j and k . The numerical flux is defined by the Roe flux [17]:

$$\Phi_{jk} = \frac{1}{2} [\mathbf{f}(\mathbf{w}_L, \mathbf{n}_{jk}) + \mathbf{f}(\mathbf{w}_R, \mathbf{n}_{jk})] - \frac{1}{2} \hat{\mathbf{D}}_n [\mathbf{u}(\mathbf{w}_R) - \mathbf{u}(\mathbf{w}_L)], \quad (2.5)$$

where $\mathbf{f} = \mathcal{F} \cdot \hat{\mathbf{n}}_{jk}$ is the flux projected along the directed-area vector $\hat{\mathbf{n}}_{jk} = \mathbf{n}_{jk}/|\mathbf{n}_{jk}|$, and $\hat{\mathbf{D}}_n = |\partial \mathbf{f} / \partial \mathbf{u}|$ is the dissipation term evaluated with the Roe averages [17]. We will also employ the Rusanov flux [18] in accuracy verification tests in two dimensions, where $\hat{\mathbf{D}}_n$ is replaced by a scalar value defined by the maximum eigenvalue evaluated with the Roe averages. The left and right states are computed by the U-MUSCL reconstruction scheme, as proposed by Burg [4],

$$\mathbf{w}_L = \mathbf{w}_j + \frac{\kappa}{2} (\mathbf{w}_k - \mathbf{w}_j) + \frac{1-\kappa}{2} \bar{\nabla} \mathbf{w}_j \cdot (\mathbf{x}_k - \mathbf{x}_j), \quad (2.6)$$

$$\mathbf{w}_R = \mathbf{w}_k - \frac{\kappa}{2} (\mathbf{w}_k - \mathbf{w}_j) - \frac{1-\kappa}{2} \bar{\nabla} \mathbf{w}_k \cdot (\mathbf{x}_k - \mathbf{x}_j), \quad (2.7)$$

where $\bar{\nabla} \mathbf{w}_j$ and $\bar{\nabla} \mathbf{w}_k$ are the solution gradients numerically computed by a linear LSQ method with edge-connected neighbors, and κ is a user-defined parameter.

2.2. Equivalent forms

An interesting equivalent form of the U-MUSCL scheme is

$$\mathbf{w}_L = \kappa \frac{\mathbf{w}_j + \mathbf{w}_k}{2} + (1-\kappa) \left[\mathbf{w}_j + \frac{1}{2} \bar{\nabla} \mathbf{w}_j \cdot (\mathbf{x}_k - \mathbf{x}_j) \right], \quad (2.8)$$

$$\mathbf{w}_R = \kappa \frac{\mathbf{w}_k + \mathbf{w}_j}{2} + (1-\kappa) \left[\mathbf{w}_k - \frac{1}{2} \bar{\nabla} \mathbf{w}_k \cdot (\mathbf{x}_k - \mathbf{x}_j) \right], \quad (2.9)$$

which shows that the U-MUSCL scheme is a weighted average of the solution average over the edge and the linear extrapolation with gradients [19].

The U-MUSCL scheme also has the following alternative form:

$$\mathbf{w}_L = \mathbf{w}_j + \frac{1}{4} [(1-\kappa)\Delta^- + (1+\kappa)\Delta^+], \quad (2.10)$$

$$\mathbf{w}_R = \mathbf{w}_k - \frac{1}{4} [(1-\kappa)\Delta^+ + (1+\kappa)\Delta^-], \quad (2.11)$$

where

$$\Delta^- = 2\bar{\nabla}\mathbf{w}_j \cdot (\mathbf{x}_k - \mathbf{x}_j) - (\mathbf{w}_k - \mathbf{w}_j), \quad \Delta^+ = \mathbf{w}_k - \mathbf{w}_j, \quad (2.12)$$

which, by definition, reduces to Van Leer's κ -scheme [13, 14] on a regular quadrilateral/hexahedral grid with the gradient $\bar{\nabla}\mathbf{w}_j$ reduced to the central difference gradient. It is noted that the β -scheme in Ref.[20] with a parameter β is equivalent to the κ -scheme with $\kappa = 1 - 2\beta$ but extended to unstructured grids in a manner specific to a node-centered finite-volume discretization.

2.3. U-MUSCL of Yang and Harris

Yang and Harris [11, 21] extended the U-MUSCL reconstruction scheme by adding a curvature term $\mathbf{C}_j/2$,

$$\mathbf{w}_L = \mathbf{w}_j + \frac{\kappa}{2}(\mathbf{w}_k - \mathbf{w}_j) + \frac{1-\kappa}{2}\bar{\nabla}\mathbf{w}_j \cdot (\mathbf{x}_k - \mathbf{x}_j) + \frac{1}{2}\mathbf{C}_j, \quad (2.13)$$

where \mathbf{C}_j is defined, with an additional parameter κ_3 , as

$$\mathbf{C}_j = \frac{\kappa_3}{4}(\bar{\nabla}\mathbf{w}_k - \bar{\nabla}\mathbf{w}_j) \cdot (\mathbf{x}_k - \mathbf{x}_j) + \frac{1-\kappa_3}{4}(\mathbf{x}_k - \mathbf{x}_j)^t \bar{\nabla}^2 \mathbf{w}_j \cdot (\mathbf{x}_k - \mathbf{x}_j), \quad (2.14)$$

They set $\kappa = -1/6$ and $\kappa_3 = -4/3$. The second derivatives, $\bar{\nabla}^2 \mathbf{w}_j$, are computed by applying a linear LSQ method twice. In Refs.[11, 21], they claim that the scheme is third-order accurate with $\kappa = -1/6$ and $\kappa_3 = 0$:

$$\mathbf{w}_L = \mathbf{w}_j + \frac{\kappa}{2}(\mathbf{w}_k - \mathbf{w}_j) + \frac{1-\kappa}{2}\bar{\nabla}\mathbf{w}_j \cdot (\mathbf{x}_k - \mathbf{x}_j) + \frac{1}{2}(\mathbf{x}_k - \mathbf{x}_j)^t \bar{\nabla}^2 \mathbf{w}_j \cdot (\mathbf{x}_k - \mathbf{x}_j). \quad (2.15)$$

As we will show later, this scheme is essentially equivalent to the original U-MUSCL scheme (2.6) with $\kappa = 1/3$ in one dimension. For third-order accuracy, however, this is not an efficient scheme because it requires the computation and the storage of the second derivatives and has a wider stencil. This scheme will be further discussed in a subsequent paper for fourth-order accuracy.

3. Clarification of U-MUSCL : third-order with $\kappa = 1/2$, $\kappa = 1/3$, or $\kappa = -1/6$

In this section, we will clarify the U-MUSCL scheme and discuss how and with what value of κ it can be third-order accurate in one dimension. Note that third-order accuracy in one dimension does not necessarily imply third-order accuracy in higher dimensions; but the one-dimensional U-MUSCL scheme is the origin of some of the confusions and thus worth discussing. In doing so, we will have to make a clear distinction between cell-averaged and point-valued solutions and source terms, whose difference is of second-order and thus matters for third- and higher-order accuracy. In short, we will show that the U-MUSCL scheme can be a third-order finite-volume scheme with point-valued solutions ($\kappa = 1/2$) or with cell-averaged solutions ($\kappa = 1/3$) for general nonlinear equations, or it can be a third-order finite-difference scheme ($\kappa = 1/3$) for linear equations. Furthermore, $\kappa = -1/6$ of U-MUSCL-YH is equivalent to the third-order finite-difference scheme with $\kappa = 1/3$; therefore, it is third-order only for linear equations.

3.1. U-MUSCL scheme in one dimension

For simplicity, we consider a scalar nonlinear conservation law:

$$\partial_t u + \partial_x f = s, \quad (3.1)$$

where f is a function of u , $f = f(u)$. The U-MUSCL scheme in a one-dimensional grid of uniform spacing h , i.e., $x_{i+1} - x_i = h$, is given by

$$\frac{du_i}{dt} + \frac{\phi_{i+1/2} - \phi_{i-1/2}}{h} = s_i, \quad (3.2)$$

where u_i denotes a solution stored at $x = x_i$, and the numerical flux ϕ is evaluated by the upwind flux:

$$\phi_{i+1/2} = \frac{1}{2}[f(u_L) + f(u_R)] - \frac{1}{2}\left|\frac{\partial f}{\partial u}\right|_{j+1/2}(u_R - u_L), \quad (3.3)$$

where the left and right states are evaluated by the U-MUSCL reconstruction scheme (equivalent to Van Leer's scheme in one dimension):

$$u_L = u_i + \frac{1}{4}[(1-\kappa)\Delta u^- + (1+\kappa)\Delta u^+], \quad u_R = u_{i+1} - \frac{1}{4}[(1-\kappa)\Delta u^+ + (1+\kappa)\Delta u^-], \quad (3.4)$$

where

$$\Delta u^- = u_i - u_{i-1}, \quad \Delta u^+ = u_{i+1} - u_i. \quad (3.5)$$

It is important to note that the U-MUSCL scheme is typically implemented as above without clear definitions of the numerical solution and the source term: point value or cell average. The distinction is not important for second-order accuracy but important for third-order accuracy because they differ by $O(h^2)$. In what follows, we will discuss three cases where the U-MUSCL scheme achieves third-order accuracy, depending on the definitions of numerical solution u_i and the source term evaluation s_i .

3.2. Finite-volume with cell-averaged solution (MUSCL): third-order with $\kappa = 1/3$

Suppose the numerical solution is a cell-averaged solution and the source term is cell-averaged, so that the U-MUSCL scheme (3.2) actually implements

$$\frac{d\bar{u}_i}{dt} + \frac{\phi_{i+1/2} - \phi_{i-1/2}}{h} = \bar{s}_i, \quad (3.6)$$

where

$$\bar{u}_i = \frac{1}{h} \int_{x_j-h/2}^{x_j+h/2} u dx, \quad \bar{s}_i = \frac{1}{h} \int_{x_j-h/2}^{x_j+h/2} s dx. \quad (3.7)$$

Then, this is a finite-volume scheme with cell-averaged solutions, known as MUSCL of Van Leer, approximating the exact integral form of the conservation law:

$$\frac{d\bar{u}_i}{dt} + \bar{f}_x = \bar{s}_i, \quad (3.8)$$

where

$$\bar{f}_x = \frac{1}{h} \int_{x_i-h/2}^{x_i+h/2} f_x dx = \frac{1}{h} [f(u_{i+1/2}) - f(u_{i-1/2})]. \quad (3.9)$$

It is important to note that arbitrary high-order accuracy is possible in the MUSCL scheme because the integral form is exact. The accuracy of the discretization is then determined solely by the solution reconstruction, at least in one dimension: second-order accuracy with a linear reconstruction and third-order with a quadratic reconstruction. Note also that the reconstruction is performed to obtain a point-valued solution at a face from cell-averaged numerical solutions stored at cells. The choice $\kappa = 1/3$ does it exactly for quadratic functions (see Ref.[15], for example).

As derived in Ref.[15], the truncation error of the scheme (3.6) is given by

$$\frac{d\bar{u}_i}{dt} + \bar{f}_x + \frac{3\kappa-1}{12} [f_{uu}u_xu_{xx} + f_uu_{xxx}] h^2 + O(h^3) = \bar{s}_i. \quad (3.10)$$

Comparing this with the target integral form (3.8), we find that the U-MUSCL scheme in one dimension achieves third-order accuracy in the cell-averaged solutions with $\kappa = 1/3$ if the numerical solution is taken as a cell-averaged solution and the source term is cell averaged.

3.3. Finite-volume with point-valued solution (QUICK): third-order with $\kappa = 1/2$

In the finite-volume scheme, the numerical solution does not have to be the cell average and it may be taken as the point value at the cell center. A simple way to construct a third-order finite-volume scheme with point values is to convert the cell-averaged solution \bar{u}_i to the point value u_i in the MUSCL finite-volume scheme (3.6). To do so, we express the cell-average \bar{u}_i in terms of the point values, u_{i+1} , u_i , and u_{i-1} :

$$\bar{u}_i = u_i + \frac{h^2}{24} \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + O(h^4) = u_i + \frac{1}{24} (u_{i+1} - 2u_i + u_{i-1}) + O(h^4), \quad (3.11)$$

and substitute it into Equation (3.6) to get

$$\frac{du_i}{dt} + \frac{1}{24} \left(\frac{du_{i+1}}{dt} - 2 \frac{du_i}{dt} + \frac{du_{i-1}}{dt} \right) + \frac{\phi_{i+1/2} - \phi_{i-1/2}}{h} = \bar{s}_i + O(h^4). \quad (3.12)$$

This is a finite-volume scheme that updates the point-valued solution; the QUICK scheme of Leonard [22] is an example. Other strategies are available for expressing the time derivative terms with the point-valued solutions as in Ref.[16]. An example is shown in Ref.[23] for a general construction of finite-volume discretizations with point-valued solutions.

Because of the switch to the point-valued solution, we obtain a slightly different truncation error, as derived in Ref.[16],

$$\frac{du_i}{dt} + \frac{1}{24} \left(\frac{du_{i+1}}{dt} - 2 \frac{du_i}{dt} + \frac{du_{i-1}}{dt} \right) + \bar{f}_x + \frac{2\kappa - 1}{8} [f_{uu}u_xu_{xx} + f_uu_{xxx}] h^2 = \bar{s}_i + O(h^3), \quad (3.13)$$

or as an approximation to the integral form (3.8),

$$\frac{d\bar{u}_i}{dt} + \bar{f}_x + \frac{2\kappa - 1}{8} [f_{uu}u_xu_{xx} + f_uu_{xxx}] h^2 = \bar{s}_i + O(h^3), \quad (3.14)$$

which shows that the scheme is third-order accurate at $\kappa = 1/2$.

Comparing this with the MUSCL scheme in the previous section, we conclude that the U-MUSCL scheme (3.2) achieves third-order accuracy in one dimension with $\kappa = 1/3$ if the numerical solution is the cell average and the source term is cell averaged, or with $\kappa = 1/2$ if the numerical solution is the point value, the source term is evaluated at the cell center, and there is no time derivative (i.e., for steady problems). In Ref. [4], Burg performed an accuracy verification test for a one-dimensional steady problem with a cell-averaged source term and apparently with point-valued solutions, which is the QUICK scheme. This is the reason that he obtained third-order accuracy with $\kappa = 1/2$, not $\kappa = 1/3$. If he had solved an unsteady problem or one with a source term, he would not have been able to obtain third-order accuracy without a consistent time-derivative treatment such as shown in Equation (3.12).

3.4. Finite-difference scheme: third-order with $\kappa = 1/3$ for linear equations

If both the solution and the source term are point values as typical in the U-MUSCL scheme,

$$\frac{du_i}{dt} + \frac{\phi_{i+1/2} - \phi_{i-1/2}}{h} = s_i, \quad (3.15)$$

where $s_i = s(x_i)$, then, this scheme is second-order accurate at best for nonlinear equations. As derived in Ref.[16], the truncation error is given below.

$$\frac{du_i}{dt} + f_x = s_i - \frac{1}{24} [f_{uuu}(u_x)^3 + 6\kappa f_{uu}u_xu_{xx} + 2(3\kappa - 1)f_uu_{xxx}] h^2 + O(h^3). \quad (3.16)$$

This shows that the third-order accuracy is achieved with $\kappa = 1/3$ for linear equations, as $f_{uuu} = f_{uu} = 0$; this fact has also been pointed out in Refs.[24, 25]. Although there are high-order accuracy verification studies seen in the literature [11, 21, 26], as shown in Ref.[12], these high-order results are due to unexpected linearization of nonlinear equations by a particular class of exact solutions (i.e., a function with a small perturbation). Later, we will show an example of this phenomenon. See Ref.[12] for further details.

3.5. U-MUSCL of Yang and Harris: $\kappa = -1/6$

In one dimension, the U-MUSCL scheme of Yang and Harris (2.13) with $\kappa_3 = 0$ reduces to:

$$u_L = u_i + \frac{1}{4} [(1-\kappa)\Delta u^- + (1+\kappa)\Delta u^+] + \frac{1}{32} (u_{i+2} - 2u_j + u_{i-2}). \quad (3.17)$$

where, the extra term affects only the curvature term in the original U-MUSCL scheme. Let us write it in the quadratic reconstruction form:

$$u_L = u_i + \frac{1}{2} \frac{u_{i+1} - u_{i-1}}{2h} \left(\frac{h}{2} \right) + \left[\frac{3\kappa}{2} \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + \frac{3}{4} \frac{u_{i+2} - 2u_i + u_{i-2}}{(2h)^2} \right] \left[\left(\frac{h}{2} \right)^2 - \frac{h^2}{12} \right]. \quad (3.18)$$

Without the extra term, the scheme becomes third-order accurate for linear equations with $\kappa = 1/3$ as discussed in the previous section; therefore, the coefficient to the curvature term should be $3\kappa/2 = 1/2$. With the extra term, this requires a different value of κ . For third-order accuracy, it suffices to consider a quadratic solution. Let u_{xx} be the constant second derivative of the quadratic solution. Then, we can write

$$u_L = u_i + \frac{1}{2} \frac{u_{i+1} - u_{i-1}}{2h} \left(\frac{h}{2} \right) + \left[\frac{3\kappa}{2} + \frac{3}{4} \right] u_{xx} \left[\left(\frac{h}{2} \right)^2 - \frac{h^2}{12} \right], \quad (3.19)$$

and solving

$$\frac{3\kappa}{2} + \frac{3}{4} = \frac{1}{2}, \quad (3.20)$$

for κ , we obtain

$$\kappa = -\frac{1}{6}. \quad (3.21)$$

Therefore, the U-MUSCL scheme of Yang and Harris is essentially equivalent (although with a much wider stencil) to the original U-MUSCL scheme with $\kappa = 1/3$. It can be a third-order finite-volume scheme with cell-averaged solutions and a cell-averaged source term, or a third-order finite-difference scheme (for linear equations only). For this reason, we will not discuss their scheme any further and focus on the original U-MUSCL scheme in the rest of the paper.

3.6. Remarks

As we have seen, the U-MUSCL scheme can be third-order accurate in one dimension in three different ways: (1) FVC, a finite-volume scheme with cell-averaged solutions and $\kappa = 1/3$, (2) FVP, a finite-volume scheme with point-valued solutions and $\kappa = 1/2$, (3) FD, a conservative finite-difference scheme with point-valued solutions, only for linear equations.

Among these cases, the third-order finite-volume schemes, FVC and FVP, do not extend to higher dimensions because the U-MUSCL scheme (2.3) has only one flux per face; higher-order flux quadrature is required to achieve third-order accuracy as a finite-volume scheme. There exist techniques to achieve high-order without high-order quadrature [27, 28], but they are currently applicable to Cartesian grids only (our interest is to develop an unstructured-grid scheme that reduces to a high-order scheme when a grid happens to be regular.) Therefore, the finite-difference scheme is the only way that the U-MUSCL scheme can achieve third-order accuracy on a regular grid in higher dimensions. But this is true only for linear equations.

Therefore, all the existing U-MUSCL schemes [4, 11, 21, 26] are second-order accurate at best for nonlinear equations such as the Euler equations, even on a regular grid. Nevertheless, these schemes have been demonstrated to provide improved resolution (see also Ref.[24]); therefore they are useful second-order schemes for practical simulations. Since genuine third- or higher-order accuracy is sought for high-fidelity simulations requiring highly refined grids, for which high-order schemes are expected to be significantly more efficient than second-order schemes, we will discuss two techniques to achieve genuine third-order accuracy on regular grids.

4. A Chain-Ruled Flux-Solution-Reconstruction Scheme with $\kappa = 1/3$ (CFSR3)

In this section, we discuss the U-MUSCL scheme made genuinely third-order accurate on regular grids by flux reconstruction. It is similar to the third-order edge-based method, but different in that it is third-order accurate only on regular grids with point-valued source and time-derivative terms whereas the third-order edge-based method is third-order on arbitrary simplex-element grids and requires special source quadrature formulas.

4.1. One dimension

We consider the finite-difference-type U-MUSCL scheme (3.15) in Section 3.4. This scheme cannot be third-order accurate for nonlinear equations because of the second-order error generated from the averaged flux term in the numerical flux (3.3), not the dissipation [12, 15, 16]. To eliminate this error, we need to reconstruct the flux directly, e.g., by applying the U-MUSCL flux reconstruction with a parameter θ :

$$\phi_{i+1/2} = \frac{1}{2}[f_L + f_R] - \frac{1}{2}\left|\frac{\partial f}{\partial u}\right|_{j+1/2}(u_R - u_L), \quad (4.1)$$

where f_L and f_R are given by

$$f_L = f_i + \frac{1}{4}[(1-\theta)\Delta^- + (1+\theta)\Delta^+], \quad f_R = f_{i+1} - \frac{1}{4}[(1-\theta)\Delta^+ + (1+\theta)\Delta^-], \quad (4.2)$$

$$\Delta^- = f_i - f_{i-1}, \quad \Delta^+ = f_{i+1} - f_i. \quad (4.3)$$

This flux-reconstruction version has the truncation error,

$$\frac{du_i}{dt} + \frac{3\theta-1}{12}f_{xxx}h^2 + O(h^3) = s_i, \quad (4.4)$$

and therefore achieves third-order accuracy with $\theta = 1/3$. Note that u_L and u_R are computed with the κ -reconstruction (3.4) and the dissipation term does not contribute to the leading second-order error. Therefore, the choice of κ is arbitrary; the scheme is third-order as long as $\theta = 1/3$. This scheme is referred to as the flux-solution-reconstruction (FSR) scheme. It is third-order accurate for general nonlinear equations and, unlike FVC and FVP, extends to multi-dimensions.

Note that the FSR scheme is a conservative finite-difference scheme, approximating the differential form (3.1) at a point $x = x_i$, which may be the cell center. It is similar to the kappa-family finite-difference scheme of Van Leer [13], the QUICKEST scheme of Leonard [22], the schemes of Shu and Osher [29] and the NLV6 scheme [25]. However, the flux reconstruction can be very expensive in multi-dimensions. In the next section, We will discuss an efficient flux reconstruction by the chain rule.

4.2. Multi-dimensions

Consider the U-MUSCL scheme on a regular quadrilateral grid, which can be expressed as

$$\frac{d\mathbf{u}_{i,j}}{dt} + \frac{\Phi_{i,i+1} - \Phi_{i,i-1}}{h_x} + \frac{\Phi_{j,j+1} - \Phi_{j,j-1}}{h_y} = \mathbf{s}_{i,j}, \quad (4.5)$$

where (i, j) notation is used for the sake of convenience, $h_x = x_{i+1,j} - x_{i,j}$, $h_y = y_{i,j+1} - y_{i,j}$, $s_{i,j} = s(x_{i,j}, y_{i,j})$, $\Phi_{i,i+1}$ denotes the numerical flux in the x -direction across the face between the nodes (i, j) and $(i+1, j)$, and similarly for $\Phi_{i,i-1}$, and $\Phi_{j,j+1}$ and $\Phi_{j,j-1}$ are numerical fluxes in the y -direction. It is emphasized that our focus is on the unstructured-grid scheme and therefore it is not actually implemented with (i, j) data structure, which is used just for convenience. Clearly, the scheme can be third-order as a finite-difference scheme with FSR applied in each coordinate direction. It can be implemented in an unstructured-grid code with the following flux function:

$$\Phi_{jk} = \frac{1}{2}[\mathbf{f}_L + \mathbf{f}_R] - \frac{1}{2}\hat{\mathbf{D}}_n[\mathbf{u}(\mathbf{w}_R) - \mathbf{u}(\mathbf{w}_L)], \quad (4.6)$$

where the left and right fluxes are directly reconstructed,

$$\mathbf{f}_L = \theta \frac{\mathbf{f}_j + \mathbf{f}_k}{2} + (1-\theta) \left[\mathbf{f}_j + \frac{1}{2} \nabla \mathbf{f}_j \cdot (\mathbf{x}_k - \mathbf{x}_j) \right], \quad (4.7)$$

$$\mathbf{f}_R = \theta \frac{\mathbf{f}_k + \mathbf{f}_j}{2} + (1-\theta) \left[\mathbf{f}_k - \frac{1}{2} \nabla \mathbf{f}_k \cdot (\mathbf{x}_k - \mathbf{x}_j) \right], \quad (4.8)$$

where $\theta = 1/3$. Note that the third-order edge-based method requires a similar flux reconstruction but θ must be zero [30]. As in one dimension, the choice of κ for the solution reconstruction is arbitrary.

Although the flux reconstruction ensures genuine third-order accuracy, it is very expensive since the flux gradients need to be computed and stored for each flux: two and three flux vectors in two and three dimensions, respectively. The FSR scheme is, therefore, not a practical scheme; but it can be made very efficient by the chain rule [12, 31, 32]:

$$\nabla \mathbf{f}_j = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{w}} \right)_j \nabla \mathbf{w}_j, \quad \nabla \mathbf{f}_k = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{w}} \right)_k \nabla \mathbf{w}_k, \quad (4.9)$$

where the flux Jacobian is given for the Euler equations, with the unit directed-area vector $\hat{\mathbf{n}}_{jk}$ and \mathbf{v} defined as column vectors and the notation $u_n = \mathbf{v} \cdot \hat{\mathbf{n}}_{jk}$, as

$$\frac{\partial \mathbf{f}}{\partial \mathbf{w}} = \begin{bmatrix} u_n & \rho \hat{\mathbf{n}}_{jk}^t & 0 \\ u_n \mathbf{v} & \rho(u_n \mathbf{I} + \mathbf{v} \otimes \hat{\mathbf{n}}_{jk}) & \hat{\mathbf{n}}_{jk} \\ u_n \mathbf{v}^2 / 2 & \rho(H \hat{\mathbf{n}}_{jk}^t + u_n \mathbf{v}^t) & \gamma u_n / (\gamma - 1) \end{bmatrix}. \quad (4.10)$$

In this way, we do not need to compute and store multiple fluxes and their gradients in two and three dimensions; it dramatically reduces the cost of flux reconstruction. In this paper, the U-MUSCL scheme with the efficient flux reconstruction is referred to as the third-order chain-rule based flux-solution-reconstruction scheme (CFSR3). Extensions to higher-order accuracy are possible, which will be discussed in a subsequent paper.

5. A Special Third-Order U-MUSCL Scheme with $\kappa = 1/2$ (U-MUSCL-SSQ)

In this section, we present a unique way of achieving genuine third-order accuracy with the U-MUSCL reconstruction scheme based on vanishing residuals. It is similar to the third-order edge-based method, but different in that flux reconstruction is not necessary and it is third-order accurate only on regular grids whereas the third-order edge-based method is third-order on arbitrary simplex-element grids and requires flux reconstruction.

5.1. One dimension

Consider the typical U-MUSCL scheme (3.15) and its truncation error (3.16), it is noted that $\kappa = 1/2$ reconstructs a quadratic solution exactly at a face from point-valued solutions stored at nodes [16]. Then, the U-MUSCL scheme (3.15) is essentially equivalent to the second-order central finite-difference scheme; indeed, for $\kappa = 1/2$, the truncation error (3.16) becomes

$$\frac{du_i}{dt} + f_x = s_i - \frac{1}{24} [f_{uuu}(u_x)^3 + 3f_{uuu}u_xu_{xx} + f_uu_{xxx}] h^2 + O(h^3). \quad (5.1)$$

which simplifies, by $f_{xxx} = f_{uuu}(u_x)^3 + 3f_{uuu}u_xu_{xx} + f_uu_{xxx}$, to

$$\frac{du_i}{dt} + f_x = s_i - \frac{1}{24} f_{xxx} h^2 + O(h^3). \quad (5.2)$$

This is clearly a second-order scheme. However, if we were able to discretize the time derivative and source terms in such a way (rather than the point evaluation) that the second-order truncation error is factored:

$$\frac{du_i}{dt} + f_x = s_i - \frac{1}{24} \left(\frac{du}{dt} + f_x - s \right)_{xx} h^2 + O(h^3), \quad (5.3)$$

then, since $\frac{du_i}{dt} + f_x - s_i = 0$ for an exact solution, we would obtain a third-order scheme:

$$\frac{du_i}{dt} + f_x = s_i + O(h^3). \quad (5.4)$$

We have discovered such a source discretization formula, which is given by

$$s_i = s(x_j) + \frac{\kappa_s}{4} [s(x_{i+1}) - 2s(x_i) + s(x_{i-1})], \quad (5.5)$$

where

$$\kappa_s = \frac{1}{6}, \quad (5.6)$$

and similarly for the time derivative term. It can be expressed with the U-MUSCL reconstruction formula as

$$s_i = \frac{s_{i+1/2} + s_{i-1/2}}{2}, \quad (5.7)$$

where

$$s_{i-1/2} = \kappa_s \frac{s(x_{i-1}) + s(x_i)}{2} + (1 - \kappa_s) \left[s(x_i) - s_x(x_i) \frac{h}{2} \right], \quad (5.8)$$

$$s_{i+1/2} = \kappa_s \frac{s(x_{i+1}) + s(x_i)}{2} + (1 - \kappa_s) \left[s(x_i) + s_x(x_i) \frac{h}{2} \right], \quad (5.9)$$

$$s_x(x_i) = \frac{s(x_{i+1}) - s(x_{i-1})}{2h}. \quad (5.10)$$

Note that the above source discretization is compact with a three-point stencil. This scheme does not seem to have been known; it is here referred to as U-MUSCL-SSQ. Remarkably, the error-cancellation property extends to higher dimensions as we will discuss in the next section.

5.2. Multi-dimensions

Consider, for simplicity, a scalar nonlinear equation with a source term, $\partial_x f(u) + \partial_y g(u) = s$, and its edge-based discretization with an extended special source discretization:

$$Res_j = \frac{1}{V_j} \sum_{k \in \{k_j\}} \phi_{jk} A_{jk} - \tilde{s}_j, \quad \tilde{s}_j = \frac{1}{V_j} \sum_{k \in \{k_j\}} \psi_{jk} V_{jk}, \quad (5.11)$$

where, with $f_n = (f, g) \cdot \hat{\mathbf{n}}_{jk}$ and $D_n = \partial f_n / \partial u$,

$$\phi_{jk} = \frac{1}{2} [f_n(u_L) + f_n(u_R)] - \frac{1}{2} \hat{D}_n(u_R - u_L), \quad (5.12)$$

$$V_{jk} = \frac{1}{4} (\mathbf{x}_k - \mathbf{x}_j) \cdot \mathbf{n}_{jk}, \quad \psi_{jk} = \kappa_s \frac{s_j + s_k}{2} + (1 - \kappa_s) \left[s_j + \frac{1}{2} \bar{\nabla} s_j \cdot (\mathbf{x}_k - \mathbf{x}_j) \right]. \quad (5.13)$$

Note that the gradients are computed by a linear LSQ method, which is sufficient for an exact quadratic extrapolation on regular grids. For regular quadrilateral grids, this scheme reduces to the one-dimensional scheme applied in each coordinate direction. Therefore, $\kappa_s = 1/6$ gives third-order accuracy.

This scheme achieves third-order accuracy also for regular triangular grids but with a different value of κ_s . To see this, we substitute a smooth solution into the residual and expand it on a triangular grid with equilateral triangles of side length h and obtain, for $\kappa = 1/2$,

$$Res_j = \partial_x f + \partial_y g - s + \frac{1}{32} [\partial_{xx}(\partial_x f + \partial_y g) + \partial_{yy}(\partial_x f + \partial_y g)] h^2 - \frac{\kappa_s}{8} [\partial_{xx}s + \partial_{yy}s] h^2 + O(h^3), \quad (5.14)$$

which suggests that we set $\kappa_s = 1/4$, so that we have

$$Res_j = r + \frac{1}{32} [\partial_{xx}r + \partial_{yy}r] h^2 + O(h^3), \quad r = \partial_x f + \partial_y g - s. \quad (5.15)$$

This gives the truncation error TE_j when the smooth solution is exact solution satisfying $r = 0$:

$$TE_j = O(h^3). \quad (5.16)$$

The result is valid for a time-dependent equation with the time derivative treated as a source term and also for a general regular grid with each node having edge-connected nodes in a symmetric configuration (e.g., a regular quadrilateral split into two right triangles). For fully irregular grids, the U-MUSCL-SSQ scheme loses third-order accuracy. Nevertheless, as we will show later, it can be a very accurate second-order scheme compared with other second-order schemes.

6. Exact quadratic reconstruction with $\kappa = 1/2$ on Arbitrary Grids

As it was known (but a proof was never provided) [31], the U-MUSCL reconstruction scheme by itself can be high-order on arbitrary grids: it can be exact for quadratic functions if the nodal gradients are computed by an algorithm exact for quadratic functions, e.g., by a quadratic LSQ method. To prove this, let us consider a quadratic function $u_{\text{quadratic}}(x, y)$ defined around a node j (or a cell center):

$$\begin{aligned} u_{\text{quadratic}}(\mathbf{x}) &= u_j + (x - x_j)\partial_x u_j + (y - y_j)\partial_y u_j + \frac{1}{2}(x - x_j)^2\partial_{xx} u_j \\ &\quad + (x - x_j)(y - y_j)\partial_{xy} u_j + \frac{1}{2}(y - y_j)^2\partial_{yy} u_j, \end{aligned} \quad (6.1)$$

where all the derivatives involved in the above expression are constants defined at j . Then, the function value at a neighbor node k is exactly expressed by

$$u_k = u_{\text{quadratic}}(\mathbf{x}_k) = u_j + \partial_{jk} u_j + \frac{1}{2}\partial_{jk}^2 u_j, \quad (6.2)$$

where we have introduced the notation:

$$\partial_{jk} \equiv (\mathbf{x}_k - \mathbf{x}_j) \cdot (\partial_x, \partial_y). \quad (6.3)$$

Assume now that gradients are computed by a quadratic LSQ method and thus can be written exactly as

$$\bar{\nabla} u_j = \nabla u_j, \quad \bar{\nabla} u_k = \nabla u_j + \partial_{jk} \nabla u_j. \quad (6.4)$$

Then, substituting these exact expressions into the U-MUSCL reconstruction schemes, we find

$$u_L = u_j + \frac{\kappa}{2}(u_k - u_j) + \frac{1-\kappa}{2}\nabla u_j \cdot (\mathbf{x}_k - \mathbf{x}_j) \quad (6.5)$$

$$= u_j + \frac{\kappa}{2}(\partial_{jk} u_j + \frac{1}{2}\partial_{jk}^2 u_j) + \frac{1-\kappa}{2}\partial_{jk} u_j \quad (6.6)$$

$$= u_j + \frac{1}{2}\partial_{jk} u_j + \frac{\kappa}{4}\partial_{jk}^2 u_j, \quad (6.7)$$

and for u_R ,

$$u_R = u_k + \frac{\kappa}{2}(u_j - u_k) + \frac{1-\kappa}{2}\nabla u_k \cdot (\mathbf{x}_j - \mathbf{x}_k) \quad (6.8)$$

$$= (u_j + \partial_{jk} u_j + \frac{1}{2}\partial_{jk}^2 u_j) - \frac{\kappa}{2}(\partial_{jk} u_j + \frac{1}{2}\partial_{jk}^2 u_j) + \frac{1-\kappa}{2}(\nabla u_j + \partial_{jk} \nabla u_j) \cdot (\mathbf{x}_j - \mathbf{x}_k) \quad (6.9)$$

$$= u_j + \left(1 - \frac{\kappa}{2} - \frac{1-\kappa}{2}\right)\partial_{jk} u_j + \left(\frac{1}{2} - \frac{\kappa}{4} - \frac{1-\kappa}{2}\right)\partial_{jk}^2 u_j \quad (6.10)$$

$$= u_j + \frac{1}{2}\partial_{jk} u_j + \frac{\kappa}{4}\partial_{jk}^2 u_j, \quad (6.11)$$

which have just proved two things. First, the jump at the face vanishes for quadratic functions on arbitrary grids for any κ :

$$u_R - u_L = \left(u_j + \frac{1}{2}\partial_{jk} u_j + \frac{\kappa}{4}\partial_{jk}^2 u_j\right) - \left(u_j + \frac{1}{2}\partial_{jk} u_j + \frac{\kappa}{4}\partial_{jk}^2 u_j\right) = 0. \quad (6.12)$$

Note that if the gradients are computed by a linear LSQ method, then the jump can vanish for a quadratic function only with $\kappa = 1$, which makes the jump identically zero for any function. Second, the U-MUSCL scheme reconstructs a quadratic function exactly on arbitrary grids with $\kappa = 1/2$:

$$u_L = u_R = u_{\text{quadratic}}\left(\frac{\mathbf{x}_j + \mathbf{x}_k}{2}\right) = u_j + \frac{1}{2}\partial_{jk} u_j + \frac{1}{8}\partial_{jk}^2 u_j. \quad (6.13)$$

Note that this is true for the reconstruction at half way between two nodes; the same will be true even when it is applied to a cell-centered scheme as long as the reconstruction is performed at the midpoint of two adjacent centroids.

Note that $\kappa = 1/2$ reconstructs the point-valued solution at the edge-midpoint from the point-valued solutions stored at nodes. This does not lead to a third-order scheme for two reasons. First, on a regular grid, it will result in the second-order central scheme as discussed in Section 5 (thus, it cannot be third-order without the special source quadrature formula). Remember also that a finite-volume scheme with point-valued solutions can be third-order with $\kappa = 1/2$ but it does not extend to higher dimensions. Second, on irregular grids, the edge-based flux quadrature formula is exact for quadratic fluxes (thus third-order) only with $\kappa = 0$ [30] and only for simplex-element grids. Nevertheless, numerical results indicate that $\kappa = 1/2$ gives superior performance over other values as we will show later.

Incidentally, the ability to quadratically interpolate the solution at the edge-midpoint is relevant to the third-order residual-distribution (or equivalently continuous Galerkin or stabilized finite element methods) scheme originally developed by Caraeni and Fuchs [33, 34]. In their method, they reconstruct a P_2 simplex element from a P_1 simplex element by interpolating the solution from two nodes of an edge to the edge-midpoint, and then calculate the element residual by integrating the governing equations of interest over the reconstructed P_2 element. Their interpolation formula for the edge-midpoint solution u_m is equivalent to the arithmetic average of the U-MUSCL reconstruction schemes with $\kappa = 1/2$:

$$u_m = \frac{u_L + u_R}{2} = \frac{u_j + u_k}{2} - \frac{1}{8} (\bar{\nabla} u_k - \bar{\nabla} u_j) \cdot (\mathbf{x}_k - \mathbf{x}_j), \quad (6.14)$$

This technique was later used by Nishikawa, Rad, and Roe for developing high-order fluctuation-splitting schemes for Cauchy-Riemann, inviscid, viscous equations [35–39].

7. U-MUSCL in Cell-Centered Methods

The U-MUSCL reconstruction scheme is very simple to implement and often employed in cell-centered finite-volume codes also. However, it loses second-order accuracy (not to mention third-order) when the face centroid is not located exactly halfway between two adjacent centroids. Ref.[19] demonstrates that the loss of second-order accuracy results in serious accuracy deterioration on unstructured grids, and proposes a suitable modification. A similar modification has been proposed also in Ref.[40] for preserving second-order accuracy on adaptive Cartesian grids. Burg mentioned this problem in his original paper [4], but no modification techniques were proposed.

To illuminate this problem, we consider the U-MUSCL reconstruction scheme in the form shown previously:

$$u_L = \kappa \frac{u_j + u_k}{2} + (1 - \kappa) [u_j + \bar{\nabla} u_j \cdot (\mathbf{x}_{f_c} - \mathbf{x}_j)], \quad (7.1)$$

where \mathbf{x}_{f_c} denotes the face centroid position. It is clear in this form that the U-MUSCL scheme is a linear combination of the linear interpolation (the first term) and the linear extrapolation (the second term). Then, we see that the linear interpolation will not be exact for linear functions on unstructured grids, where the face centroid does not coincide with the midpoint of the two data points (nodes or cell centroids) at j and k . As a result, second-order accuracy is lost. As suggested in Ref.[19], one way to recover second-order accuracy is to replace u_k by a linearly-exact value u_p :

$$u_L = \kappa \frac{u_j + u_p}{2} + (1 - \kappa) [u_j + \nabla u_j \cdot (\mathbf{x}_{f_c} - \mathbf{x}_j)], \quad (7.2)$$

where

$$u_p = u_k + \nabla u_k \cdot (\mathbf{x}_{f_c} - \mathbf{x}_j) - \mathbf{x}_k. \quad (7.3)$$

This modification makes the reconstruction exact for linear functions on arbitrary grids. A similar modification needs to be applied to the right state u_R also. See Ref.[19] for further details and its impact on accuracy in unstructured grids.

A cell-centered finite-volume method with the U-MUSCL reconstruction scheme can achieve third-order accuracy in the same way as the edge-based method on regular quadrilateral grids. Therefore, as we have concluded in Section 3.6, all the U-MUSCL-based cell-centered finite-volume codes are second-order accurate at best for any value of κ for the nonlinear Euler equations.

Type	κ	θ	Solution	Reconstruction	s and $\frac{\partial u}{\partial t}$	Order of Accuracy	
						Linear	Nonlinear
U-MUSCL: FVC	1/3	–	cell average	solution	cell average	$O(h^3)$	$O(h^3)$
U-MUSCL: FVP	1/2	–	point value	solution	cell average	$O(h^3)$	$O(h^3)$
U-MUSCL: FD	1/3	–	point value	solution	point value	$O(h^3)$	$O(h^2)$
FSR	\mathbb{R}	1/3	point value	flux and solution	point value	$O(h^3)$	$O(h^3)$
U-MUSCL-SSQ	1/2	–	point value	solution	quadrature (5.7)	$O(h^3)$	$O(h^3)$

Table 1: Third-order schemes in the form of the U-MUSCL scheme in one dimension. \mathbb{R} indicates any real value.

One can apply CFSR3 or U-MUSCL-SSQ and achieve third-order accuracy on regular quadrilateral grids. It is emphasized that third-order accuracy will be obtained in the point-valued solution, not in the cell-averaged solution. Moreover, time derivatives and source terms must be evaluated at the cell center (not integrated) for CFSR3 and discretized with the special formula for U-MUSCL-SSQ. Hence, the scheme must be treated as a finite-difference scheme with point-valued solutions, rather than a finite-volume scheme. It is also pointed out that third-order accuracy remains to be demonstrated for regular triangular grids, in which the cell-centered stencil is not the same everywhere (there are two different configurations) and therefore it is not clear if error cancellations occur. Studies on cell-centered schemes are left as future work.

8. Results

8.1. Numerical tests in one dimensions

We present numerical results for third-order schemes in one dimension. The finite-volume scheme with cell averaged solutions as discussed in Section 3.2 will be referred to as FVC. Similarly, FVP denotes the finite-volume scheme with point valued solutions in Section 3.3. The U-MUSCL scheme in the form (3.15) is simply referred to as U-MUSCL. These schemes are summarized along with FSR and U-MUSCL-SSQ in Table 1.

8.1.1. Accuracy verification with steady problems

We first consider a steady state problem for Burgers' equation: with the exact solution $u(x) = C \exp(ax)$ satisfying

$$\partial_x f = s(x), \quad (8.1)$$

where $f = u^2/2$ and $s(x) = aC^2 \exp(2ax)$, so that the exact solution is given by

$$u(x) = C \exp(ax), \quad (8.2)$$

where we set $C = 1.57$ and $a = 1.23$. We solve the problem over a unit domain in a series of uniform grids with n points, where $n = 16, 32, 64, 128$, which are taken to be the cell centers. The uniform grid spacing is denoted by $h = 1/(n-1)$. To focus on the accuracy in the interior, we specify the exact solution, point value or cell average,

$$\bar{u}_i = \frac{1}{h} \int_{x_i-h/2}^{x_i+h/2} u dx = \frac{C}{ah} [\exp(a(x_i + h/2)) - \exp(a(x_i - h/2))], \quad (8.3)$$

at the left boundary point and its neighbor, and similarly for the right boundary. The forcing term $s(x)$ is discretized either by a point evaluation $s(x_i)$ or a cell average:

$$\bar{s}_i = \frac{1}{h} \int_{x_i-h/2}^{x_i+h/2} s dx = \frac{C^2}{2h} [\exp(2a(x_i + h/2)) - \exp(2a(x_i - h/2))]. \quad (8.4)$$

First, three schemes are considered: FVC(1/3), FVP(1/2), and FSR(1/3). Note again that FVP(1/2) is the scheme Burg used for accuracy verification of U-MUSCL for a one-dimensional problem in his paper [4]. We compute

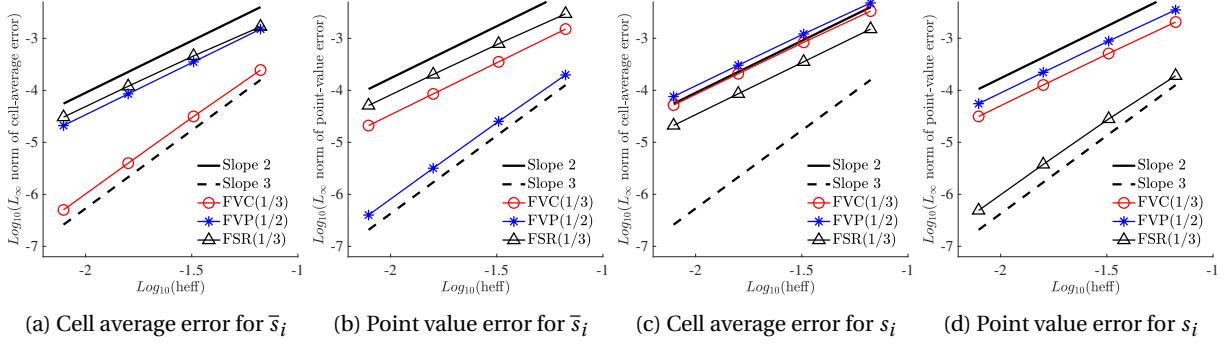


Figure 2: Error convergence results for the steady Burgers equation: $\partial_x f = s$, where $f = u^2/2$, with the cell-averaged source term in (a) and (b), and with the point-value source term in (c) and (d)

steady state solutions by iterating with the explicit SSP RK3 time-stepping scheme [41] until the residual is reduced by seven orders of magnitude.

Results with the cell-averaged source are shown in Figures 2(a) and 2(b). In Figure 2(a), the discretization errors computed against the exact cell-averaged solution are presented. As expected, FVC(1/3) is third-order accurate; and others are second-order accurate. Figure 2(a) shows the discretization errors computed against the exact point-valued solution at the cell center. Here, as expected, only FVP(1/2) gives third-order accuracy. Results with the point-valued source are shown in Figures 2(c) and 2(d). In this case, FSR(1/3) is third-order accurate in the point-valued solution; others are second-order accurate.

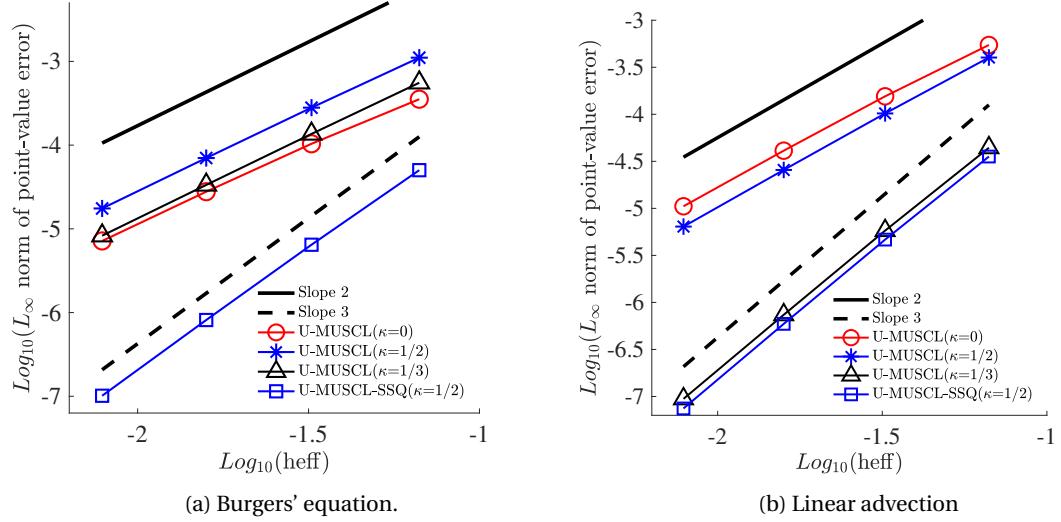


Figure 3: Error convergence results of the U-MUSCL scheme (3.15) with various values of κ for steady Burger's and linear advection equations.

We then tested the U-MUSCL scheme (3.15) without flux reconstruction, which is the form typically used in practical solvers, for $\kappa = 0, 1/3, 1/2$, and also the U-MUSCL-SSQ scheme, which is with $\kappa = 1/2$ and the special source term (5.7). To demonstrate third-order accuracy of U-MUSCL for linear equations, we performed the test also for a linear advection equation with $f = u$. Results are shown in Figure 3. As expected, the U-MUSCL scheme with $\kappa = 1/3$ is third-order accurate for the linear equation but second-order accurate for the nonlinear Burgers equation. On the other hand, the U-MUSCL-SSQ scheme is third-order accurate for both Burgers' and linear advection equations.

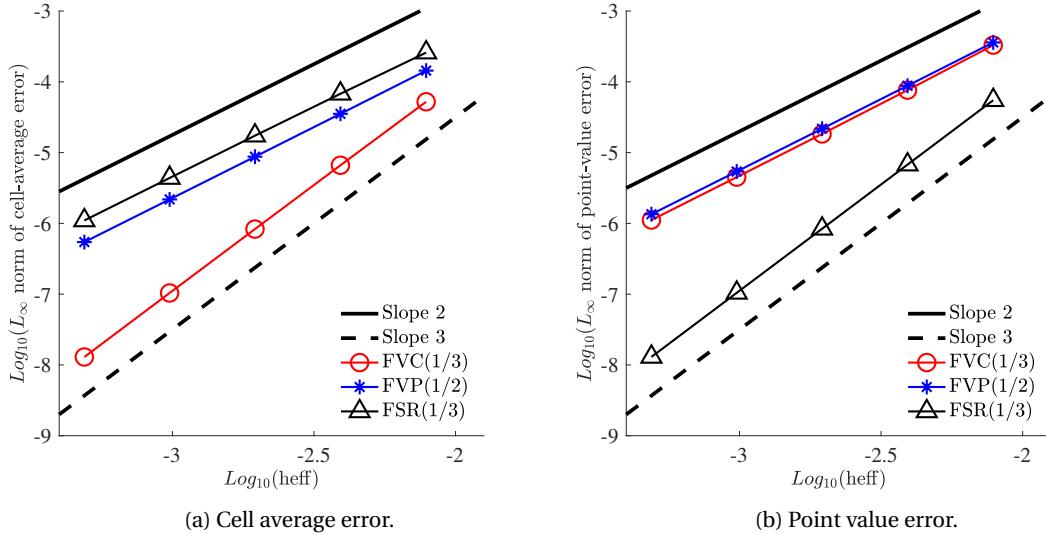


Figure 4: Error convergence results for the unsteady Burgers equation: $\partial_t u + \partial_x f = 0$, where $f = u^2/2$.

8.1.2. Accuracy verification with unsteady problems

Next, we consider an unsteady problem for Burgers' equation in the domain $x \in [0, 1]$ for a series of uniform grids with $2^{m+6}-1$ cells, where $m = 1, 2, 3, 4, 5$. The initial solution is set to be $\sin(2\pi x)$. The time integration is performed with the explicit SSP RK3 scheme for 800 time steps with $\Delta t = 0.0001$ to the final time $t_f = 0.08$. The left and right boundaries are made periodic; thus there is no boundary. The exact solution (a moment before a shock is formed) is computed by iteratively solving $u_i = \sin(2\pi(x_i - u_i t_f))$ at the cell center $x = x_i$ and the exact cell average is computed by a three-point Gaussian quadrature formula with the exact cell center solution and the exact solutions at the left and right faces of the cell.

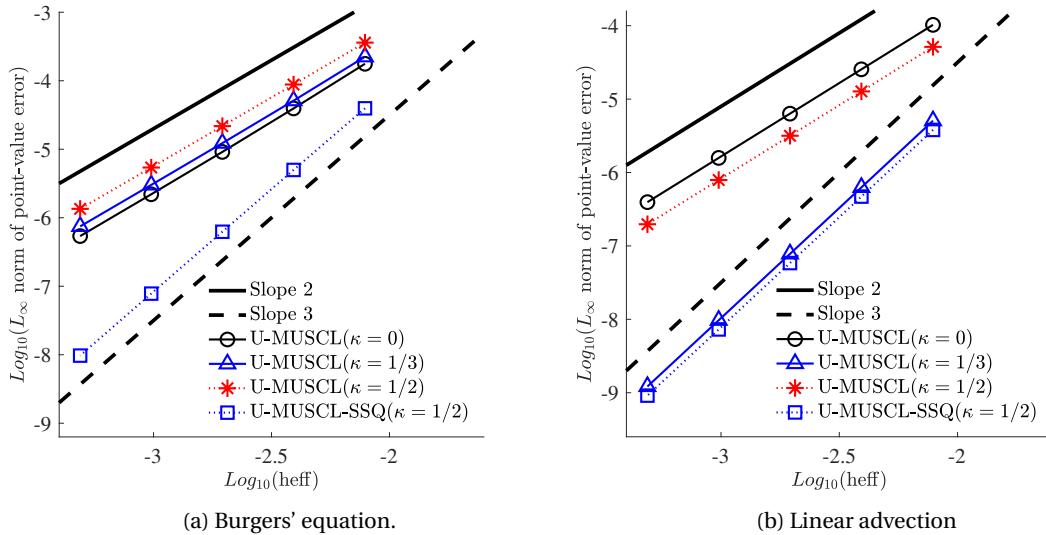


Figure 5: Error convergence results of the U-MUSCL scheme (3.15) with various values of κ for unsteady Burger's and linear advection equations.

We tested the same three schemes but simply by integrating the form (3.6) with $s_i = 0$, as typically done in the U-MUSCL scheme. Results are shown in Figure 4. As expected, FVC(1/3) achieves third-order accuracy in the cell-averaged solution and FSR(1/3) achieves third-order accuracy in the point-valued solution. FVP(1/2) is only second-order accurate because the coupling terms in the time derivative as in Equation (3.12) are ignored. This

means that the U-MUSCL scheme in the typical form (i.e., without a consistent spatial discretization of the time derivative term or without the flux reconstruction) cannot be third-order accurate.

Finally, we performed the same unsteady calculations with the U-MUSCL scheme (3.15) with $\kappa = 0$, $\kappa = 1/3$, and $\kappa = 1/2$, and U-MUSCL-SSQ for the nonlinear Burgers equation and a linear advection equation: $\partial_t u + \partial_x f = 0$, where $f = u^2/2$ and $f = u$, respectively. The same exact solution as before was used in both cases. In the linear case, the initial sine wave is simply convected to the right.

As expected, the U-MUSCL scheme is second-order accurate when applied to the nonlinear Burgers equation as shown in Figure 5(a), but achieves third-order accuracy with $\kappa = 1/3$ for the linear advection equation as shown in Figure 5(b). On the other hand, U-MUSCL-SSQ is third-order accurate for both linear and nonlinear equations.

Not shown, but the FVP scheme gives third-order accuracy if the time derivatives are discretized in a compatible manner as demonstrated in Ref[16]. To the best of the authors' knowledge, such compatible time-derivative discretizations, including that of U-MUSCL-SSQ, are never used with the U-MUSCL scheme in practical codes.

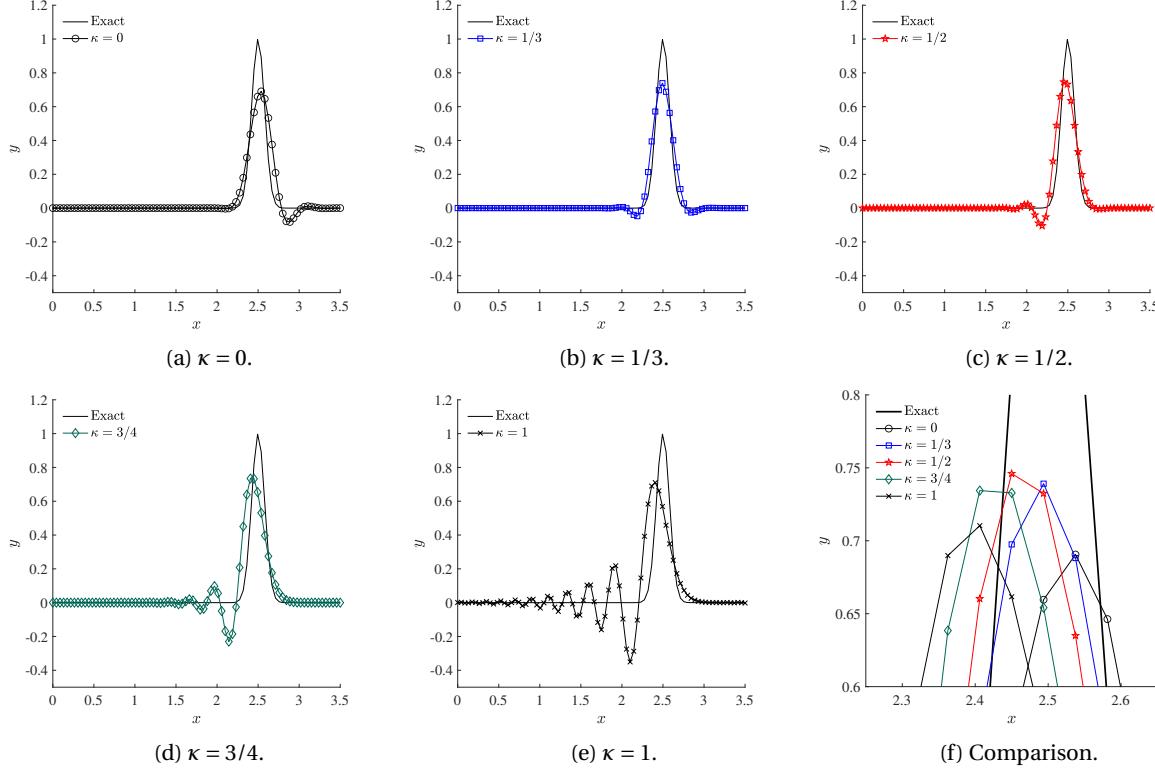


Figure 6: Numerical solutions obtained with various values of κ for the linear advection: $\partial_t u + \partial_x u = 0$ at the final time $t = 2.0$ on a uniform grid with 81 nodes.

8.1.3. Dissipative and dispersive errors

We provide a qualitative comparison among various values of κ for a linear advection equation $\partial_t u + a\partial_x u = 0$ with a Gaussian pulse: $u(x) = \exp(-80(x - 1/2)^2)$ traveling to the right at a speed $a = 1$ to the final time $t = 2.0$. The grid is a uniform grid with 81 nodes. The time integration is performed with the third-order SSP Runge-Kutta scheme as before with CFL= 0.1.

Results are shown in Figure 6. We observe severe dispersive errors towards $\kappa = 1$ (the central scheme); high-frequency modes travel at wrong speeds and lag behind. This is a typical behavior of second-order schemes whose leading error is dispersive. The most accurate solution is obtained with $\kappa = 1/3$, which is third-order for linear problems and thus eliminates the leading dispersive error of second-order schemes. Hence, it is not necessarily less dissipative than second-order schemes. To reduce dissipation, one has to employ a fourth-order scheme or a low-dissipation numerical flux. The latter has been demonstrated as a practical approach in Ref.[42]. It is important to note also that less dissipative schemes with κ closer to 1 does not necessarily better preserve the peak of

the Gaussian pulse. See Figure 6(f). The peak is better kept with $\kappa = 1/2$ or $\kappa = 1/3$, not with the zero dissipation scheme of $\kappa = 1$. We will have a similar observation for a nonlinear problems later.

8.2. Zero jump for any κ and exact quadratic extrapolation with $\kappa = 1/2$

To verify the zero solution jump for any κ and the exact quadratic extrapolation with $\kappa = 1/2$ when gradients are computed by a quadratic LSQ method as discussed in Section 6, we consider a series of irregular triangular grids with 2304, 4096, 6400, 9216, 12544 nodes in a unit square domain (similar to the one shown in Figure 12(a)). We compute the maximum norms of the jump $|u_R - u_L|$ given by the U-MUSCL reconstruction scheme at each edge-midpoint and the error $|u_L - u_{exact}| + |u_R - u_{exact}|$, where u_{exact} is a given function evaluated at the edge-midpoint. We consider two types of functions: a sine function,

$$u(x, y) = 1 + 0.2 \sin(2.3\pi x + 2.5\pi y), \quad (8.5)$$

and a quadratic function,

$$u(x, y) = 8.75 - 1.3x + 3.7y + 2.1x^2 + 0.3xy - 7.5y^2. \quad (8.6)$$

The L_∞ norms of the solution jump and the error are computed over all edges in a given grid.

First, we employed a linear LSQ method and obtained the results shown in Figure 7. For the sine function, both the jump and the error are of $O(h^2)$ as expected. The same results are obtained for the quadratic function. Note that the jump decreases as κ increases. This is expected because the jump will vanish identically with $\kappa = 1$. Then, if we compare $\kappa = 1/3$ and $\kappa = 1/2$, we see that the latter can be slightly less dissipative.

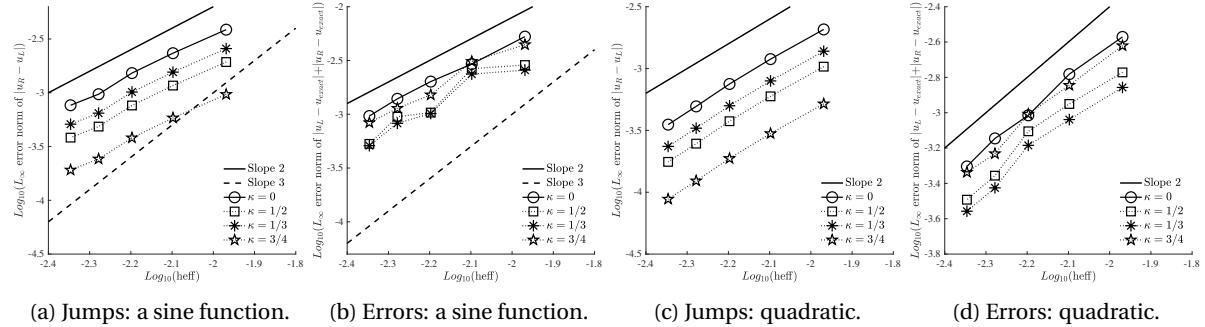


Figure 7: Solution jump and error at the edge midpoint with linear LSQ gradients.

Next, we employed a quadratic LSQ method and obtained the results shown in Figure 8. For the sine function, the jump is of $O(h^3)$ as shown in Figure 8(c); this is expected because the jump vanishes for any κ for a quadratic function, which is then numerically confirmed as shown in Figure 8(c). On the other hand, the error is $O(h^2)$ for $\kappa = 0, 1/3$, and $3/4$ but $O(h^3)$ for $\kappa = 1/2$; this is expected because the reconstruction is exact for quadratic functions with $\kappa = 1/2$ as confirmed in Figure 8(d). Again, the jump is smaller with $\kappa = 1/2$ than with $\kappa = 1/3$.

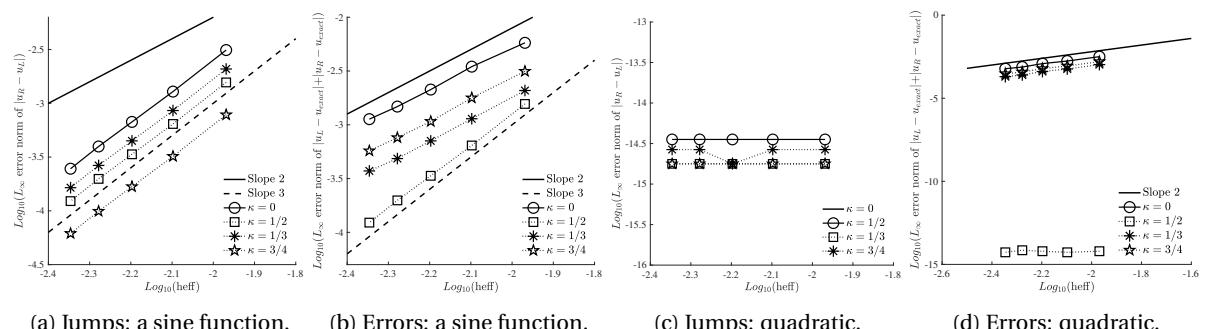


Figure 8: Solution jump and error at the edge midpoint with quadratic LSQ gradients.

Type	κ	θ	$s, \frac{\partial u}{\partial t}$	Regular quadrilaterals	Regular triangles	Irregular grids
U-MUSCL	\mathbb{R}	–	point evaluation	$O(h^2)$	$O(h^2)$	$O(h^2)$
CFSR3	\mathbb{R}	1/3	point evaluation	$\textcolor{red}{O(h^3)}$	$\textcolor{red}{O(h^3)}$	$O(h^2)$
U-MUSCL-SSQ	1/2	–	quadrature (5.11)	$\textcolor{red}{O(h^3)}, \kappa_s = 1/6$	$\textcolor{red}{O(h^3)}, \kappa_s = 1/4$	$O(h^2)$

Table 2: U-MUSCL-type schemes in two dimensions. \mathbb{R} indicates any real value. The numerical solution is a point value in all schemes. The order of accuracy is for general nonlinear equations.

8.3. Third-order accuracy verification

In this section, we present accuracy verification results obtained for the Euler equations by the method of manufactured solutions on various types of grids. We will compare the U-MUSCL scheme, CFSR4, and U-MUSCL-SSQ as summarized in Table 2. For the U-MUSCL scheme, we will further compare various values of κ . In all cases, the residual equations are solved by an implicit defect-correction solver with the exact Jacobian of the first-order residual (i.e., no LSQ gradients) relaxed by the Gauss-Seidel scheme. The Rusanov flux is employed here for robustness in the nonlinear solver, which is effective especially for $\kappa = 3/4$. The iteration is taken to be converged when the residual is reduced by six orders of magnitude in the L_1 norm. To focus on accuracy of the interior scheme, we specify the exact solution,

$$\rho = 1.0 + C \sin(\pi(2.3x + 2.5y)), \quad u = 0.15 + C \sin(\pi(2x + 2y)), \quad (8.7)$$

$$v = 0.02 + C \sin(\pi(2x + 2y)), \quad p = 1.0 + C \sin(\pi(2.3x + 2.5y)), \quad (8.8)$$

where $C = 0.2$, at boundary nodes, their neighbors, and the neighbors of the neighbors. The source term and its gradient are computed numerically and stored at each node. The discretization error will be shown for the pressure (results are similar for the other variables).

8.3.1. Regular quadrilaterals

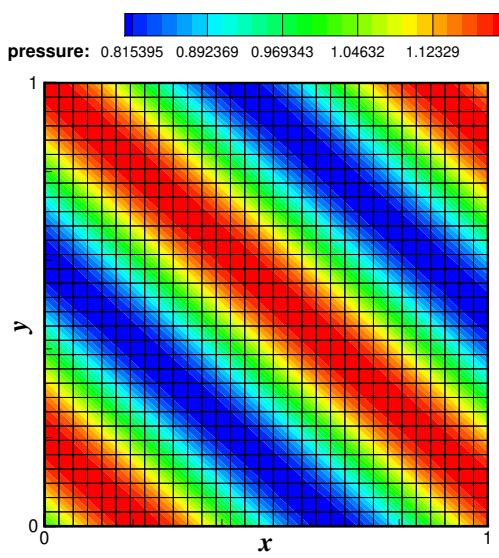
We first consider a series of regular quadrilateral grids with 1024, 2304, 4096, 6400, 9216, 12544, and 16384 nodes. Figure 9(a) shows the coarsest grid and the exact solution contours. The error convergence results are shown in Figure 9(b). As expected, CFSR3 and U-MUSCL-SSQ with $\kappa_s = 1/6$ give third-order accuracy. All others are second-order accurate. Among them, $\kappa = 1/3$ gives the lowest errors. It is close to third-order; the superior accuracy is considered due to the perturbed form of the exact solution used, which effectively linearizes the Euler equations as shown in Ref.[12], and the fact that $\kappa = 1/3$ gives third-order accuracy for linear equations. To verify this, we included results for the U-MUSCL($\kappa = 1/3$) with a smaller amplitude $C = 0.1$, as denoted by U-MUSCL($\kappa = 1/3$):0.1 in Figure 9(b). Clearly, the scheme achieves third-order accuracy. See Ref.[12] for further details. Incidentally, $\kappa = 1/3$ here corresponds to the third-order version of U-MUSCL of Yang and Harris ($\kappa_3 = 0$) [11, 21]. It is indeed an accurate scheme for problems with small perturbations.

8.3.2. Regular equilateral triangles

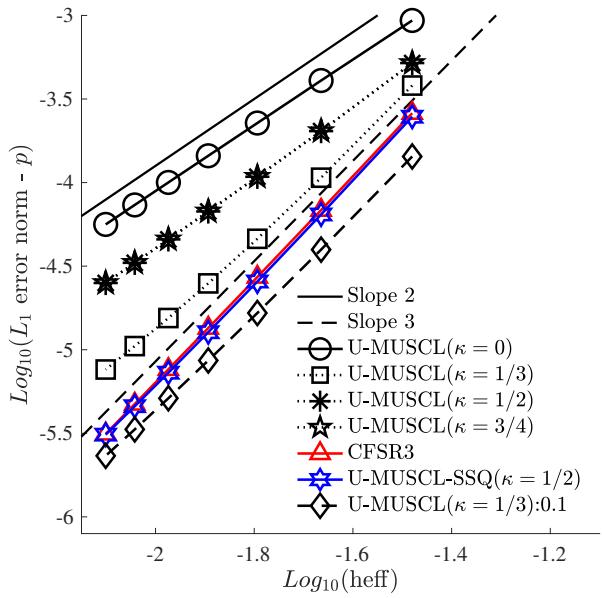
Next, we consider regular equilateral triangular grids with 1024, 2304, 4096, 6400, 12544, and 16384 nodes. The coarsest grid is shown in Figure 10(a). Error convergence results are shown in Figure 10(b). First, we see that CFSR3 and U-MUSCL-SSQ with $\kappa_s = 1/4$ achieve third-order accuracy as expected. All other schemes are second-order accurate. Again, $\kappa = 1/3$ gives the lowest second-order error.

8.3.3. Regular right-isosceles triangles

Next, we consider regular right-isosceles triangular grids with 1024, 2304, 4096, 6400, 12544, and 16384 nodes. The coarsest grid is shown in Figure 11(a). Error convergence results are shown in Figure 11(b). As expected, the results are very similar to those for the equilateral triangular grids in the previous section. CFSR3 and U-MUSCL-SSQ with $\kappa_s = 1/4$ are both third-order accurate as long as the grid is regular and triangular.

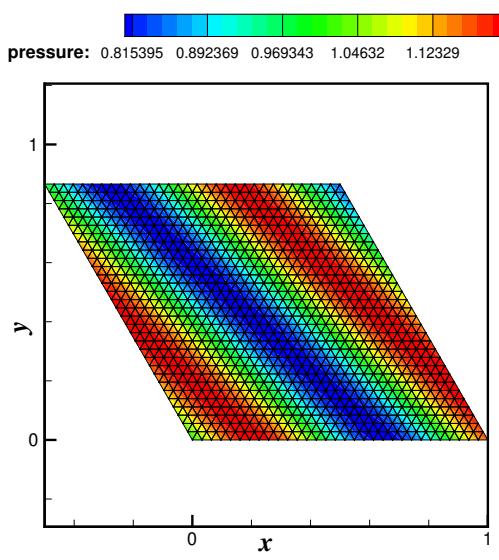


(a) The coarsest grid and the exact solution.

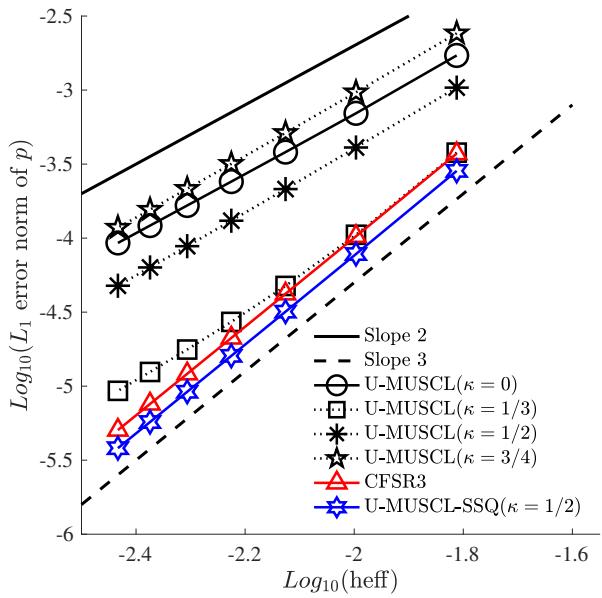


(b) Error convergence for the pressure.

Figure 9: Error convergence results for the Euler equations on regular quadrilateral grids.

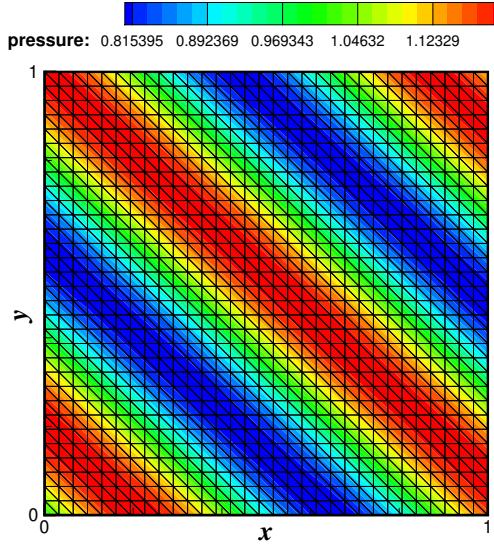


(a) The coarsest grid and the exact solution.

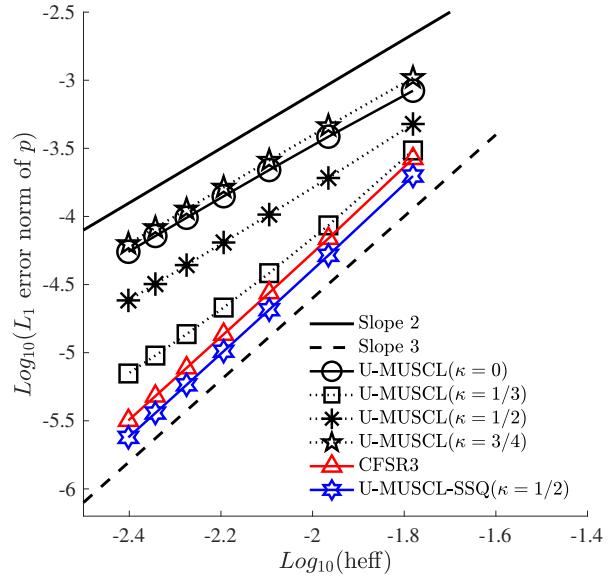


(b) Error convergence for the pressure.

Figure 10: Error convergence results for the Euler equations on equilateral triangular grids.



(a) The coarsest grid and the exact solution.

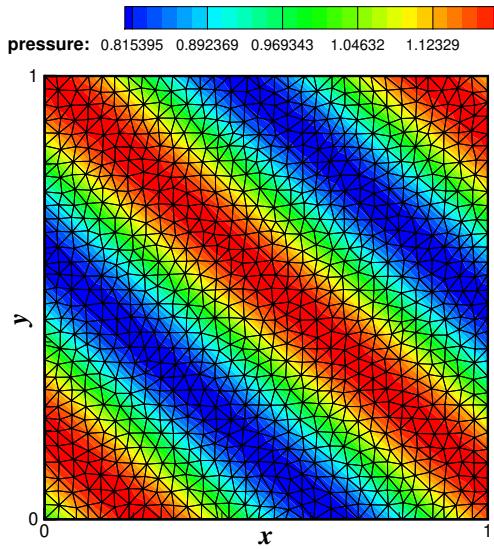


(b) Error convergence for the pressure.

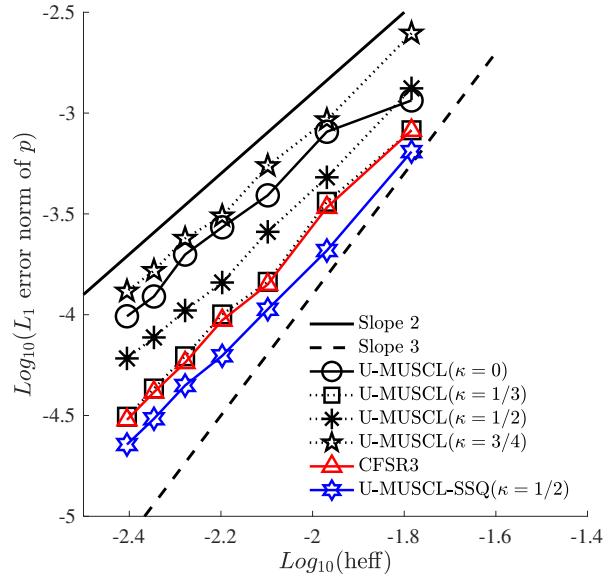
Figure 11: Error convergence results for the Euler equations on right triangular grids.

8.3.4. Irregular triangles

Next, we consider irregular triangular grids with 1024, 2304, 4096, 6400, 12544, and 16384 nodes. The coarsest grid is shown in Figure 12(a). Error convergence results are shown in Figure 12(b). In this case, no schemes achieve third-order accuracy. However, CFSR3 and U-MUSCL-SSQ still provide accurate solutions compared with others except U-MUSCL($\kappa = 1/3$), which is as accurate as CFSR3. It is interesting to note that U-MUSCL-SSQ with $\kappa_s = 1/4$ gives the lowest errors for all triangular-grid cases.



(a) The coarsest grid and the exact solution.



(b) Error convergence for the pressure.

Figure 12: Error convergence results for the Euler equations on irregular triangular grids.

8.4. Unsteady vortex transport problem in two dimensions

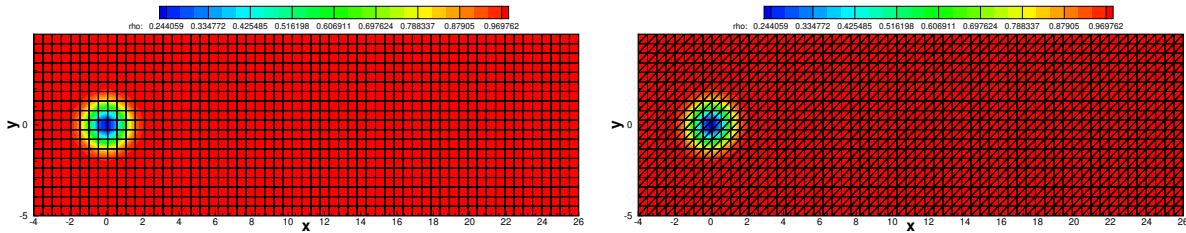
To investigate the impact of various schemes on unsteady problems, we consider an inviscid vortex-transport problem, whose solution is an exact solution to the Euler equations without source term:

$$u = u_\infty - \frac{K\bar{y}}{2\pi} \exp\left(\frac{1-\bar{r}^2}{2}\right), \quad v = v_\infty + \frac{K\bar{x}}{2\pi} \exp\left(\frac{1-\bar{r}^2}{2}\right), \quad (8.9)$$

and

$$T = 1 - \frac{K^2(\gamma-1)}{8\pi^2} \exp(1-\bar{r}^2), \quad \rho = T^{\frac{1}{\gamma-1}}, \quad p = \frac{\rho^\gamma}{\gamma}, \quad (8.10)$$

where $\bar{x} = x - u_\infty t$, $\bar{y} = y - v_\infty t$, $\bar{r}^2 = \bar{x}^2 + \bar{y}^2$, and $(u_\infty, v_\infty) = (0.5, 0.0)$. The initial solution at $t = 0$ is shown in Figure 13(a). Here, K is a key parameter. As discussed in Ref.[12], a small value of K effectively linearizes the Euler equations and thus leads to fake high-order accuracy of U-MUSCL schemes. To avoid this problem, we set $K = 6$, which is large enough. See Ref.[12] for further details. The solution is computed at the final time $t = 36.0$ with the third-order SSP Runge-Kutta time-integration scheme with a fixed time step $\Delta t = 0.0005$. For this problem, as the third-order Runge-Kutta scheme is stable even for $\kappa = 1$, the Roe flux is used.



(a) Quadrilateral grid and the initial density contours.

(b) Triangular grid and the initial density contours.

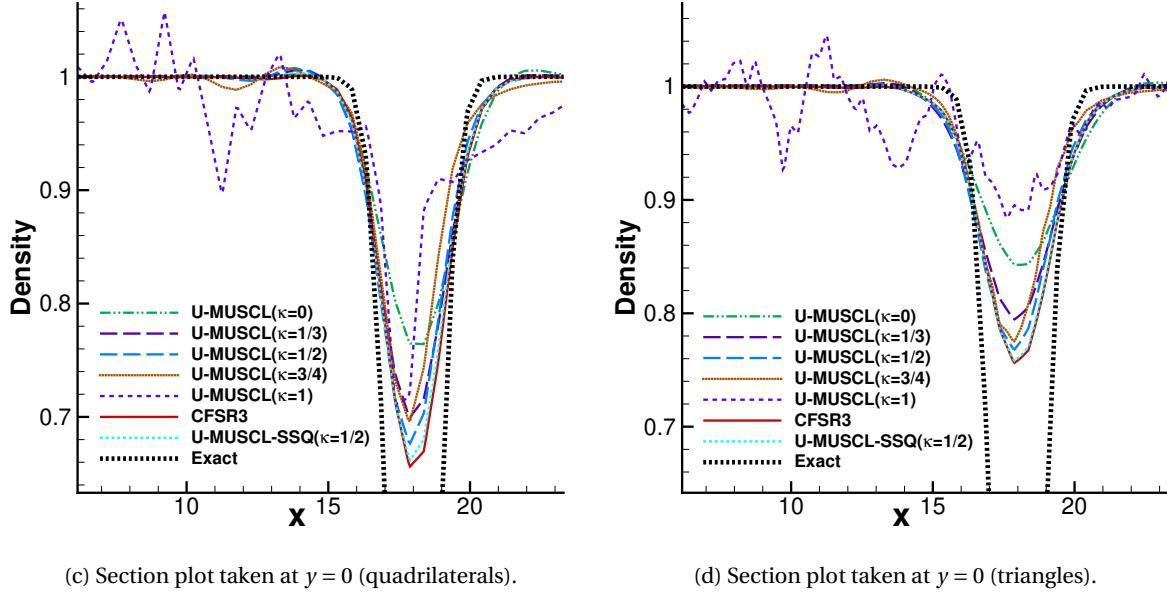


Figure 13: Vortex-transport problem on a regular quadrilateral grid.

We consider a regular quadrilateral grid and also a regular triangular grid, both of which have 60×20 nodes, as shown in Figures 13(a) and 13(b), respectively. The solutions obtained with various schemes at the final time are compared in Figures 13(c) and 13(d). For the quadrilateral grid, we observe that CFSR3 is the most accurate as expected. U-MUSCL-SSQ is also very accurate (this scheme requires the inversion of the mass matrix at each stage of the RK scheme; it took four Gauss-Seidel relaxations to reduce the linear residual by six orders of magnitude).

Among the U-MUSCL schemes, we find that unlike the previous test cases, U-MUSCL($\kappa = 1/2$) is more accurate than U-MUSCL($\kappa = 1/3$) for this fully nonlinear problem. The slightly superior performance of $\kappa = 1/2$ may be due to the less dissipative nature as discussed in Section 8.2. However, the dissipation cannot be too small. The solution obtained with $\kappa = 3/4$ has a larger dispersive error and the peak is not better maintained than $\kappa = 1/2$. Finally, the central scheme ($\kappa = 1$) suffers from severe dispersive errors as can be clearly seen.

The solutions on the triangular grid exhibit the same trend, which are more dissipative than those on the regular quadrilateral grid. The dispersive errors are significant for $\kappa = 0$, $\kappa = 3/4$, and $\kappa = 1$. On the other hand, U-MUSCL($\kappa = 1/2$) and U-MUSCL($\kappa = 1/3$) have a better dispersion property (close to the third-order scheme) although they are theoretically second-order accurate. Again, U-MUSCL($\kappa = 1/2$) keeps the peak better than $\kappa = 1/3$.

In both quadrilateral and triangular grids, the improvements by $\kappa = 1/3$, $\kappa = 1/2$, U-MUSCL-SSQ, and CFSR3 are somewhat minor. As we will show in a subsequent paper, more significant improvements can be achieved by fourth-order versions.

8.5. Unsteady separated laminar flow over a cylinder

Finally, we show a comparison of the CFSR3 scheme and the U-MUSCL scheme implemented in FUN3D with varying κ for a laminar flow over a cylinder with the Mach number 0.1 at zero angle of attack, the Reynolds number 3,900, and the free stream temperature 460.0[R]. The cylinder of diameter 1.0 has a span-wise length of 2.0 and the outer boundary is a rectangular box at the distance 100 from the center of the cylinder. The domain is taken as periodic at two planes with the minimum and maximum y -coordinates. The mesh used for these simulations is tetrahedral shown in Figure 14, and it has 8,337,840 nodes and 48,951,648 tetrahedra. Simulations were run with the default Roe flux, the Galerkin viscous discretization, and the second-order backward-difference (BDF2) time-stepping scheme with a non-dimensionalized time step of 0.05 for 21,000 time steps, with 400 cores. The CFSR3 was run with $\kappa = 1/2$ for the dissipation calculation and $\theta = 1/3$ for robustness and accuracy respectively. It is noted that a nonlinear solver diverged with $\kappa = 0$; $\kappa = 1/2$ is considered as more robust because it reduces the effect of LSQ gradients (see Eqs.(2.8) and (2.9)).

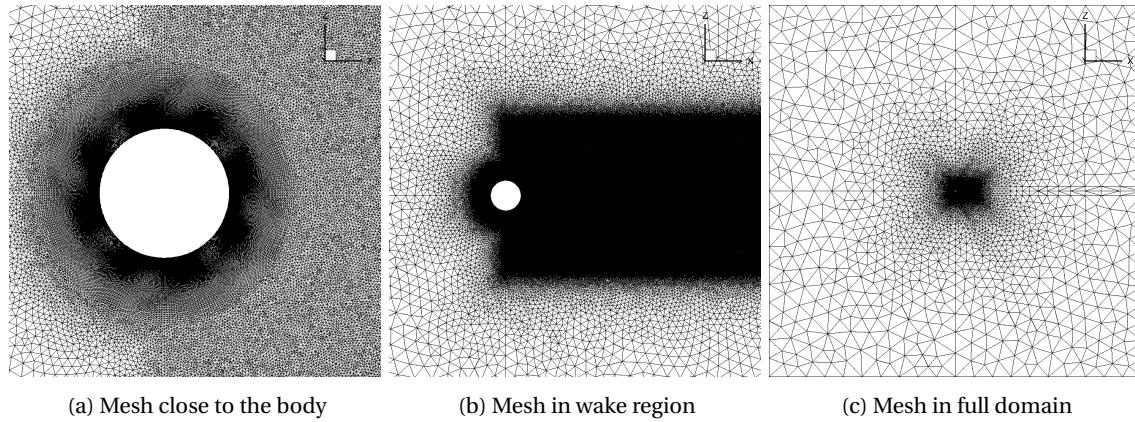


Figure 14: Tetrahedral mesh for the laminar cylinder test case

The Q-isosurface (colored by the vorticity magnitude) plots are shown in Figures 15 and 16. We can see in Figure 15 that the choice of κ does not make a great deal of difference. Nevertheless, the U-MUSCL scheme with $\kappa = 1/2$ appears to show better capturing of the near wake region behind the cylinder and better overall detail. In contrast, Figure 16 shows noticeably different structure and more detail. The most noticeable area of improvement is the near wake region, where the CFSR3 captures much more of the small scale behavior.

9. Concluding Remarks

In this paper, we have resolved confusions over third-order accuracy of the U-MUSCL scheme in the edge-based discretization. In one dimension, the U-MUSCL scheme can be third-order in the following cases:

1. Finite-volume scheme with cell-averaged solutions and source terms with $\kappa = 1/3$,

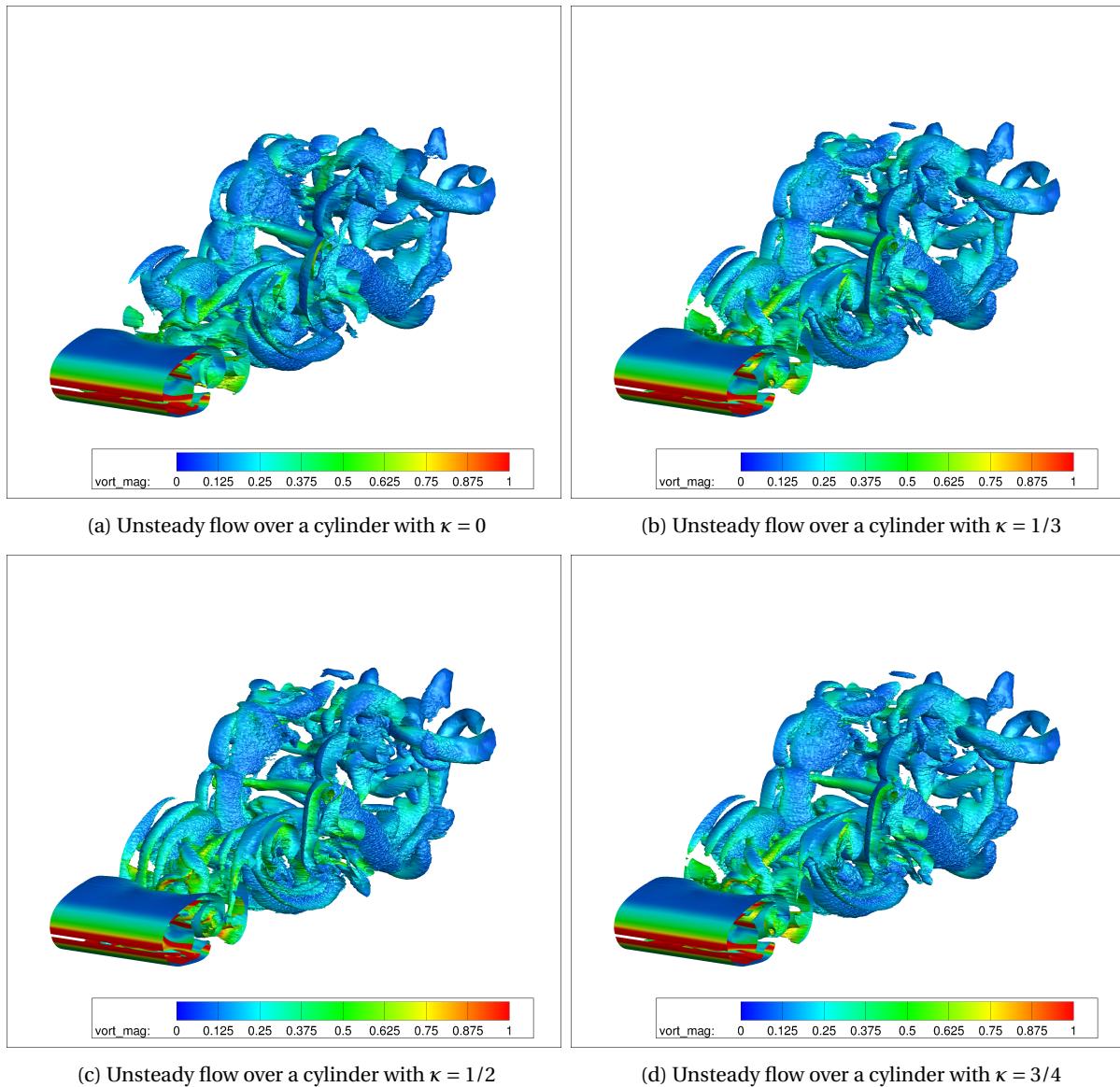


Figure 15: Unsteady separated laminar flow for varying kappa

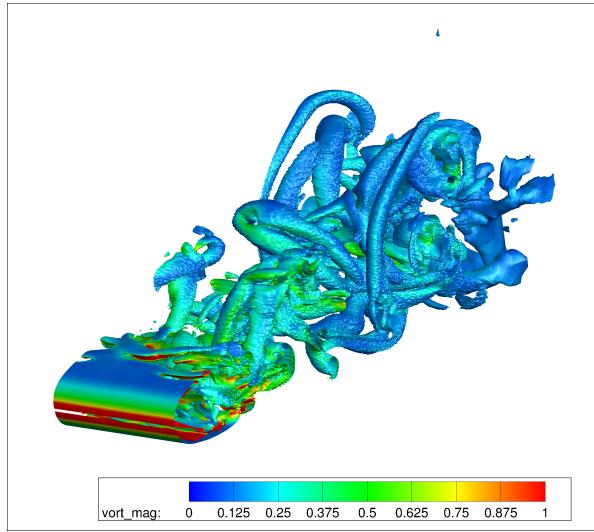


Figure 16: Unsteady flow over a cylinder with CFSR scheme $\theta = 1/3$

2. Finite-volume scheme with point-valued solutions and cell-averaged source terms with $\kappa = 1/2$,
3. Finite-difference scheme with point-valued solutions and source terms with $\kappa = 1/3$ (for linear equations only).

The first two schemes do not extend to higher dimensions. The third one achieves third-order accuracy on regular grids in higher dimensions, but only for linear equations. There is a very special case that it gives third-order accuracy as a finite-difference scheme with $\kappa = 1/2$ in multi-dimensions, but only for steady nonlinear equations without source terms. This is a special case of the proposed third-order U-MUSCL-SSQ scheme. In conclusion, the U-MUSCL scheme as typically implemented in practical CFD codes is second-order accurate at best even on regular grids for realistic applications.

In accuracy verification by the method of manufactured solutions, the U-MUSCL scheme was found more accurate with $\kappa = 1/3$ than with $\kappa = 1/2$. This is shown to be due to the perturbed form of the exact solution, which (as verified theoretically in Ref.[12]) effectively linearizes the Euler equations with small perturbation and the fact that $\kappa = 1/3$ gives third-order accuracy for linear equations. On the other hand, for an inviscid vortex transport problem, we obtained slightly less dissipative solutions with $\kappa = 1/2$ than with $\kappa = 1/3$; apparently, the superior performance is due to the low diffusion with $\kappa = 1/2$. Further reducing the dissipation with $\kappa > 1/2$ did not improve the solution due to large dispersive errors. These results indicate that $\kappa = 1/3$ is suitable for problems with small perturbations and $\kappa = 1/2$ otherwise.

It is also noted that $\kappa = 1/2$ tends to provide more robust iterative convergence on unstructured grids as it reduces the effect of LSQ gradients in the U-MUSCL solution reconstruction scheme. This could be another major reason for the popularity of $\kappa = 1/2$.

We then discussed two techniques to achieve genuine third-order accuracy when the grid is regular:

1. CFSR3: Conservative finite-difference scheme with point-valued solutions and the U-MUSCL flux and solution reconstruction at $\theta = 1/3$, as originally developed in Ref.[12]. This scheme is called FSR and an efficient version based on the chain rule is called CFSR3.
2. U-MUSCL-SSQ: Conservative finite-difference scheme with point-valued solutions with no flux reconstruction but with a special source discretization at $\kappa = 1/2$. This scheme is called U-MUSCL-SSQ.

These two schemes have been demonstrated to give third-order accuracy on regular quadrilateral and triangular grids. For irregular triangular grids, the U-MUSCL-SSQ scheme still gives the lowest level of errors.

The only thing that we have been unable to explain is why Burg obtained the lowest truncation error with $\kappa = 1/2$ for the continuity equation on equilateral triangular girds [4]. This result remains a mystery; it is not clear

what kind of source quadrature he used for the source terms in the method of manufactured solutions (his other papers indicate that he integrated source terms). A possibility is that his quadrature was something similar to the compatible formula in Equation (5.11), so that his U-MUSCL scheme behaved like the third-order U-MUSCL-SSQ scheme but not just exactly.

In a subsequent paper, we will discuss extensions of the CFSR and U-MUSCL-SSQ schemes to fourth and fifth-order accuracy.

Acknowledgments

The authors gratefully acknowledge support from the U.S. Army Research Office under the contract/grant number W911NF-19-1-0429 with Dr. Matthew Munson as the program manager. The second author acknowledges support also from Software CRADLE, part of Hexagon and the Hypersonic Technology Project, through the Hypersonic Airbreathing Propulsion Branch of the NASA Langley Research Center, under Contract No. 80LARC17C0004. The authors are grateful to Dr. Yi Liu (National Institute of Aerospace) for his assistance in the setup of three-dimensional test cases.

References

- ¹Katz, A. and Sankaran, V., "An Efficient Correction Method to Obtain a Formally Third-Order Accurate Flow Solver for Node-Centered Unstructured Grids," *J. Sci. Comput.*, Vol. 51, 2012, pp. 375–393.
- ²Liu, Y. and Nishikawa, H., "Third-Order Inviscid and Second-Order Hyperbolic Navier-Stokes Solvers for Three-Dimensional Unsteady Inviscid and Viscous Flows," *55th AIAA Aerospace Sciences Meeting*, AIAA Paper 2017-0738, Grapevine, Texas, 2017.
- ³Nishikawa, H. and Liu, Y., "Accuracy-Preserving Source Term Quadrature for Third-Order Edge-Based Discretization," *J. Comput. Phys.*, Vol. 344, 2017, pp. 595–622.
- ⁴Burg, C. O. E., "Higher Order Variable Extrapolation for Unstructured Finite Volume RANS Flow Solvers," AIAA Paper 2005-4999, 2005.
- ⁵"FUN3D online manual," <http://fun3d.larc.nasa.gov>.
- ⁶Murayama, M. and Yamamoto, K., "Comparison Study of Drag Prediction by Structured and Unstructured Mesh Method," *J. Aircraft*, Vol. 45, No. 3, 2008, pp. 799–822.
- ⁷Hashimoto, A., Murayama, M., Yamamoto, K., and Aoyama, T., "Turbulent Flow Solver Validation of FaSTAR and UPACS," *52nd AIAA Aerospace Sciences Meeting*, AIAA Paper 2014-0240, National Harbor, 2014.
- ⁸White, J. A., Nishikawa, H., and Baurle, R. A., "Weighted Least-squares Cell-Average Gradient Construction Methods for the VULCAN-CFD Second-Order Accurate Unstructured-Grid Cell-Centered Finite-Volume Solver," *AIAA Scitech 2019 Forum*, AIAA Paper 2019-0127, San Diego, CA, 2019.
- ⁹Higo, Y., Nakashima, Y., Fujiyama, K., Irie, T., and Nishikawa, H., "RANS Solutions on Three-Dimensional Benchmark Configurations with scFLOW, a Polyhedral Finite-Volume Solver," *AIAA Aviation 2020 Forum*, AIAA Paper 2020-3029, 2020.
- ¹⁰"CFL3D online manual," <http://cfl3d.larc.nasa.gov>.
- ¹¹Yang, H. Q. and Harris, R. E., "Development of Vertex-Centered High-Order Schemes and Implementation in FUN3D," *AIAA J.*, Vol. 54, 2016, pp. 3742–3760.
- ¹²Nishikawa, H., "On Fake Accuracy Verification," arXiv:2008.12164 [math.NA], 27 August, 2020.
- ¹³van Leer, B., "Towards the Ultimate Conservative Difference Scheme. III. Upstream-centered Finite Difference Schemes for Ideal Compressible Flow," *J. Comput. Phys.*, Vol. 23, 1977, pp. 263–275.
- ¹⁴van Leer, B., "Upwind-Difference Methods for Aerodynamic Problems Governed by the Euler Equations," *Fluid Mechanics, Lectures in Applied Mathematics*, Vol. 22, 1985, pp. 327–335.
- ¹⁵Nishikawa, H., "A Truncation Error Analysis of Third-Order MUSCL Scheme for Nonlinear Conservation Laws," *Int. J. Numer. Meth. Fluids*, 2020, in press.
- ¹⁶Nishikawa, H., "The QUICK Scheme is a Third-Order Finite-Volume Scheme with Point-Valued Numerical Solutions," arXiv:2006.15143 [math.NA], 28 June, 2020.
- ¹⁷Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *J. Comput. Phys.*, Vol. 43, 1981, pp. 357–372.
- ¹⁸Rusanov, V. V., "Calculation of Interaction of Non-Steady Shock Waves with Obstacles," *J. Compt. Math. Phys. USSR*, Vol. 1, 1961, pp. 267–279.
- ¹⁹Nishikawa, H., "On the Loss and Recovery of Second-Order Accuracy with U-MUSCL," *J. Comput. Phys.*, Vol. 417, 2020, pp. 109600.
- ²⁰Debiez, C. and Dervieux, A., "Mixed-element-volume MUSCL methods with weak viscosity for steady and unsteady flow calculations," *Comput. Fluids*, Vol. 29, 2000, pp. 89–118.
- ²¹Yang, H. Q. and Harris, R. E., "High-Order Vertex-Centered U-MUSCL Schemes for Turbulent Flows," *Commun. Comput. Phys.*, Vol. 24, No. 2, 2018, pp. 356–382.
- ²²Leonard, B. P., "A Stable and Accurate Convective Modelling Procedure based on Quadratic Upstream Interpolation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 19, 1979, pp. 59–98.
- ²³Denaro, F. M., "Towards a New Model-Free Simulation of High-Reynolds-Flows: Local Average Direct Numerical," *Int. J. Numer. Meth. Fluids*, Vol. 23, 1996, pp. 125–142.
- ²⁴Zhang, R., Zhang, M., and Shu, C.-W., "On the Order of Accuracy and Numerical Performance of Two Classes of Finite Volume WENO Schemes," *Commun. Comput. Phys.*, Vol. 9, No. 3, 2011, pp. 807–827.

- ²⁵Koobus, B., Wornom, S., Camarri, S., Salvetti, M.-V., and Dervieux, A., "NonLinear V6 Schemes for Compressible Flow," INRIA-00224120v2, 2008.
- ²⁶Dement, D. C. and Ruffin, S. M., "Higher Order Cell Centered Finite Volume Schemes for Unstructured Cartesian Grids," *56th AIAA Aerospace Sciences Meeting*, AIAA Paper 2018-1305, Kissimmee, Florida, 2018.
- ²⁷Buchmüller, P. and Helzel, C., "Improved Accuracy of High-Order WENO Finite Volume Methods on Cartesian Grids," *J. Sci. Comput.*, Vol. 61, 2014, pp. 343–368.
- ²⁸Tamaki, Y. and Imamura, T., "Efficient Dimension-by-Dimension Higher Order Finite-Volume Methods for a Cartesian grid with Cell-Based Refinement," *Comput. Fluids*, Vol. 144, 2017, pp. 74–85.
- ²⁹Shu, C.-W. and Osher, S. J., "Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes, II," *J. Comput. Phys.*, Vol. 83, 1989, pp. 32–78.
- ³⁰Nishikawa, H., "Accuracy-Preserving Boundary Flux Quadrature for Finite-Volume Discretization on Unstructured Grids," *J. Comput. Phys.*, Vol. 281, 2015, pp. 518–555.
- ³¹Nishikawa, H., "Alternative Formulations for First-, Second-, and Third-Order Hyperbolic Navier-Stokes Schemes," *Proc. of 22nd AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2015-2451, Dallas, TX, 2015.
- ³²Liu, Y. and Nishikawa, H., "Third-Order Inviscid and Second-Order Hyperbolic Navier-Stokes Solvers for Three-Dimensional Inviscid and Viscous Flows," *46th AIAA Fluid Dynamics Conference*, AIAA Paper 2016-3969, Washington, D.C., 2016.
- ³³Caraeni, D. and Fuchs, L., "A New Compact High Order Multidimensional Upwind Discretization," *Proc. of 4th World CSCC conference*, Vouliagmeni, Greece, July 10–15 2000.
- ³⁴Caraeni, D. and Fuchs, L., "Compact Third-Order Multidimensional Upwind Scheme for Navier-Stokes Simulations," *Appl. Math.*, Vol. 15, 2002, pp. 373–401.
- ³⁵Nishikawa, H., Rad, M., and Roe, P., "A Third-Order Fluctuation-Splitting Scheme That Preserves Potential Flow," *Proc. of 15th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 01-2595, Anaheim, 2001.
- ³⁶Nishikawa, H. and Roe, P. L., "On High-Order Fluctuation-Splitting Schemes for Navier-Stokes Equations," *Computational Fluid Dynamics 2004*, edited by C. Groth and D. W. Zingg, Springer-Verlag, 2004, pp. 799–804.
- ³⁷Nishikawa, H., "Higher-Order Discretization of Diffusion Terms in Residual-Distribution Methods," *VKI Lecture Series 2006-01: CFD - High Order Discretization Methods*, edited by H. Deconinck and M. Ricchiuto, von Karman Institute for Fluid Dynamics, Belgium, 2006.
- ³⁸Nishikawa, H. and Roe, P. L., "High-Order Fluctuation-Splitting Schemes for Advection-Diffusion Equations," *Computational Fluid Dynamics 2006*, edited by H. Deconinck and E. Dick, Springer-Verlag, 2006.
- ³⁹Nishikawa, H., "Multigrid Third-Order Least-Squares Solution of Cauchy-Riemann Equations on Unstructured Triangular Grids," *Int. J. Numer. Meth. Fluids*, Vol. 53, 2007, pp. 443–454.
- ⁴⁰Tamaki, Y., *Turbulent Flow Simulations around Aircraft using Hierarchical Cartesian Grids and the Immersed Boundary Method*, Ph.D. thesis, University of Tokyo, March 2018.
- ⁴¹Gottlieb, S., Shu, C.-W., and Tadmor, E., "Strong Stability-Preserving High-Order Time Discretization Methods," *SIAM Rev.*, Vol. 43, No. 1, 2001, pp. 89–112.
- ⁴²Nishikawa, H. and Liu, Y., "Third-Order Edge-Based Scheme for Unsteady Problems," *AIAA 2018 Fluid Dynamics Conference*, AIAA Paper 2018-4166, Atlanta, Georgia, 2018.