Journal of Computational Physics 446 (2021) 110640

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Towards the ultimate understanding of MUSCL: Pitfalls in achieving third-order accuracy

Bram van Leer[a], Hiroaki Nishikawa[b,*]

[a] Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA
[b] National Institute of Aerospace, Hampton, VA 23666, USA

ARTICLE INFO

Article history:
Available online 24 August 2021

Keywords:
MUSCL
QUICK
Finite-volume
Finite-difference
Third-order
Deconvolution

ABSTRACT

We present a proof by analysis and numerical results that Van Leer's MUSCL conservative scheme with the discretization parameter κ is third-order accurate for $\kappa = 1/3$. We include both the original finite-volume MUSCL family, updating cell-averaged values of the solution, and the related finite-difference version, updating point values. The presentation is needed because in the CFD literature claims have been made that not $\kappa = 1/3$ but $\kappa = 1/2$ yields third-order accuracy, or even that *no* value of κ can yield third-order accuracy. These false claims are the consequence of mixing up finite-difference concepts with finite-volume concepts. In a series of Pitfalls, we show how incorrect conclusions can be drawn when pointwise values of the discrete solution are interchanged with cell-averaged values. All flawed schemes presented in the Pitfalls, and some correct ones for comparison, are tested numerically and shown to behave as predicted by the analysis. We conclude with firm recommendations on how to achieve third-order accuracy at all output times, or just in a steady state.

1. Introduction

In the 1970s, Van Leer published a series of articles under the title "Towards the Ultimate Conservative Difference Scheme", in which a family of higher-order, yet monotonicity preserving, upwind-biased schemes for hyperbolic conservation laws was developed [1–3]. These schemes can be regarded as higher-order sequels to Godunov's method [4]. Although originally formulated by Godunov as a finite-difference method updating point values, it is customary to present it as a finite-volume method with initial values in each cell taken constant at the cell-averaged level, and interface fluxes computed by a Riemann solver to create upwind bias. Van Leer discovered that higher-order upwind-biased schemes will result from reconstructing a linear or quadratic initial-value distribution in each cell by matching it to the average values in the cell and its direct neighbors. The interface flux is again taken from a Riemann solver. A code based on linear distributions in the cells was developed by Woodward and Van Leer [3] under the name MUSCL, or Monotonic Upstream-centered Scheme for Conservation Laws.

All schemes in the series "Towards..." are full space-time discretizations. In the MUSCL scheme, accurate time derivatives of the interface fluxes are computed based on characteristic theory, in order to center the fluxes in time. In 1980, Steve Hancock (Physics International, San Leandro, CA) pointed out that characteristic theory is not needed and formulated a simple predictor-corrector method as reported in Ref. [5] that is still being used [6]. The preferred approach, though, is to

[*] Corresponding author.
E-mail addresses: bram@umich.edu (B. van Leer), hiro@nianet.org (H. Nishikawa).

https://doi.org/10.1016/j.jcp.2021.110640

regard the MUSCL-type initial-value representation as a semi-discretization and combine it with multi-stage time-marching [7]; this is suited for both time-accurate and steady-state calculations.

The discretization principle of reconstructing initial values from cell averages followed by an upwind flux calculation has become known as the MUSCL approach; schemes built upon it are called MUSCL-type schemes or just MUSCL schemes. MUSCL schemes are finite-volume schemes; their nonlinear finite-difference version, with the solution given by point values, was developed by Osher and Shu [8,9] in the context of WENO schemes. Instead of reconstructing the solution in a cell from solution averages in the vicinity, we may choose to store cell-based moments of the solution as independent variables; such methods are called Discontinuous Galerkin methods [2,10]. There also are mixed methods with lower moments stored independently and higher derivatives obtained by reconstruction, in order to limit the number of independent variables [11,12].

The schemes in the series "Towards..." form a one-parameter family known as the "κ-schemes", after the parameter κ. Its useful range is $(-1, 1)$, with $\kappa = -1$ generating the fully one-sided second-order scheme [1,13], and $\kappa = 1$ yielding (unstable) central differencing; the MUSCL code was based on the choice $\kappa = 0$, which for a linear flux generates Fromm's advection scheme [14]. All κ-schemes are second-order accurate except for $\kappa = 1/3$, which yields third-order accuracy on a uniform grid. This has made $\kappa = 1/3$ a popular choice; for example, NASA's CFL3D code, developed by Anderson, Thomas, Walters and Van Leer [15,16] is based on it. Higher-order accuracy can be achieved on a wider upwind-biased stencil; see, for instance, Ullrich [17], who achieves fourth-order accuracy on the "cubed sphere".

In spite of its popularity, the choice $\kappa = 1/3$ has led to confusion and controversy. At stake is the alleged third-order accuracy of the scheme: from its publication onward there have been "Third-order deniers", i.e., researchers who maintain the scheme is only second-order accurate [18–21]. The authors of Ref. [19] even go so far as to announce it in their title: "Non-Existence of Third Order MUSCL Schemes". A milder judgement is found in Ref. [22], where the author argues that the MUSCL scheme can be third-order accurate with $\kappa = 1/3$, if only for steady conservation laws. Other authors do support the notion of third-order accuracy for $\kappa = 1/3$, but they derive a truncation error only for a linear advection equation and/or numerically demonstrate third-order accuracy only for linear and weakly nonlinear problems [23–26]. On the other hand, some argue that third-order accuracy is actually obtained with $\kappa = 1/2$ [27–29]. In today's language, this may be called "fake news"; it is not without danger. Imagine a student of CFD programming the $\kappa = 1/3$ scheme and getting only second-order-accurate results. Having read such an article will not motivate the student to search for the bug in the code.

The root cause of all this confusion is that the authors do not consistently follow the rules for designing or analyzing either a finite-difference method or a finite-volume method. At some stage in their design or analysis, they inadvertently regard a point value as a cell-averaged value, or vice versa. Since a solution value at a point and the average value in a cell centered on that point differ by $O(h^2)$, where h is the mesh width, the result of such a mix-up is often a second-order error in the solution. Exceptions to this rule are found only among schemes meant for marching to a steady solution.

The above examples of confusion regarding the third-order MUSCL scheme are taken from research papers; unfortunately, the confusion has also crept into textbooks on CFD. Authors of textbooks in general appear to be aware of the distinction between point values and cell-averaged values (see, e.g., Ref. [30], page 66 and Ref. [31], page 192) and, so long as they stick to describing the second-order finite-volume MUSCL scheme from Ref. [3], they do not err. However, in three well-known books [32–34] that do describe and analyze the third-order finite-volume MUSCL scheme, the truncation-error derived ends up being one for a pointwise update scheme. This can only lead to the incorrect conclusion that the scheme is third-order accurate only for linear or weakly nonlinear problems.

This paper attempts to put an end, once and for all, to the false notion that the κ-scheme with $\kappa = 1/3$ is only second-order accurate. In order to achieve this, we will proceed in the most thorough manner. In Section 2, we introduce the κ-family, presenting both its finite-difference and finite-volume forms, and show the relation between the two. In Section 3, we list various potential pitfalls resulting from confusing point values with cell-averaged values at different stages of the design or analysis process, and the authors that fell into some of these. Section 4 contains numerical experiments with κ-schemes illustrating both correct and incorrect uses of the finite-volume and finite-difference approach. The paper concludes with firm recommendations on how to achieve third-order accuracy at all output times, or just in the steady state.

Note that the distrust in MUSCL's third-order accuracy already exists with the scheme implemented on a uniform grid; uniform spacing is therefore assumed throughout our analysis. To preserve this third-order accuracy on a nonuniform grid is an entirely different matter, which will be discussed elsewhere (see, e.g., Refs. [35–37] for successful demonstrations of high-order MUSCL schemes on unstructured grids). Furthermore, throughout the paper, we only analyze and test "clean" MUSCL schemes free of limiters, as these devices obscure the accuracy of the underlying scheme.

The adverse effect on accuracy of confusing cell averages with point values and vice versa is not unique to the MUSCL scheme with kappa = 1/3. It is a potential danger to any conservative scheme of higher than second-order accuracy. By going into considerable detail for the third-order MUSCL scheme in the series of Pitfall-sections, we trust that the reader will understand how to avoid similar pitfalls when dealing with a different higher-order scheme.

2. The κ-family of upwind biased schemes

2.1. Notation and definitions

In this section, we restrict ourselves to the discussion of a single one-dimensional nonlinear or linear conservation law of the form,

$$\frac{\partial u}{\partial t}+\frac{\partial f(u)}{\partial x}=0, \tag{1}$$

with the associated characteristic speed $df(u)/du=a(u)$ assumed to be positive everywhere. We may do this because the issue at hand exists in one dimension, is not influenced by the number of conservation laws in the hyperbolic system, and neither to the upwind switch in the numerical flux formula. Discretizations of Eqn. (1) are made on a uniform grid with $x_j = jh$, representing either a grid point or the center of a finite-volume cell extending from $x_{j-1/2}=x_j-h/2$ to $x_{j+1/2}=x_j+h/2$. Point values of the solution and the flux will be denoted by u_j and f_j, cell averages by $\overline{u}_j$ and $\overline{f}_j$. In the present limited context, we shall call a scheme a finite-difference (FD) scheme when it updates point values; when it updates cell-averaged values, we shall call it a finite-volume (FV) scheme.[1] Certain mixed or hybrid forms are suited for marching to a steady solution; these appear in Sections 3.6-3.10.

2.2. Finite-difference form

The κ-scheme introduced in Ref. [1] is a *space-time* finite-difference discretization of a linear conservation law; we therefore need some extra notation. The updated point solution will be denoted by u^j, the time step is τ, and the CFL number $a\tau/h$, with constant a, is denoted by ν. For brevity, we introduce the backward-differencing operator $\Delta u_j=u_j-u_{j-1}$, and similarly for other grid functions. The κ-scheme derives from the backward method of characteristics:

$$u^j=u(x_j-a\tau), \tag{2}$$

where the right-hand side is evaluated at the initial time level. In general, the point $x_j-a\tau$ is not a grid point, so interpolation between grid-point values is needed. Using the upwind-biased stencil $(j-2, j-1, j, j+1)$, which allows cubic interpolation, the κ-scheme becomes:

$$u^j=u_j-\nu\Delta u_j-\frac{\nu(1-\nu)}{4}(\Delta u_{j+1}-\Delta u_{j-1})-\frac{\kappa\nu(1-\nu)(1-2\nu)}{4}(\Delta u_{j+1}-2\Delta u_j+\Delta u_{j-1}). \tag{3}$$

For $\kappa=1/3$, the scheme is third-order accurate. One may verify this by inserting the following values for ν: 2, 1, 0 and -1; the right-hand side of Eqn. (3) will then assume the values u_{j-2}, u_{j-1}, u_j and u_{j+1}, respectively, i.e., the cubic goes through all four grid-point values.

To reduce this scheme to a semi-discretization, we bring the first term on the right, u_j, to the left, divide both sides by τ, and take the limit for $\tau\to 0$. The result is:

$$\frac{du_j}{dt}=-\frac{a}{h}\Delta\left\{u_j+\frac{1}{4}(\Delta u_{j+1}+\Delta u_j)+\frac{\kappa}{4}(\Delta u_{j+1}-\Delta u_j)\right\}. \tag{4}$$

The expression between braces is an approximation of the solution at $x_{j+1/2}$. For $\kappa=1/3$, however, it is *not* the most accurate approximation possible, given the values of u_{j-1}, u_j and u_{j+1}. Rather, *it is the value that, when differenced across point j and multiplied by a/h, yields the best approximation of $\frac{\partial f}{\partial x}$ in the point j*. We shall come back to this in Section 3 under Pitfall 1. For future reference we write

$$f^{FD}_{j+1/2}=f_j+\frac{1}{4}(\Delta f_{j+1}+\Delta f_j)+\frac{\kappa}{4}(\Delta f_{j+1}-\Delta f_j), \quad f=au, \tag{5}$$

where the superscript FD indicates it is part of a finite-difference scheme.

Next, we allow the flux to be a nonlinear function of u, but we stay with the pointwise equation (1). To reduce the choice of schemes, we apply the constraint that the approximation of $\frac{\partial f}{\partial x}$ must vanish if the flux is constant on the grid; this automatically produces the conservation form. Without loss of generality, we may write:

$$\frac{du_j}{dt}=-\frac{1}{h}\left\{\alpha\Delta f_{j-1}+(1-\alpha-\beta)\Delta f_j+\beta\Delta f_{j+1}\right\}, \tag{6}$$

where α and β are constants. This may be rewritten as

[1] In the wider context of multidimensional computations on unstructured grids, an alternative definition is often used. A scheme is called FV if the spatial residual is integrated over a finite volume; it is called FD if the residual is evaluated pointwise [38]. This uncouples the definitions from the solution representation.

$$\frac{du_j}{dt} = -\frac{1}{h}\Delta\left\{f_j + \frac{\beta-\alpha}{2}(\Delta f_{j+1} + \Delta f_j) + \frac{\alpha+\beta}{2}(\Delta f_{j+1} - \Delta f_j)\right\}. \tag{7}$$

But we know what the constants in Eqn. (7) are from the linear case; they appear in Eqns. (4) and (5). Again, the expression between braces is an approximation for $\kappa = 1/3$ (not the most accurate one) of the flux value at $x_{j+1/2}$. So, for any flux, linear or nonlinear, we have

$$f^{FD}_{j+1/2} = f_j + \frac{1}{4}(\Delta f_{j+1} + \Delta f_j) + \frac{\kappa}{4}(\Delta f_{j+1} - \Delta f_j). \tag{8}$$

2.3. Finite-volume form

We are now ready to move to the finite-volume approach. For this purpose, we assume, before spatial discretization, a full initial-value distribution $u(x)$ is given. We broaden the definition of the backward-differencing operator:

$$\Delta u(x) = u(x) - u(x-h), \tag{9}$$

and rewrite Eqn. (4) as:

$$\frac{\partial u(x)}{\partial t} = -\frac{1}{h}\Delta\left\{f(x) + \frac{1}{4}[\Delta f(x+h) + \Delta f(x)] + \frac{\kappa}{4}[\Delta f(x+h) - \Delta f(x)]\right\}. \tag{10}$$

To arrive at a finite-volume scheme, simply integrate this equation from $x_{j-1/2}$ to $x_{j+1/2}$ and divide by h; this operation changes all point values into cell averages:

$$\frac{\partial \overline{u}_j}{\partial t} = -\frac{1}{h}\Delta\left\{\overline{f}_j + \frac{1}{4}[\Delta\overline{f}_{j+1} + \Delta\overline{f}_j] + \frac{\kappa}{4}[\Delta\overline{f}_{j+1} - \Delta\overline{f}_j]\right\}. \tag{11}$$

This is our first attempt to designing an FV scheme. Three things stand out:

1. The differencing operator in the FD scheme (4) is identical to the one in the finite-volume scheme (11). In the former, the operator is applied to point values of f, in the latter it is applied to cell averages of f.
2. The expression between braces is an approximation of the flux at $x_{j+1/2}$ *with the maximum accuracy for* $\kappa = 1/3$; we shall prove this further below. For future reference we write:

$$f^{FV}_{j+1/2} = \overline{f}_j + \frac{1}{4}[\Delta\overline{f}_{j+1} + \Delta\overline{f}_j] + \frac{\kappa}{4}[\Delta\overline{f}_{j+1} - \Delta\overline{f}_j]. \tag{12}$$

3. This is not a practical scheme since it uses cell-averaged flux values $\overline{f}$ that are not readily available in a finite-volume scheme. Rather, it is the solution average $\overline{u}$ that is available in each cell. To obtain $\overline{f}_j$, one first has to reconstruct the initial-value distribution in cell j from the cell averages in cells $j-1$, j and $j+1$, then compute $\overline{f}_j$ from it analytically or by numerical integration. The second step, however, is redundant, as we may immediately obtain $f_{j+1/2}$ from $u_{j+1/2}$ – and that is the MUSCL approach. Moreover, obtaining $\overline{f}_{j\pm1}$ spreads the stencil beyond that of the κ-scheme.

We shall now show how to reconstruct $u(x)$ in cell j and vicinity so that it matches the cell averages in cells $j-1$, j and $j+1$. To this purpose, it is most convenient [2,13] to expand the solution in terms of Legendre polynomials scaled to the interval $(x_{j-1/2}, x_{j+1/2})$. We therefore introduce the local nondimensional coordinate ξ:

$$\xi = \frac{2}{h}(x - x_j), \quad x_{j-3/2} \le x \le x_{j+3/2}, \tag{13}$$

and the first three polynomials $P_0(\xi) = 1$, $P_1(\xi) = \xi$, $P_2(\xi) = \xi^2 - 1/3$; these are orthogonal on the interval $(-1, 1)$. We write, using fresh coefficients:

$$u(\xi) = \alpha + \beta\xi + \gamma\left(\xi^2 - \frac{1}{3}\right), \quad -3 \le \xi \le 3, \tag{14}$$

and find the coefficients through the following constraints:

$$\overline{u}_j = \frac{1}{2}\int_{-1}^{1} u(\xi)d\xi = \alpha, \tag{15}$$

$$\overline{u}_{j+1} = \frac{1}{2}\int_{1}^{3} u(\xi)d\xi = \alpha + 2\beta + 4\gamma, \tag{16}$$

$$\overline{u}_{j-1} = \frac{1}{2}\int_{-3}^{-1} u(\xi)d\xi = \alpha - 2\beta + 4\gamma. \tag{17}$$

From Equations (16) and (17), we find:

$$\beta = \frac{1}{4}(\overline{u}_{j+1} - \overline{u}_{j-1}), \tag{18}$$

$$\gamma = \frac{1}{8}(\overline{u}_{j+1} - 2\overline{u}_j + \overline{u}_{j-1}). \tag{19}$$

So, the reconstructed initial-value distribution in the cell j becomes:

$$u(\xi) = \overline{u}_j + \frac{1}{4}(\overline{u}_{j+1} - \overline{u}_{j-1})\xi + \frac{1}{8}(\overline{u}_{j+1} - 2\overline{u}_j + \overline{u}_{j-1})\left(\xi^2 - \frac{1}{3}\right), \quad -1 \le \xi \le 1. \tag{20}$$

The value of $u_{j+1/2}$ is found by inserting $\xi = 1$:

$$u_{j+1/2} = \overline{u}_j + \frac{1}{4}(\Delta\overline{u}_{j+1} + \Delta\overline{u}_j) + \frac{1}{12}(\Delta\overline{u}_{j+1} - \Delta\overline{u}_j). \tag{21}$$

If we replace the coefficient 1/12 of the second-order term by $\kappa/4$, we retrieve the value of $u_{j+1/2}$ used in the κ-scheme (4). The MUSCL approach thus is defined by the following equations:

$$\begin{aligned} u^{FV}_{j+1/2} &= \overline{u}_j + \frac{1}{4}\left(\Delta\overline{u}_{j+1} + \Delta\overline{u}_j\right) + \frac{\kappa}{4}\left(\Delta\overline{u}_{j+1} - \Delta\overline{u}_j\right) \\ &= \overline{u}_j + \frac{1-\kappa}{4}\Delta\overline{u}_j + \frac{1+\kappa}{4}\Delta\overline{u}_{j+1} \end{aligned} \tag{22}$$

$$f^{FV}_{j+1/2} = f(u^{FV}_{j+1/2}). \tag{23}$$

We note that Eqn. (12) is identical to Eqn. (22) with u replaced by f, showing the equivalence of flux reconstruction from $\overline{f}$-values and solution reconstruction from $\overline{u}$-values. Looking back farther, to Eqn. (8), we see that the same reconstruction operator is used in the FD flux formula, applied to point values of the flux. We shall come back to this observation in Section 3.13. The equality of the coefficients in flux and solution reconstruction is not special to MUSCL schemes; the above derivation can be easily extended to conservative schemes of any order.

2.4. Proof of third-order accuracy for $\kappa = 1/3$

We may now turn to the matter of the order of accuracy of the MUSCL FV scheme with $\kappa = 1/3$. We know it is at least second-order accurate because the interface value (22) is exact if the initial values in cells $j-2$ through $j+1$ are given by one and the same quadratic polynomial. But is it third-order accurate?

To investigate this, let us assume the initial-value distribution in cells $j-2$ through $j+1$ is a cubic polynomial. When reconstructing the initial values in cell j after discretization, we must add a term proportional to $\partial^3 u/\partial x^3 h^3$, with spatial variation given by the next Legendre polynomial, $P_3(\xi) = \xi^3 - 3/5\xi$, in order not to affect the value of $\overline{u}_j$, which is exact. Hence, we write:

$$u(\xi) = \overline{u}_j + \beta'\xi + \gamma'\left(\xi^2 - \frac{1}{3}\right) + \epsilon u_{xxx}h^3\left(\xi^3 - \frac{3}{5}\xi\right), \quad -3 \le \xi \le 3, \tag{24}$$

where β' and γ' are modified coefficients. The third derivative of u will not be discretized, as that would expand the stencil of the scheme. Both u_{xxx} and the numerical constant ϵ, which need not be specified, are constant on the stencil. The two remaining constraints are:

$$\overline{u}_{j+1} = \frac{1}{2}\int_{1}^{3} u(\xi)d\xi = \overline{u}_j + 2\beta' + 4\gamma' + \frac{44}{5}\epsilon u_{xxx}h^3, \tag{25}$$

$$\overline{u}_{j-1} = \frac{1}{2}\int_{-3}^{-1} u(\xi)d\xi = \overline{u}_j - 2\beta' + 4\gamma' - \frac{44}{5}\epsilon u_{xxx}h^3. \tag{26}$$

The new coefficients are:

$$\beta' = \frac{1}{4}(\overline{u}_{j+1} - \overline{u}_{j-1}) - \frac{22}{5}\epsilon u_{xxx}h^3 = \beta - \frac{22}{5}\epsilon u_{xxx}h^3, \tag{27}$$

$$\gamma' = \frac{1}{8}(\overline{u}_{j+1} - 2\overline{u}_j + \overline{u}_{j-1}) = \gamma. \tag{28}$$

It is seen that the coefficient β has become contaminated by the third-order term; γ has not changed. Inserting these values into Eqn. (24) yields:

$$u(\xi) = \overline{u}_j + \beta\xi + \gamma\left(\xi^2 - \frac{1}{3}\right) + \epsilon u_{xxx}h^3\left\{\left(\xi^3 - \frac{3}{5}\xi\right) - \frac{22}{5}\xi\right\}, \quad -1 \le \xi \le 1, \tag{29}$$

and the interface value at $j+1/2$, which is exact, becomes:

$$u^{ex}_{j+1/2} = u^{FV}_{j+1/2} - 4\epsilon u_{xxx}h^3, \tag{30}$$

where $u^{FV}_{j+1/2}$ is taken from Eqn. (22), with $\kappa = 1/3$. Similarly, we find:

$$u^{ex}_{j-1/2} = u^{FV}_{j-1/2} - 4\epsilon u_{xxx}h^3. \tag{31}$$

Consider first the case of a linear flux, $f = au$. Both MUSCL-fluxes in Equations (30) and (31) would have a third-order error, but in the flux difference across cell j these errors would cancel, making the flux difference exact. The scheme would therefore be third-order accurate. If the flux is nonlinear, we expand it around its exact value:

$$f(u^{FV}_{j+1/2}) = f(u^{ex}_{j+1/2}) + 4\epsilon u_{xxx}h^3\left(\frac{\partial f}{\partial u}\right)_{j+1/2} + O(h^6), \tag{32}$$

and the flux difference across cell j becomes:

$$\begin{aligned} f^{FV}_{j+1/2} - f^{FV}_{j-1/2} &= f^{ex}_{j+1/2} - f^{ex}_{j-1/2} + 4\epsilon u_{xxx}h^3\left[\left(\frac{\partial f}{\partial u}\right)_{j+1/2} - \left(\frac{\partial f}{\partial u}\right)_{j-1/2}\right] + O(h^6) \\ &= f^{ex}_{j+1/2} - f^{ex}_{j-1/2} + 4\epsilon u_{xxx}h^3\left(\frac{\partial^2 f}{\partial u^2}\right)_j\left(u^{ex}_{j+1/2} - u^{ex}_{j-1/2}\right) + O(h^6) \\ &= f^{ex}_{j+1/2} - f^{ex}_{j-1/2} + 4\epsilon u_{xxx}u_x h^4\left(\frac{\partial^2 f}{\partial u^2}\right)_j + O(h^6). \end{aligned} \tag{33}$$

The purely nonlinear error in the scheme's flux difference across cell j appears to be proportional to h^4, an order higher than the degree of the initial-value polynomial, hence the MUSCL scheme with $\kappa = 1/3$ is third-order accurate.

3. Pitfalls

3.1. Overview

In the previous section, we analyzed both the finite-difference and the finite-volume version of the κ-family of schemes while being careful to keep the two approaches separated. Both versions are conservative, meaning that unique fluxes are defined halfway between the grid points or, equivalently, at cell interfaces. When designing such a numerical flux, there is a subtle difference between the FD and FV approaches in the way accuracy is achieved. In the FD approach, we look for the flux that, when differenced across a grid point, produces the most accurate approximation of $\partial f/\partial x$ in the point. In the FV approach, one simply looks for the most accurate flux. When these approaches are inadvertently mixed, loss of accuracy results. When programming a κ-scheme (or one with a wider stencil, aiming at higher than third-order accuracy), it is most prudent to adhere to the following rules:

1. For an FD scheme, use only pointwise data (initial values, solution values, source values) and construct interface fluxes only by direct flux reconstruction;
2. For an FV scheme, use only cell-averaged data and construct interface fluxes only by solution reconstruction.

With kappa = 1/3, one will be sure to achieve third-order accuracy. While these rules are safe, they are overly restrictive, limiting the programmer to schemes (8) and (23). There are examples of schemes using a mix of direct flux reconstruction and solution reconstruction that nevertheless are third-order accurate in solving both steady and unsteady problems (skip to Table 3). And there are schemes that mix point values and cell averages and still achieve third-order accuracy in the steady state (skip to Table 4). Below we list a range of pitfalls that may occur when one deviates from the above ground rules. Specifically, mixing reconstruction approaches may bring on Pitfalls 1 through 4.

When a source term $s(u, x)$ is added to the conservation law (1), more opportunities for inconsistency arise. In an FD scheme, which updates point values, the source term should generally be evaluated at a grid point; in an FV scheme, which updates cell averages, the source term should be cell-averaged. Mixing up these choices will cause loss of accuracy. This is illustrated in Pitfall 6. If, however, the scheme is used solely to march to a steady state, a mixed or hybrid discretization in some cases may improve the accuracy of the steady solution. This surprising fact is illustrated in Pitfall 5 and also plays a role in Pitfalls 6, 7 and 8.

Furthermore, there are pre- and post-processing errors that may be made, for instance, starting an FV calculation with pointwise instead of cell-averaged values, or comparing the cell-averaged values of an FV calculation with the point values of a reference solution. These are covered in Pitfalls 8 and 9, respectively.

Finally, Pitfalls 10 and 11 expose certain mistakes in the application of MUSCL. In Pitfall 10, the FV MUSCL scheme is applied to pointwise data, a common error; this changes it into an FD scheme that is third-order accurate only for linear and weakly nonlinear problems. In Pitfall 11, the FD version of MUSCL is erroneously mixed with an FV update, rendering the method third-order accurate only for steady - albeit nonlinear - problems.

3.2. Pitfall 1: Wrong choice of $\kappa\,(=1/2)$ for maximum-accuracy FD scheme

Researcher X plans to design a most accurate upwind-biased FD scheme. He has heard of flux reconstruction and begins with reconstructing a distribution of f centered on grid-point j:

$$f(x) = f_j + \frac{f_{j+1} - f_{j-1}}{2h}(x - x_j) + \frac{1}{2}\frac{f_{j+1} - 2f_j + f_{j-1}}{h^2}(x - x_j)^2; \tag{34}$$

this is the most accurate distribution one can obtain using the data in points $j-1$, j and $j+1$, so X expects the highest order of accuracy, i.e., third-order accuracy, to be delivered by the interface flux:

$$f_{j+1/2} = f_j + \frac{f_{j+1} - f_{j-1}}{4} + \frac{f_{j+1} - 2f_j + f_{j-1}}{8}. \tag{35}$$

This flux yields a perfectly valid FD scheme: it is the FD κ-scheme with $\kappa = 1/2$, of second-order accuracy. X has made the mistake of maximizing the accuracy of $f_{j+1/2}$ instead of maximizing the accuracy of $(\partial f/\partial x)_j$, which would require $\kappa = 1/3$. X finds that his numerical tests return only second-order accurate solutions.

Two incorrect conclusions are made by X. Before the test results have come in, he insists that $\kappa = 1/2$ not $\kappa = 1/3$, gives third-order accuracy. After the test results have come in, he concludes that no scheme of the κ-family can be third-order accurate. Both conclusions are found in the literature: the first one in Refs. [18,20,21], the second one in Ref. [19].

Nonetheless, third-order accuracy is hiding in the above scheme. Because the interface flux has the maximum accuracy, the flux difference $f_{j+1/2} - f_{j-1/2}$ can be used to update not u_j but $\overline{u}_j$ with third-order accuracy:

$$\frac{\partial \overline{u}_j}{\partial t} = -\frac{f_{j+1/2} - f_{j-1/2}}{h} + O(h^3). \tag{36}$$

This is of little practical use since the scheme is not meant to update $\overline{u}_j$ but the point-value u_j. We may try to eliminate $\overline{u}_j$ from this expression using the relation:

$$\overline{u}_j = u_j + \frac{u_{j+1} - 2u_j + u_{j-1}}{24} + O(h^4), \tag{37}$$

from which follows:

$$\frac{\partial \overline{u}_j}{\partial t} = \frac{\partial u_j}{\partial t} + \frac{1}{24}\left(\frac{\partial u_{j+1}}{\partial t} - 2\frac{\partial u_j}{\partial t} + \frac{\partial u_{j-1}}{\partial t}\right) + O(h^4). \tag{38}$$

This suggests that obtaining $\frac{\partial u_j}{\partial t}$ requires solving a tridiagonal system.[2] Within the margin of the target scheme's truncation error, though, we may eliminate the time derivatives in the second-order term using the first-order upwind or Godunov scheme:

$$\frac{\partial \overline{u}_j}{\partial t} = -\frac{\Delta f_j}{h} + O(h); \tag{39}$$

Eqn. (38) then becomes explicit:

$$\frac{\partial \overline{u}_j}{\partial t} = \frac{\partial u_j}{\partial t} - \frac{1}{24h}\left(\Delta f_{j+1} - 2\Delta f_j + \Delta f_{j-1}\right) + O(h^3). \tag{40}$$

[2] Recovering u_j from $\overline{u}_j$ actually is a deconvolution process; for a full treatment of the theory, with applications to FV schemes, see Refs. [39,40].

Combining (40) with (36) yields the following update scheme for u_j:

$$\frac{\partial u_j}{\partial t} = -\frac{1}{h}\left[\Delta f_j + \frac{\Delta f_{j+1} - \Delta f_{j-1}}{4} + \left(\frac{1}{8} - \frac{1}{24}\right)\Delta^3 f_{j+1}\right], \tag{41}$$

where

$$\Delta^3 f_{j+1} = \Delta f_{j+1} - 2\Delta f_j + \Delta f_{j-1}. \tag{42}$$

Thus, the correction term resulting from the difference between u_j and $\overline{u}_j$ in the end reduces the coefficient of $\Delta^3 f_{j+1}$ from 1/8, or $\kappa/4$ with $\kappa = 1/2$, to 1/12, or $\kappa/4$ with $\kappa = 1/3$, i.e., the third-order scheme is recovered.

Obviously, it is more practical to use $\kappa = 1/3$ in the flux formula from the start. Still, the notion that a correction term may exist restoring third-order accuracy in a scheme designed with $\kappa = 1/2$ has merit. Indeed, in Ref. [41], such a correction term is proposed; see further under Pitfall 5.

3.3. *Pitfall 2: Mixing FV with FD approach; $\kappa = 1/2$*

Researcher Y has the same idea as X but does not know about direct flux reconstruction. She reconstructs u, finding:

$$u_{j+1/2} = u_j + \frac{u_{j+1} - u_{j-1}}{4} + \frac{u_{j+1} - 2u_j + u_{j-1}}{8}, \tag{43}$$

$$f_{j+1/2} = f(u_{j+1/2}), \tag{44}$$

in the fashion of MUSCL. The resulting scheme in fact is a MUSCL scheme with $\kappa = 1/2$, applied to point values instead of cell averages, and it does produce a second-order solution of u_j (see further below). This suggests the following thought experiment: Assume the point values u_j happen to be the cell averages of a different initial-value distribution $v(x)$, that is,

$$u_j = \overline{v}_j. \tag{45}$$

Then scheme (43), (44) may be regarded as a regular MUSCL scheme with $\kappa = 1/2$ solving a slightly modified initial-value problem, namely, one with initial point values v_j related to u_j by:

$$u_j = \overline{v}_j = v_j + \frac{v_{j-1} - 2v_j + v_{j-1}}{24} + O(h^4). \tag{46}$$

This relation can be made explicit without changing its order of approximation:

$$v_j = \overline{v}_j - \frac{\overline{v}_{j+1} - 2\overline{v}_j + \overline{v}_{j-1}}{24} = u_j - \frac{u_{j+1} - 2u_j + u_{j-1}}{24} + O(h^4). \tag{47}$$

Eqn. (47) may be used at any time-level to approximately obtain the point values underlying the cell-averaged solution. Scheme defined by (43) and (44) thus solves two initial-value problems: one with initial values u_j, the other one with initial values v_j.

Is it possible to design a local correction term that restores third-order accuracy? To investigate this, we expand the flux $f(u_{j+1/2})$ with respect to the point j, making frequent use of the Roe-average [42] of the characteristic speed $a(u)$:

$$F(v) - f(u) = a(u,v)(v-u), \quad a(u,u) = a(u). \tag{48}$$

We may write:

$$\begin{aligned}
f(u_{j+1/2}) &= f(u_j) + a(u_j, u_{j+1/2})(u_{j+1/2} - u_j) \\
&= f_j + a(u_j, u_{j+1/2})\frac{\Delta u_j}{8} + a(u_j, u_{j+1/2})\frac{3\Delta u_{j+1}}{8}, \\
&= f_j + \frac{1}{8}a(u_{j-1}, u_j)\Delta u_j + \frac{3}{8}a(u_j, u_{j+1})\Delta u_{j+1} + \frac{1}{8}\left\{a(u_j, u_{j+1/2}) - a(u_{j-1}, u_j)\right\}\Delta u_j \\
&+ \frac{3}{8}\left\{a(u_j, u_{j+1/2}) - a(u_j, u_{j+1})\right\}\Delta u_{j+1}, \\
&= f_j + \frac{1}{8}\Delta f_j + \frac{3}{8}\Delta f_{j+1} + NLE.
\end{aligned} \tag{49}$$

The first three terms on the right hand side form the desired value $f^{FD}_{j+1/2}$, the remainder is the purely nonlinear error, abbreviated as NLE. To estimate the NLE, we first convert the Roe averages to local values. To begin with, we have:

$$a(u_j, u_{j+1/2}) = \frac{f_{j+1/2} - f_j}{u_{j+1/2} - u_j} = \frac{\frac{h}{2}\left(\frac{\partial f}{\partial x}\right)_{j+1/4} + O[(h/2)^3]}{\frac{h}{2}\left(\frac{\partial u}{\partial x}\right)_{j+1/4} + O[(h/2)^3]} = a_{j+1/4} + O[(h/2)^2]. \tag{50}$$

In the same way, we find

$$a(u_{j-1}, u_{j+1/2}) = a_{j-1/2} + O[(h/2)^2], \quad a(u_j, u_{j+1}) = a_j + O[(h/2)^2]. \tag{51}$$

Note that the error terms in Eqns. (50) and (51) scale differently and will not approximately cancel when subtracted from one another. We may now estimate the NLE:

$$\begin{aligned} NLE &= \frac{1}{8}\left[a(u_j, u_{j+1/2}) - a(u_{j-1}, u_j)\right]\Delta u_j + \frac{3}{8}\left[a(u_j, u_{j+1/2}) - a(u_j, u_{j+1})\right]\Delta u_{j+1} \\ &= \frac{1}{8}\left[\frac{3h}{4}(a_x)_j + O(h^2)\right]\left[h(u_x)_j + O(h^2)\right] - \frac{3}{8}\left[\frac{h}{4}(a_x)_j + O(h^2)\right]\left[h(u_x)_j + O(h^2)\right] \\ &= O(h^3). \end{aligned} \tag{52}$$

The purely nonlinear term is an order higher in h than the κ-term in $f^{FD}_{j+1/2}$, so the scheme is second-order accurate, as anticipated. Moreover, adding the correction term

$$-\frac{1}{24}\Delta^3 f_{j+1}, \tag{53}$$

to the flux difference across cell j will make it a third-order scheme. See further under Pitfall 5, where a source term is added.

3.4. Pitfall 3: Incorrect use of solution reconstruction in FD scheme; $\kappa = 1/3$

Researcher Z wants to design an FD scheme with third-order accuracy. He knows he must use $\kappa = 1/3$ but is not familiar with flux reconstruction. His scheme is a MUSCL scheme applied to point values:

$$u_{j+1/2} = u_j + \frac{1-\kappa}{4}\Delta u_j + \frac{1+\kappa}{4}\Delta u_{j+1}, \tag{54}$$

$$f_{j+1/2} = f(u_{j+1/2}), \tag{55}$$

where κ is understood to have the value 1/3. It is a third-order scheme if we regard the values u_j as cell averages of a modified initial-value distribution $v(x)$ with grid values v_j, as in the previous Pitfall. But is it third-order accurate as an FD scheme for updating u_j? The value of $u_{j+1/2}$ obtained for $\kappa = 1/3$ is not quadratically correct (compare to flux reconstruction in Eqn. (34)), and so there is no reason to expect of the scheme an exceptional accuracy.

Following the derivation presented in Pitfall 2, we arrive at the generalization of Eqn. (49):

$$\begin{aligned} f_{j+1/2} &= f^{FD}_{j+1/2} + \left[\frac{3(1-\kappa)}{16}(a_x)_{j-1/8}(u_x)_{j-1/2} - \frac{1+\kappa}{16}(a_x)_{j+3/8}(u_x)_{j+1/2}\right]h^2 + O(h^3) \\ &= f^{FD}_{j+1/2} + \frac{1}{4}\left(\frac{1}{2} - \kappa\right)(a_x)_{j+1/8}(u_x)_j h^2 + O(h^3). \end{aligned} \tag{56}$$

We see that the κ-term in $f^{FD}_{j+1/2}$ is swamped by the nonlinear truncation error for all values of κ except $\kappa = 1/2$; those schemes can therefore not be called κ-schemes. The case $\kappa = 1/2$ was analyzed under Pitfall 2 and was found to be second-order accurate, with the possibility of adding a correction term that produces third-order accuracy.

We conclude that using the FV version of the κ-scheme to update a solution given by point values never achieves higher accuracy than second order. This explains why in the CFD literature, e.g., in Ref. [19], one finds the statement that MUSCL cannot be third-order accurate. It is based on improper use of the FV operator.

Note that Z's scheme becomes the third-order FD scheme for linear equations with $\kappa = 1/3$ because a is a constant and solution reconstruction is then equivalent to flux reconstruction. This is another source of confusion: in some references, MUSCL-type schemes applied to point values are claimed to be third- or higher-order accurate based on accuracy verification studies performed for linear equations only [23–26]. It has been observed that high-order accuracy can still be achieved with those schemes for nonlinear equations if the targeted solution has a small enough variation to effectively linearize the equation [43]. See also Ref. [44] for a discussion of similar schemes of WENO type.

3.5. Pitfall 4: Incorrect use of flux reconstruction in FV scheme; $\kappa = 1/3$

Researcher Q wants to program an FV scheme for solving her favorite equations. She knows about MUSCL and the preference for $\kappa = 1/3$, but also has heard about direct flux reconstruction. Applying the latter method becomes her pitfall because she does not appreciate the difference between the average of the flux and the flux of the average. Instead of reconstructing $f_{j+1/2}$ from cell averages $\overline{f}_j$, admittedly not a practical method (see Sec. 2, Eqn. (12) and below), she reconstructs it from values of $f(\overline{u}_j)$:

$$f^{FV}_{j+1/2} = f(\overline{u}_j) + \frac{\Delta f(\overline{u}_{j+1}) + \Delta f(\overline{u}_j)}{4} + \frac{\kappa}{4}\Delta^2 f(\overline{u}_{j+1}), \tag{57}$$

which would be correct only in the case of a linear flux function. To demonstrate the difference between $\overline{f}_j$ and $f(\overline{u}_j)$, we interpolate both u and f quadratically from point values and find:

$$\overline{u}_j = uj + \frac{u_{j+1} - 2u_j + u_{j-1}}{24} + O(h^4), \tag{58}$$

$$\overline{f}_j = f_j + \frac{f_{j+1} - 2f_j + f_{j-1}}{24} + O(h^4). \tag{59}$$

To compute $f(\overline{u}_j)$, we use the expansion technique from Pitfall 2 and arrive at:

$$f(\overline{u}_j) = \overline{f}_j - \frac{1}{24}\left(\frac{\partial^2 f}{\partial u^2}\right)_j (u_x)^2_j h^2 + O(h^3). \tag{60}$$

The purely nonlinear term in $f(\overline{u}_j)$ gives $f_{j+1/2}$ an error $O(h^2)$, so the scheme is not a genuine κ-scheme and will only be second-order accurate.

The alternate view is to regard the cell averages $\overline{u}_j$ updated by the scheme as point values of a distribution $v(x)$; then the FV scheme changes into a legitimate FD scheme with flux reconstruction for $\kappa = 1/3$, i.e., a third-order scheme. The scheme then solves a slightly different initial-value problem, as we have:

$$v_j = \overline{u}_j, \quad u_j = v_j - \frac{v_{j+1} - 2v_j + v_{j-1}}{24} + O(h^4). \tag{61}$$

Is the use of $f(\overline{u}_j)$ as an estimate of $f(u_j)$ or $\overline{f}_j$ entirely without merit? Suppose Q had used solution reconstruction from $\overline{u}$ with $\kappa = 1/2$ and limited the use of $f(\overline{u}_j)$ to a flux-correction term:

$$-\frac{1}{24}\Delta^2 f(\overline{u}_{j+1}). \tag{62}$$

In this second-order term, it does not matter that $f(\overline{u}_j)$ is not equal to $\overline{f}_j$ or $f(u_j)$; the error made in the term is only $O(h^4)$. One might therefore expect the resulting FV scheme to be third-order accurate. In the alternate view, the scheme's flux function reduces to the flux function of Pitfall 2 with the correction term from Pitfall 1, yielding third-order accuracy; this is the basis of Leonard's QUICKEST scheme [45], see Pitfall 5.

An analysis of the flux function reveals, again, that it contains a second-order, purely nonlinear error term, so third-order accuracy is not achieved. Moreover, no matter how the flux function is divided up into a reconstruction step and a correction step, i.e., no matter which value of κ is used in the reconstruction, third-order accuracy is unattainable. To prove this, we may adapt the analysis given under Pitfall 3. In the solution-reconstruction step, we first write the lead term $\overline{u}_j$ as the point value u_j plus an approximate convolution term $\Delta^2 u_{j+1}/24$; the latter term is then included in the κ-term:

$$u_{j+1/2} = u_j + \frac{1}{4}\left(\Delta\overline{u}_{j+1} + \Delta\overline{u}_j\right) + \frac{\kappa'}{4}\Delta^2\overline{u}_{j+1} + O(h^4), \quad \kappa' = \kappa + 1/6. \tag{63}$$

Next, we realize that, within a margin $O(h^3)$, all differences of $\overline{u}_j$ in Eqn. (63) may be replaced by differences of point values u:

$$u_{j+1/2} = u_j + \frac{1}{4}\left(\Delta u_{j+1} + \Delta u_j\right) + \frac{\kappa'}{4}\Delta^2 u_{j+1} + O(h^4), \quad \kappa' = \kappa + 1/6. \tag{64}$$

Thus, we have reduced the analysis to that of Pitfall 3, with a modified κ-value. It follows that $\kappa' = 1/2$, hence $\kappa = 1/3$, is the only value that allows of flux correction toward third-order accuracy - but for that value the scheme already is third-order accurate. The exceptional value $\kappa = 1/2$ for FD schemes appears to correspond to the value $\kappa = 1/3$ for FV schemes.

Yet, Eqn. (63) suggests there is another way to utilize $f(\overline{u}_j)$ in an FV scheme. If we reduce the solution-reconstruction step to the deconvolution of the lead term $\overline{u}_j$ and then compute f_j:

$$f_j = f\left(\overline{u}_j - \frac{1}{24}\Delta^2\overline{u}_{j+1}\right) + O(h^4), \tag{65}$$

we may compute the missing part of the flux function using $f(\overline{u}_j)$:

$$f_{j+1/2} = f\left(\overline{u}_j - \frac{1}{24}\Delta^2\overline{u}_{j+1}\right) + \frac{\Delta f(\overline{u}_{j+1}) + \Delta f(\overline{u}_j)}{4} + \frac{\kappa'}{4}\Delta^2 f(\overline{u}_{j+1}) + O(h^3). \tag{66}$$

But f_j may also be written as follows:

$$f_j = \overline{f}_j - \frac{1}{24}\Delta^2 f(\overline{u}_{j+1}) + O(h^4), \tag{67}$$

so that Eqn. (66) can also be written as:

$$f_{j+1/2} = \overline{f}_j + \frac{\Delta f(\overline{u}_{j+1}) + \Delta f(\overline{u}_j)}{4} + \frac{\kappa}{4}\Delta^2 f(\overline{u}_{j+1}) + O(h^3), \tag{68}$$

$$\overline{f}_j = f\left(\overline{u}_j - \frac{1}{24}\Delta^2\overline{u}_{j+1}\right) + \frac{1}{24}\Delta^2 f(\overline{u}_{j+1}) + O(h^4). \tag{69}$$

This is as close as one can get to the FV flux function (12) at limited computational expense. For $\kappa = 1/3$, it has a third-order truncation error; the deconvolution of $\overline{u}_j$ and reconvolution of f_j in Eqn. (69) eliminates the purely nonlinear second-order errors that plague the fluxes (57) and (60). Note that these operations have not spread the stencil of the flux formula.

3.6. *Pitfall 5: Hybrid scheme*

Researcher R wants to develop an FV scheme, complete with source term, but does not want to give up the point-valued solution. He starts from the extended conservation law:

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = s(u, x), \tag{70}$$

and correctly integrates it over a finite-volume cell, yielding:

$$\frac{\partial \overline{u}_j}{\partial t} = -\frac{1}{h}\left(f_{j+1/2} - f_{j-1/2}\right) + \overline{s}_j \equiv Res_j, \tag{71}$$

where Res stands for “residual”. The residual is computed with $f_{j\pm 1/2}$ obtained by quadratic solution reconstruction from point values, as in Pitfall 2, Eqn. (44); the source term is averaged over the cell. Because the solution is given in grid points, R has no other choice than to apply the residual to u_j. The scheme thus becomes:

$$\frac{\partial u_j}{\partial t} = -\frac{1}{h}\left(f^{FV}_{j+1/2} - f^{FV}_{j-1/2}\right) + \overline{s}_j, \tag{72}$$

where $f^{FV}_{j+1/2}$ is given by Eqn. (23). This is an example of a hybrid scheme, combining a pointwise solution with a cell-averaged source term; its utility remains to be discovered. To the disappointment of R, this scheme gives only second-order accurate results. We readily understand this upon rewriting Eqn. (71) as:

$$\frac{\partial(u_j + O(h^2))}{\partial t} = Res_j, \tag{73}$$

which is valid regardless how accurately the residual is calculated. Subsequently, using scheme (72) to march to a steady state afforded by the source term, R is surprised to find that the steady solution is third-order accurate, after all. He has iteratively solved, within round-off error, the equation $Res_j = 0$ for all j, or

$$\frac{1}{h}\left(f^{FV}_{j+1/2} - f^{FV}_{j-1/2}\right) = \overline{s}_j. \tag{74}$$

The truncation error in each flux and in the source term is $O(h^3)$: so is the error in the divided flux difference. The balance between the flux difference and the source term is exact, so the $O(h^3)$ error must be in the solution itself.

R now aspires to make the time accurate scheme also third-order accurate. He realizes that he has to correct for the difference between u_j and $\overline{u}_j$ but does not want to tamper with the established code; instead he formulates a correction term, to be added to the RHS of scheme (72). Starting from

$$\frac{\partial \overline{u}_j}{\partial t} = \frac{\partial u_j}{\partial t} + \frac{1}{24}\left(\frac{\partial \overline{u}_{j+1}}{\partial t} - 2\frac{\partial \overline{u}_j}{\partial t} + \frac{\partial \overline{u}_{j-1}}{\partial t}\right) + O(h^4), \tag{75}$$

he arrives at the following corrected scheme:

$$\frac{\partial u_j}{\partial t} = Res_j - \frac{1}{24}\left(Res_{j+1} - 2Res_j + Res_{j-1}\right). \tag{76}$$

Not only does the correction term contain the flux correction (40) to make the switch from $\kappa = 1/2$ to $\kappa = 1/3$ (see Pitfall 1, Eqn. (41)), it also includes the term $-\Delta^2 \overline{s}_{j+1}/24$ that changes the source term from $\overline{s}_j$ to s_j. The scheme is now comparable to the FD κ-scheme with $\kappa = 1/3$, flux reconstruction and pointwise source term; it is third-order accurate.

The correction term in (76) is a convenient choice but widens the stencil with two points. In such a case, one always wonders if the extra information could not have been put to a better use, specifically, to make the upwind-biased scheme fourth- or fifth-order accurate. Within the original stencil, the correction term could be evaluated as $(\Delta^3 f_{j+1} - \Delta^2 s_{j+1})/24$.

The above scenario closely follows the work of Leonard[3] [41]. Scheme (72) compares to Leonard's QUICK scheme, and scheme (76) compares to his QUICKEST scheme.

Eqn. (72) is also the scheme with which Burg [28] achieved third-order accuracy with $\kappa = 1/2$ in solving a one-dimensional nonlinear steady-state problem. Clearly, Burg's U-MUSCL scheme cannot be third-order accurate in an unsteady simulation, regardless of the time-integration scheme used, and not even in one dimension.

Another way to convert the QUICK scheme into one that is third-order accurate at any time is to actually use the FV residual to update $\overline{u}_j$, as in Eqn. (71). The scheme is now an FV scheme, with the solution stored as cell averages, but we may still choose to compute the fluxes from point values. We obtain these auxiliary point values from a variation on Eqn. (47):

$$u_j = \overline{u}_j - \frac{1}{24}\Delta^2 \overline{u}_j + O(h^4). \tag{77}$$

To see the effect of this conversion, we insert Eqn. (77) into Eqn. (43) and find:

$$u_{j+1/2} = \overline{u}_j + \frac{1}{4}\left(\Delta \overline{u}_{j+1} + \Delta \overline{u}_j\right) + \frac{1}{12}\Delta^2 \overline{u}_{j+1} - \frac{1}{24}\Delta^2 \left\{ \frac{1}{4}\left(\Delta \overline{u}_{j+2} + \Delta \overline{u}_{j+1}\right) + \frac{1}{8}\Delta^2 \overline{u}_{j+2} \right\}. \tag{78}$$

We observe two things: (1) the residual stencil will spread into cells $j-3$ and $j+2$; (2) the new solution value at $x_{j+1/2}$ equals the value (22) appearing in the FV MUSCL scheme with $\kappa = 1/3$, augmented by third- and fourth-order terms that do not make the scheme more accurate. Clearly, using the original MUSCL scheme is preferable.

3.7. Pitfall 6: Another hybrid scheme; incorrect source-term discretization

Researcher Z from Pitfall 3 has learned about second-order accurate schemes producing third-order accurate steady solutions. He adds a source term to his FD code, evaluated pointwise, and has the code march to a steady state on several grids. The refinement study shows that the pointwise results are still second-order accurate.

Z has used the proper source term for an FD scheme but not for getting the benefit of third-order accuracy in the steady state. Had he used a cell-averaged source term, he would have found that the steady numerical solution, compared to the *cell-averaged* exact solution, indeed has a third-order error. In the steady state, it does not matter whether the cell values were originally designated point values or cell averages. Provided that the source term is cell-averaged, $\kappa = 1/3$ will bring third-order accuracy with respect to the cell-averaged exact solution, and $\kappa = 1/2$ will bring third-order accuracy with respect to the point values of the exact solution. The latter is the case of the U-MUSCL scheme of Burg [28] and also of the QUICK scheme of Leonard [41]; see Pitfall 5.

Disappointed, Z decides to try his luck with an FV scheme. To convert his FD scheme, with $\kappa = 1/3$ and solution reconstruction, into an FV MUSCL scheme, all he has to do is start from cell-averaged initial values and at output time compare the numerical values to the cell-averaged exact solution. To his relief, he finds his scheme has become third-order accurate. But when he switches on the source term and marches to a steady state, the scheme reverts to the second order.

Z has forgotten to change the point-valued source term held over from his FD code to a cell-averaged source term. As he has become more assertive since his recent success, he realizes his oversight and finally achieves third-order accuracy in the steady state using a cell-averaged source term. From now on, he tells his students to stay away from FD schemes and embrace the FV method.

Note that Z would have obtained third-order accurate solutions with $\kappa = 1/3$ and a pointwise source term, had he restricted himself to solving a linear equation. As mentioned under Pitfall 3, multiple references claim third-order accuracy based only on an analysis of linear or weakly nonlinear advection [28,23–26,43] (see Pitfall 10). To his credit, researcher Z did not step into that one.

[3] Leonard considers QUICK as an FV scheme; in the context of the present paper, we must call it an FV-FD hybrid. QUICKEST, again, is an FD scheme; the FV aspect is completely undone by the correction term.

3.8. Pitfall 7: Yet another hybrid scheme; post-processing error

Researcher Q from Pitfall 4 also has learned that some second-order accurate discretizations surprisingly produce third-order accurate steady solutions. She adds to her FV code a source term $s(x)$, correctly cell-averaged, and has the code march in time until the residual bottoms at round-off level. A grid-refinement study shows that the results are still second-order accurate. She sends the program and output files to her graduate student C, to have her look for a bug or irregularity.

The output file for each grid has a column with the point values of the exact solution, a column with the exact cell averages, computed from the point values by quadrature, and a column with the cell values of the numerical solution. C does not find anything remarkable in the data, nor can she find any bug in the code. She decides to go to the source.

From the original paper by researcher R, she learns he has combined an FD scheme with a *cell-averaged* source term. Perhaps, she thinks, "I should do the opposite: combine my FV code with a pointwise source term". And so she does, but the converged steady solutions still differ $O(h^2)$ with the cell-averaged exact solution. When looking at the new data, C observes that on each grid the numerical cell values are closer to the pointwise exact solution than to the cell-averaged exact solution. She recomputes the error norms with respect to the pointwise exact solution and finds that the errors now decay as h^3, lending a happy ending to this Pitfall.

The discrete operator in Q's FV scheme is identical to the one in the FD scheme with flux reconstruction for $\kappa = 1/3$, see Eqn. (8). The latter is meant to act on pointwise initial values. If given cell-averaged initial values but a pointwise source value, it still will reach the steady pointwise solution with third-order accuracy because the steady solution does not depend on the initial-value distribution. The next FV scheme in Pitfall 4, with flux correction (62), will behave similarly when provided with a pointwise source term. The last FV scheme from Pitfall 4, with flux given by Eqns. (68) and (69), is third-order accurate for all times when provided with a cell-averaged source term, but it produces only a second-order accurate steady solution when combined with a pointwise source term. Reason: the expression for $\overline{f}_j$ in Eqn. (69) reduces to $f_j + O(h^2)$ when all cell averages of u are replaced by point values.

3.9. Pitfall 8: Post-processing error

Researcher S is interested in accurately finding steady solutions generated by a source term $s(x)$. She programs the MUSCL FV method with solution reconstruction for arbitrary κ, and correctly cell-averages the source term. Her initial-value discretization, though, does not include cell averaging; she merely chooses point values of a function far removed from the steady solution, in order to challenge the scheme.

S first takes $\kappa = 1/3$, aiming at third-order accuracy, and has the scheme march in time on a sequence of grids. Having reached the known steady state within round-off error, she makes her mistake: she compares the numerical solution values to *point* values of the known steady solution instead of cell-averaged values. Thus she finds her solution to be only second-order accurate.

Next, she makes the calculation with $\kappa = 1/2$ and now finds her steady solution to be third-order accurate, when compared to the point values of the exact solution. (We know from Pitfall 5 why this is so.) When publishing these results, she makes the assertion that MUSCL is second-order accurate with $\kappa = 1/3$ and third-order accurate with $\kappa = 1/2$.

Had S post-processed correctly, she would have made the reverse conclusion. The cell values of her steady solution for $\kappa = 1/3$ would have shown third-order accuracy when compared to the cell-averaged exact solution. In contrast, the cell values of her steady solution for $\kappa = 1/2$ would have shown second-order accuracy when compared with the cell-averaged exact solution. The analysis presented in Ref. [28] is an example of such confusion.

3.10. Pitfall 9: Pre-processing error

When Researcher S from Pitfall 8 presents her results at a conference, someone in the audience warns her that her conclusions are premature and that she should try to solve a time-dependent problem. She heeds this advice and now finds that for *both* values of κ the numerical solutions are second-order accurate.

Unfortunately, S is not well-positioned to do time-accurate calculations with FV schemes. Not only will she make the post-processing error from Pitfall 8, she also makes a pre-processing error: her code still starts a calculation with point values instead of cell averages of the given continuous initial-value function. There is no chance she will uncover the true accuracy of MUSCL FV with $\kappa = 1/3$. For the tests with $\kappa = 1/2$, the pre- and post-processing errors do not matter; these are of the same order as the truncation error of the scheme. With pointwise initial values, the scheme will be identical to QUICK, discussed under Pitfall 5, which is second-order accurate in the absence of flux correction.

3.11. Pitfall 10: Good intentions, faulty analysis I

Researcher T is enthusiastic about MUSCL and wants to show, once and for all, that it is third-order accurate for $\kappa = 1/3$. He works thoroughly, starting out with a linear truncation-error analysis. Alas, he applies the MUSCL operator, including solution reconstruction, to point values of the solution, as in an FD scheme, instead of to cell averages, as in an FV scheme; this mistake does not immediately show up because for a linear flux, $f = au$, a constant, there is no difference between solution reconstruction and direct flux reconstruction. So, his analysis indicates third-order accuracy and he confirms this with numerical tests of linear advection.

Encouraged by these results, he changes the flux into Burgers' flux, $f = u^2/2$, adds a pointwise source term $s(x)$, and solves the steady problem on a sequence of grids. To his disappointment, the results are now only second-order accurate. T tries to salvage the experiment by choosing as the steady solution a constant u_0 plus a perturbation with small amplitude ϵ, thus making the problem only weakly nonlinear. Indeed, for sufficiently small ϵ, he does recover third-order convergence. His published conclusion is that MUSCL with $\kappa = 1/3$ is third-order accurate only for weakly nonlinear steady-state problems.

T has followed the tracks of Researcher Z, stepping into Pitfalls 3 and 6. His final conclusion can be readily understood: for small ϵ, his scheme has two competing major components in the truncation error: a quasilinear[4] error $\sim \epsilon h^3$ and a purely nonlinear error $\sim \epsilon^2 h^2$; see Eqn. (56). If $h > C\epsilon/u_0$, where C is a small constant, the quasilinear error dominates and cubic convergence is observed; if $h < C\epsilon/u_0$ the nonlinear error dominates and convergence is only quadratic. This Pitfall is analyzed in more detail and illustrated numerically in Section 4.4.

The incorrect application of FV κ-schemes to a point-valued solution, as in the above example, or, equivalently, the use of solution reconstruction in an FD scheme, is a major source of false statements about MUSCL's accuracy. Sometimes the authors omit indicating what choice of data they use, but their incorrect analysis results are tell-tale. This is the case in Ref. [19], which bears the title: "Non-Existence of Third Order MUSCL Schemes". Furthermore, in three textbooks on CFD [32–34] that include an analysis of the third-order MUSCL scheme, the authors choose to discuss and analyze the MUSCL and QUICK schemes in the same framework; this inevitably leads to tacit application of the third-order MUSCL operator to point values. There also are authors [26,46,47] who claim to have extended MUSCL to the fourth order of accuracy, although their claim is supported only by linear and weakly nonlinear numerical test results. It is doubtful that their schemes are fourth-order accurate for fully nonlinear problems, especially because there is evidence that they, too, have treated their solution data as point values. Nevertheless, some of their test results show increased resolution compared to those obtainable on the same grid with a second-order method. A truly fourth-order extension of FV MUSCL is described in Ref. [17], with applications to atmospheric flow over the "cubed" sphere (the outward projection of a cube's Cartesian surface grid onto the circumscribed sphere).

3.12. Pitfall 11: Good intentions, faulty analysis II

Researcher Y has learned a lot about CFD since her first attempt to construct an accurate scheme, described in Pitfall 2, and decides to give it another try. She starts out with the finite-volume form of the conservation law:

$$\frac{\partial \overline{u}}{\partial t} + \frac{\Delta f_{j+1/2}}{h} = \overline{s}_j, \tag{79}$$

and as the numerical flux function $f_{j+1/2}$, she takes $f^{FD}_{j+1/2}$ from Eqn. (8), direct flux reconstruction from *point values* via the κ-formula, with the intent to use $\kappa = 1/3$. Thus, the scheme becomes hybrid, and some work needs to be done to make this a fully pointwise update scheme. A truncation-error analysis produces the following approximate equation representing the scheme in a point:

$$u_t + f_x + \frac{h^2}{24} u_{txx} + \frac{h^2}{4}\left(\kappa - \frac{1}{3}\right) f_{xxx} = s + \frac{h^2}{24} s_{xx} + O(h^3). \tag{80}$$

It is seen that the contribution to the truncation error by the flux term indeed vanishes for $\kappa = 1/3$, but now the term $\sim u_{txx}$ is particularly worrisome. It may be approximated by $\Delta^2(u_t)_{j+1}/24$, but that makes the update scheme implicit. Moreover, it indicates that third order accuracy may only be attained in the steady state, when all time derivatives vanish. Note that in that case, third-order accuracy is achieved for $\kappa = 1/2$ rather than $\kappa = 1/3$. In the end, Y drops both the terms $\sim u_{txx}$ and $\sim s_{xx}$, when writing her computer code, i.e., she abandons the cell-averaging of u_t and the source term. To her surprise, her numerical experiments produce both steady and unsteady solutions with third-order accuracy. We know why: she has rediscovered the standard FD scheme with $\kappa = 1/3$.

The above derivation of the hybrid scheme (79), found in Ref. [22], is a faulty attempt to obtain an FV scheme starting from (8); the correct derivation, given in Section 2.3, leads to Eqn. (11). To correctly change (79) back into an FD scheme, we start from the representative equation (80), and simplify it to:

$$u_t + f_x = s + O(h^2). \tag{81}$$

Apply the operator $(h^2/24)\partial^2/\partial x^2$ to both sides of it and subtract the resulting equation from (79). The terms $\sim u_{txx}$ and $\sim s_{xx}$ have the effect of reducing $(\overline{u}_j)_t$ and $\overline{s}_j$ back to their point values $(u_j)_t$ and s_j, while the term $\sim f_{xxx}$ changes the coefficient $\kappa/4$ in the flux function to $\kappa'/4 = (\kappa - 1/6)/4$. This means Y could have simulated her hybrid scheme with

[4] Quasilinear: linear in f, although nonlinear in u.

a regular FD scheme incorporating $\kappa' = 1/3 - 1/6 = 1/6$, which indeed does not yield third-order accuracy. By choosing $\kappa' = 1/3$, or $\kappa = 1/2$, her hybrid scheme would have become third-order accurate in unsteady calculations – but the scheme would remain difficult to execute.

The value $\kappa = 1/2$ also appears in the hybrid schemes discussed in Pitfalls 1 and 5; the latter scheme is QUICK; it uses solution reconstruction for its flux but behaves similarly to one that uses flux reconstruction. The scheme is actually third-order accurate for unsteady calculations in its hybrid form, but again is difficult to execute. By applying the same deconvolution procedure as done above to scheme (79), we can reduce QUICK to an FD scheme; this decreases the effective value of κ to $1/2 - 1/6 = 1/3$; in other words, the hybrid QUICK scheme is equivalent to the FD scheme with $\kappa = 1/3$, confirming its third-order accuracy.

The appearance of the value $\kappa = 1/2$ in the analysis of schemes that are wanting in some way, is no accident. We therefore end this parade of Pitfalls with some observations on the significance of this value, and the associated role of deconvolution in both FV and FD schemes.

3.13. A final question: Why are the reconstruction operators for third-order accurate FD and FV the same?

It is somewhat surprising that the differencing operator for flux reconstruction in the third-order accurate FD κ-scheme is identical to the one used for solution reconstruction in third-order accurate MUSCL. Both reconstructions feature $\kappa = 1/3$. Although this is evident from the derivation of the FV scheme from the FD version, given in Section 2.3 (see Eqn. (11)), additional clarification may be welcome.

The basis lies in the fact that both FD and FV schemes use the same quadratic interpolant, for flux and solution reconstruction, respectively. This is the most accurate reconstruction possible using values from cells $j-1$, j, and $j+1$, and the only way to achieve third-order accuracy in the scheme. Because of this, and in order to realize the maximum accuracy, the residual formed from differencing the interface flux across cell j,

$$Res_j = \frac{f_{j+1/2} - f_{j-1/2}}{h}, \tag{82}$$

must be regarded as an FV residual in both schemes.

Quadratic interpolation of point values of the flux is described in Pitfall 1, Eqn.(35); quadratic interpolation of point values of the solution is described in Pitfall 2, Eqn. (43). In both cases, the reconstructed interface value, of flux and solution, respectively, is given by the standard formula with $\kappa = 1/2$. Yet, in the final form of the schemes, the value $\kappa = 1/3$ appears in both. The reason is that in both schemes at some point, a deconvolution is needed from cell average to point value.

In the FD scheme, which updates point values, it is the residual itself that must be converted, because it represents the time derivative of the cell-averaged solution, $\partial \overline{u}_j/\partial t$, not $\partial u_j/\partial t$. How to do this without increasing the scheme's stencil is described in Pitfall 1, Eqn. (41). The result is a flux-correction term that has the effect of lowering κ from 1/2 to 1/3.

In the FV scheme, the FV residual is appropriate because the solution is stored as a cell average. A consequence is that the quadratic solution reconstruction must use cell averages rather than point values; as shown in Eqn. (37), these differ by $O(h^2)$. No adjustment is needed in the first- and second-order terms of the quadratic polynomial as replacing point values with average values produces errors that fall within the truncation error of the reconstruction. The lead term u_j, though, must be deconvoluted according to Eqn. (47); this introduces the by now well-known second-order term that lowers κ from 1/2 to 1/3.

4. Numerical results

4.1. Schemes and notations

We proceed with testing all distinct schemes described earlier on their order of accuracy. All schemes are capable of producing second-order results in time-accurate and steady-state calculations; some achieve third-order accuracy.[5] Below are two tables with the schemes in their order of appearance in the Pitfalls. The schemes in Table 1 from Pitfalls 1-4 do not include a source term; they are tested on the time-dependent problem of Sec. 4.2. They are second-order accurate, except MUSCL(1/3) and its FD counterpart, but some can be corrected to yield third-order accuracy. The schemes in Table 2 from Pitfalls 5-9 include a source term and are tested on the steady-state problem of Sec. 4.3. They all yield third-order accuracy, bar pre- or postprocessing errors and except one scheme that is third-order accurate only for linear and weakly nonlinear problems. The following abbreviations are used henceforth.

[5] Note that, to ensure third-order accuracy will be observable, the initial values and final solutions of any numerical test problem must be at least three times continuously differentiable.

Table 1
List of schemes for unsteady problem. Incorrect pre- or/and postprocessing will lower OoA to 2 for any third-order scheme. Note: (149) in FV-FR(1/3,(149)) should be understood as (66), i.e., the use of the flux reconstruction (66).

Scheme	Reconstr.	FC possible?	OoA unsteady	Pitfall #	Comment
FD-FR(1/2)	(34)	yes (40)	2	1	
FD-FR(1/2)-FC	(34)	included (40)	3 in u_j	1	Identical to FD-FR(1/3).
FD-FR(1/3)	(8)		3 in u_j	1, 4, 11	FD counterpart of MUSCL(1/3).
FD-SR(1/2)	(43)	yes (40)	2	2	
FD-SR(1/2)-FC	(43)	included (40)	3 in u_j	2	
FD-SR(1/3)	(4)		2	3	Not a true FD κ-scheme.
FV-SR(1/3)	(23)		3 in $\overline{u}_j$	3, 6	MUSCL(1/3).
FV-FR(1/3)	(57)		2	4	FR by $f(\overline{u}_j)$ is incorrect; not a true FV κ-scheme.
FV-SR(1/2)-FC	(23)	included (62)	2	4	FC by $f(\overline{u}_j)$ is correct; still not a κ-scheme.
FV-FR(1/3,(149))	(66)		3 in $\overline{u}_j$	4	Flux function approximates Eqn. (12).

FD: Finite difference scheme; solution given as point values.
FV: Finite volume scheme; solution given as cell-averages.
FR: Flux reconstruction; $f_{j+1/2}$ computed by direct flux reconstruction.
The κ-value is indicated in parentheses.
SR: Solution reconstruction; $f_{j+1/2}$ computed as $f(u_{j+1/2})$, where $u_{j+1/2}$ is obtained
by solution reconstruction. The κ-value is indicated in parentheses.
FC: Flux correction; included is a flux-correction term raising the order of accuracy from 2 to 3.
SP: Source term included, pointwise evaluated.
SA: Source term included, cell-averaged.
SC: Source-term correction; included is a correction term to convert the source term
from cell-averaged to pointwise.
MUSCL: FV scheme with solution reconstruction; the κ-value in parentheses.
OoA: Order of accuracy.

Example: FD-SR(1/2)-SA indicates a finite-difference scheme of the κ-family with solution reconstruction using $\kappa = 1/2$ and including a cell-averaged source term; FD-SR(1/2)-FC-SA/SC is the same scheme modified by flux-and source correction terms. These are Leonard's QUICK and QUICKEST schemes, respectively. See Ref. [45].

4.2. Unsteady problem

We consider a time-dependent problem governed by Burgers' equation ($f = u^2/2$) with initial values,

$$u(x,0) = 1.5 + \sin(2\pi x), \quad x \in [0,1], \tag{83}$$

and periodic boundary conditions; thus no special boundary schemes are required. The initial values and final solution at time $t = t_f = 0.1$ are shown in Figs. 1(a) and 1(b), respectively.

The solution will develop a shock at time $t_s = \max_x (\partial u(x,0)/\partial x)^{-1} = 1/(2\pi)$; at $t = t_f$ it is still continuous. The solution and, therefore, the characteristic speed $df/du = u$ are always positive, as required by the analysis of Section 2. All schemes are integrated in time by the three-stage SSP Runge-Kutta scheme [48] for a total of 1000 time steps with a constant time step $\Delta t = 0.0001$. To verify its order of accuracy, we have each scheme solve the problem on a series of grids with the number of cells $n = 127, 255, 511, 1023, 2047$. On the finest grid, the maximum CFL number found in the solution equals $\max_x u(x)\Delta t/h = 2.5 \times 0.0001/(1/2047) \approx 0.512$, less than the stability limit 1.0 of the multistage method used; this implies stability on all coarser grids. All schemes in Table 1 are tested except FD-FR(1/2)-FC because it is identical to FD-FR(1/3).

To establish the accuracy of the FD, FV, and hybrid schemes, we need both the point values and cell-averaged values of the solution in each cell at the initial and final time. At $t = 0$, we have by (83),

$$u_j = 1.5 + \sin(2\pi x_j), \tag{84}$$

$$\overline{u}_j = 1.5 + \frac{1}{h\pi}\sin(\pi h)\sin(2\pi x_j). \tag{85}$$

The solution at the final time is obtained by the backward method of characteristics:

$$u(x, t_f) = u(\tilde{x}, 0), \tag{86}$$

Table 2
List of schemes for steady problem. Incorrect postprocessing will lower OoA to 2 for any third-order scheme.

Scheme	Reconstr.	FC possible?	Source term	OoA		Pitfall #	Comment
				unsteady	steady		
FD-SR(1/2)-SA	(43)	yes (40)	$\bar{s}_j$	2 in u_j	3 in u_j	5	Hybrid scheme, QUICK.
FD-SR(1/2)-FC-SA-SC	(43)	included (40)	$\bar{s}_j$	3 in u_j	3 in u_j	5	QUICKEST; not tested.
FD-FR(1/2)-SA	(34)	yes (40)	$\bar{s}_j$	2 in u_j	3 in u_j	5, 11	Hybrid scheme similar to QUICK.
FV-SR(1/2)-SA	(23)		$\bar{s}_j$	2	3 in u_j	5, 8	MUSCL(1/2)-SA, identical to QUICK in steady state.
FD-SR(1/3)-SA	(4)		$\bar{s}_j$	2	3 in $\bar{u}_j$	6	Hybrid scheme; becomes MUSCL(1/3)-SA in steady state.
FD-SR(1/3)-SP	(4)		s_j	2	2	10	3d-order only for linear and weakly nonlinear problems.
FV-SR(1/3)-SA	(23)		$\bar{s}_j$	3 in $\bar{u}_j$	3 in $\bar{u}_j$	6, 8, 9	MUSCL(1/3)-SA, incorrect post- or/and preprocessing lowers OoA to 2.
FV-FR(1/3)-SP	(57)		s_j	2	3 in u_j	7	Hybrid scheme; becomes FD-FR(1/3)-SP in steady state.
FV-SR(1/2)-FC-SP	(23)	included (62)	s_j	2	3 in u_j	7	Hybrid scheme; becomes FD-SR(1/2)-FC-SP in steady state.
FV-FR(1/3,(149))-SA	(66)		$\bar{s}_j$	3 in $\bar{u}_j$	3 in $\bar{u}_j$	7, 10	Not suited for hybridization.
FD-FR(1/3)-SP	(8)		s_j	3 in u_j	3 in u_j	7, 10	FD counterpart of MUSCL(1/3)-SA.

$$\tilde{x} = x - t_f u(\tilde{x}, 0). \tag{87}$$

This pair of equations can easily be solved iteratively with a fixed-point method:

$$\tilde{x}^{k+1} = x - t_f u(\tilde{x}^k, 0), \quad \tilde{x}^1 = x. \tag{88}$$

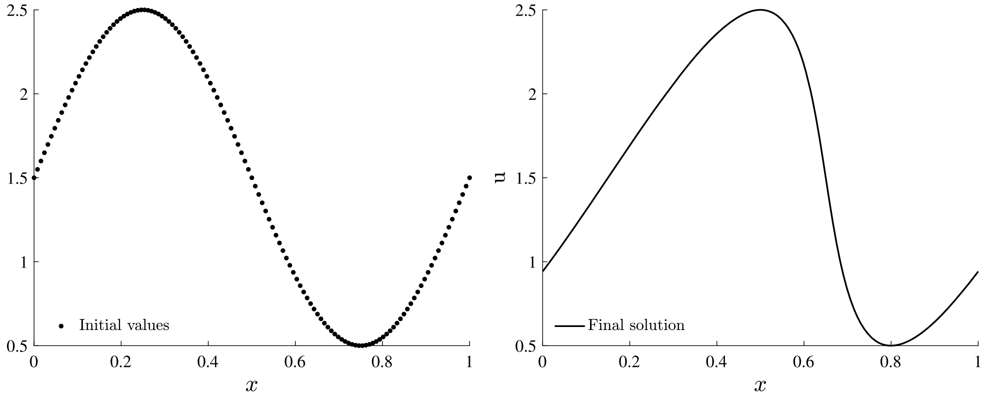
For the cell average in cell j at time t_f we find:

$$\overline{u}(t_f) = \frac{1}{h} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t_f)\, dx \tag{89}$$

$$= \frac{1}{h} \int_{x_{j-1/2}}^{x_{j+1/2}} \left(1 + t_f \frac{\partial u(\tilde{x}, 0)}{\partial \tilde{x}}\right) u(\tilde{x}, 0)\, d\tilde{x} \tag{90}$$

$$= 1.5 + \frac{t_f}{2h\pi} \left\{\cos(2\pi \tilde{x}_{j-1/2}) - \cos(2\pi \tilde{x}_{j+1/2})\right\} \left[1.5 \left\{\sin(2\pi \tilde{x}_{j+1/2}) - \sin(2\pi \tilde{x}_{j-1/2})\right\} + \frac{1}{4h} \left\{\sin(4\pi \tilde{x}_{j+1/2}) - \sin(4\pi \tilde{x}_{j-1/2})\right\} \right]. \tag{91}$$

Thus, to evaluate the cell averages of the solution at time t_f, we only need to iterate at the cell faces. It follows that we will know the point values and cell averages on all grids if we iterate at all cell centers and cell faces of the finest grid.



(a) Initial values.

(b) Final solution.

Fig. 1. Initial values and final solution for the case of the unsteady Burgers equation. (a) Initial values, (b) Final solution at $t = t_f = 0.1$.

To emphasize the distinction between cell averages and point values, we compute error norms of the numerical solution with respect to both the exact pointwise (superscript p) and cell averaged (superscript a) solutions. We always use the maximum or L_∞ norm:

$$L_\infty^p = \max_{j=1,2,\cdots,n} |w_j - u_j^{exact}|, \tag{92}$$

$$L_\infty^a = \max_{j=1,2,\cdots,n} |w_j - \overline{u}_j^{exact}|, \tag{93}$$

here w_j denotes a numerical solution, specifically, $w_j = u_j$ for schemes updating point values and $w_j = \overline{u}_j$ for those updating cell averages.

We first solve the problem with the proper initial-value discretizations: point values for FD schemes and cell averages for FV schemes. Fig. 2(a) shows the convergence of the maximum error with respect to the cell-averaged exact solution, Eqn. (93). As expected, only MUSCL(1/3) (= FV-SR(1/3)) and FV-FR(1/3,(149)) achieve third-order accuracy. On the other hand, the maximum error (92) with respect to the pointwise exact solution, displayed in Fig. 2(b), shows third-order convergence only for FD-FR(1/3) and FD-SR(1/2)-FC, again as expected. These results confirm that all other would-be third-order schemes from Pitfalls 1-4 are faulty.

Next, we solve the same problem with inconsistent initial-value discretizations: cell averages for FD schemes, point values for FV schemes. These tests fall under Pitfall 9, “Pre-processing error”. Indeed, the results in Fig. 3(a) for cell-average errors show that MUSCL(1/3) and FV-FR(1/3,(149)) no longer are third-order accurate. Now, however, FD-SR(1/3) becomes third-order accurate as it has become MUSCL(1/3), having the same SR operator applied to cell averages from the start. From the pointwise errors, shown in Fig. 3(b), we see that FD-FR(1/3) and FD-SR(1/2)-FC are no longer third-order accurate, while FV-FR(1/3) and FV-SR(1/2)-FC now yield third-order accuracy. The former has become FD-FR(1/3), having the same FR operator applied to point values from the start, the latter has become FD-SR(1/2)-FC for a similar reason.

4.3. Steady problem

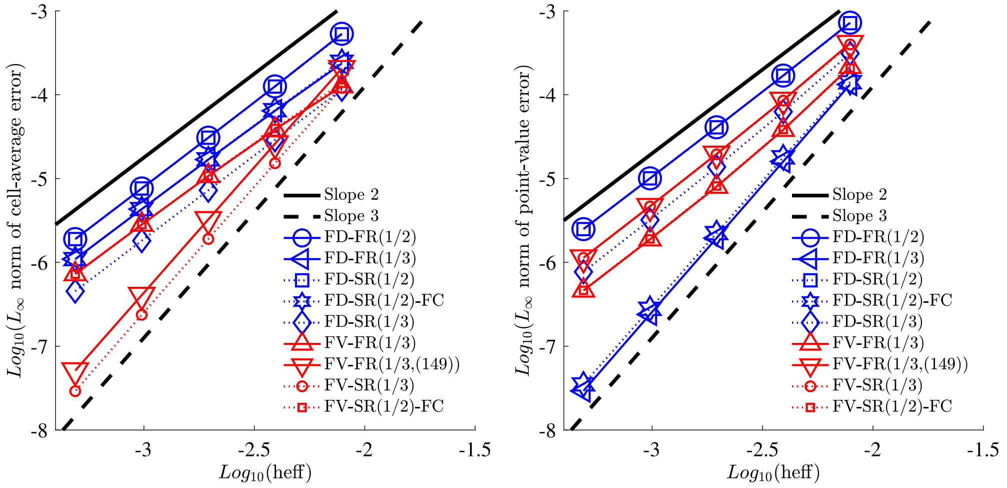
The steady-state problem we consider is governed again by Burgers’ equation and has, as its asymptotic solution, the periodic initial-value distribution of the earlier time-dependent problem:

$$u_{ss} = 1.5 + \sin(2\pi x), \quad x \in [0, 1]. \tag{94}$$

The source term that leads to this steady state is

$$s(x) = u_{ss}(x)\frac{du_{ss}(x)}{dx} = \pi[3\cos(2\pi x) + \sin(4\pi x)]. \tag{95}$$

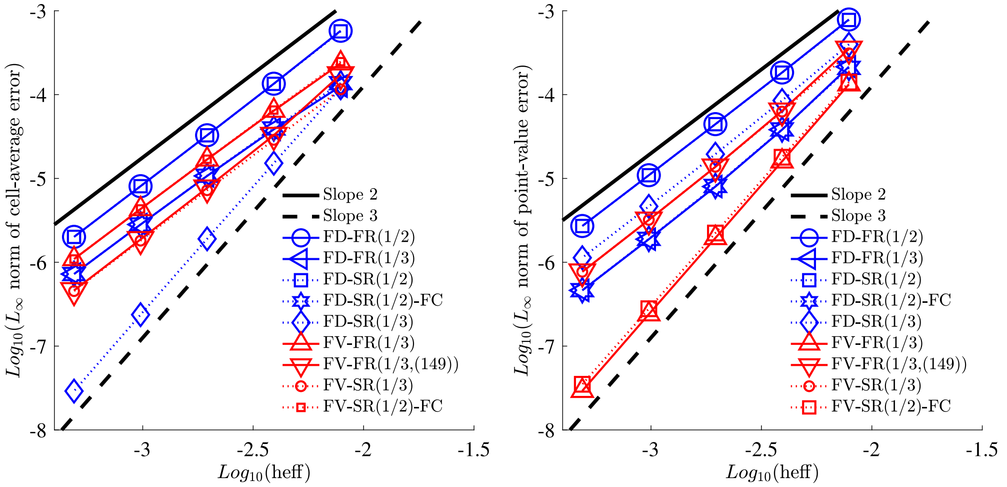
We find the steady numerical solutions by starting from the discretized steady exact solution and iterating with the FD or FV scheme using, for efficiency, an implicit but conservative time-marching method. The unit domain is discretized with n cells, $n = 62, 126, 254, 510$. Caution is needed when solving a periodic steady-state problem, as the solution is not anchored to a fixed boundary value. Instead, the domain sum of the discrete solution values, whether u_j or $\overline{u}_j$, is conserved and necessarily should start out with its exact steady-state value, in this case 1.5. To prevent the sum from drifting away from this value due to rounding errors, it is advisable to regularly reset the conservation-sum value by subtracting the drift, divided by n, from each cell value.



(a) Results at $t = 0.1$ for L^a_∞ (maximum error with respect to cell-averaged exact solution).

(b) Results at $t = 0.1$ for L^p_∞ (maximum error with respect to pointwise exact solution).

Fig. 2. Grid convergence of two different error norms for the schemes of Table 1 applied to the unsteady Burgers problem, with correct initial-value discretization type.



(a) Results at $t = 0.1$ for L^a_∞ (maximum error with respect to cell-averaged exact solution).

(b) Results at $t = 0.1$ for L^p_∞ (maximum error with respect to pointwise exact solution).

Fig. 3. Grid convergence of two different error norms for the schemes of Table 1 applied to the unsteady Burgers problem, with inconsistent initial-value discretization type.

Provided that the conservation sum is given the correct value, it does not matter in these steady-state tests what the precise initial values are, so we need not make a distinction between initial discretization choices. Instead, we emphasize the importance of the source-term discretization, by running two series of tests, one with pointwise source term values:

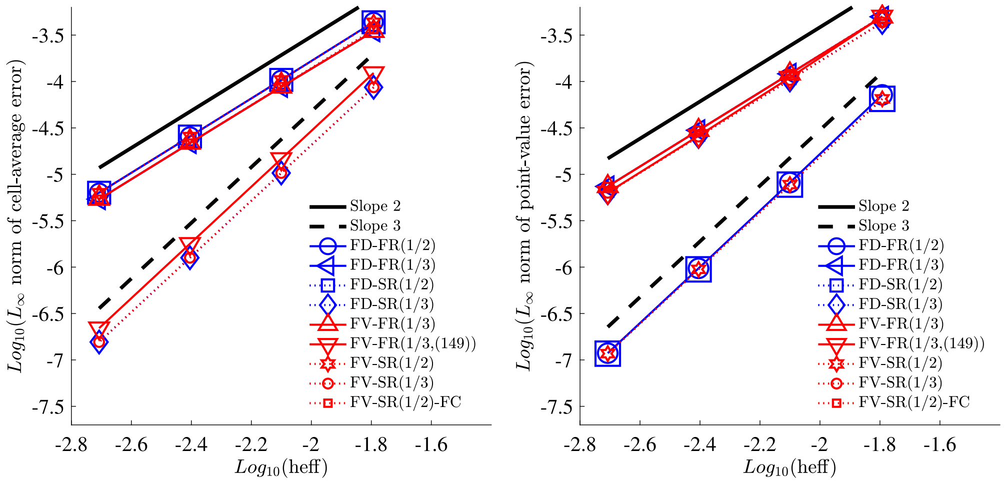
$$s_j = \pi[3\cos(2\pi x_j) + \sin(4\pi x_j)], \tag{96}$$

and one with cell-averaged source-term values:

$$\bar{s}_j = \frac{6\sin(\pi h)\cos(2\pi x_j) + \sin(2\pi h)\sin(4\pi x_j)}{2h}. \tag{97}$$

As before, the maximum solution errors are computed with respect to both the pointwise and cell-averaged steady-state solutions. Note that, among the schemes from Table 2, not all produce distinct steady solutions. QUICKEST was not tested since in essence it is the same as FD-SR(1/2)-FC-SP.

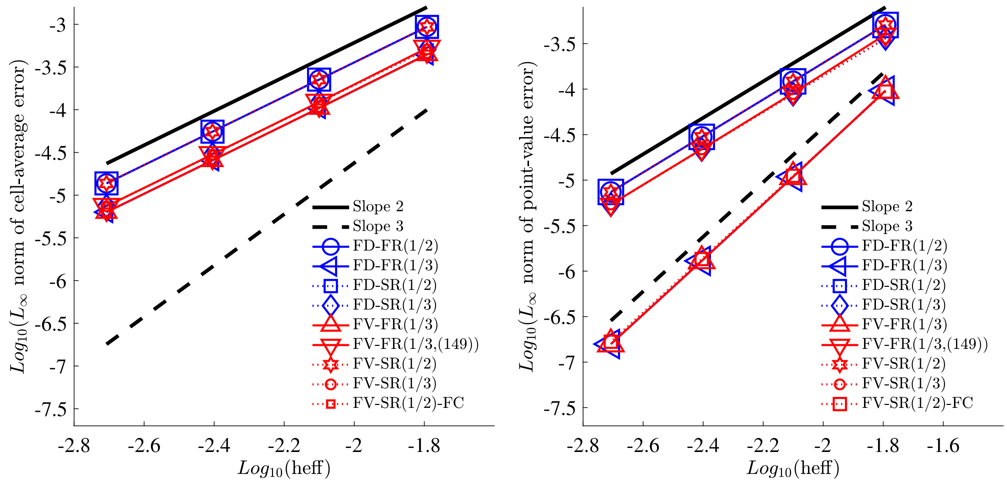
Figs. 4(a) and 4(b) show error-convergence results for all schemes from Table 2 when cell-averaged source-term values are used. For Fig. 4(a), the error was measured with respect to the cell-averaged exact steady solution. Three schemes



(a) Steady results for L_∞^a (maximum error with respect to cell-averaged exact steady solution).

(b) Steady results for L_∞^p (maximum error with respect to pointwise exact steady solution).

Fig. 4. Grid convergence of two different error norms for the schemes of Table 2 applied to the steady Burgers problem, with cell-averaged source term.



(a) Steady results for L_∞^a (maximum error with respect to cell-averaged exact steady solution).

(b) Steady results for L_∞^p (maximum error with respect to pointwise exact steady solution).

Fig. 5. Grid convergence of two different error norms for the schemes of Table 2 applied to the steady Burgers problem, with pointwise source term.

show third-order convergence MUSCL(1/3)-SA (= FV-SR(1/3)-SA), FV-FR(1/3,(149))-SA, as expected, and FD-SR(1/3)-SA; the latter scheme becomes identical to MUSCL in the steady state. The pointwise errors are presented in Fig. 4(b); again, three schemes are found to be third-order accurate:

- FD-SR(1/2)-SA or QUICK [45], the original hybrid scheme,
- FD-FR(1/2)-SA, a hybrid scheme formed by combining the FD scheme from Pitfall 1 with a cell-averaged source term,
- MUSCL(1/2)-SA (= FV-SR(1/2)-SA), a regular MUSCL scheme of second-order accuracy that becomes identical to QUICK in the steady state.

Note that FD-FR(1/3) is no longer third-order accurate when supplemented with a cell-averaged source term, an example of Pitfall 6.

Figs. 5(a) and 5(b) show error-convergence results for the schemes from Table 2 when pointwise source-term values are used. For Fig. 5(a), the error was measured with respect to the cell-averaged exact steady solution. It is seen that none of the schemes achieves third-order accuracy, demonstrating, among other things, that MUSCL(1/3) loses its third-order accuracy when supplemented with a point-valued source term (Pitfall 6). Finally, pointwise errors are shown in Fig. 5(b).

Three schemes achieve third-order convergence: FD-FR(1/3)-SP, as expected, FV-FR(1/3)-SP, and FV-SR(1/2)-FC-SP; the latter two become identical to FD-FR(1/3)-SP and FD-SR(1/2)-FC-SP, respectively, in the steady state.

4.4. *Weakly nonlinear steady problem*

This section illustrates Pitfall 10, in which a weakly nonlinear steady problem figures. We modify the steady problem of Section 4.3 by bringing a smallness parameter ϵ into the steady solution:

$$u_{ss}(x) = 1.5 + \epsilon \sin(2\pi x), \quad x \in [0, 1]. \tag{98}$$

The source term that leads to this steady state is

$$s(x) = u_{ss}(x) \frac{du_{ss}(x)}{dx} = \pi \epsilon \left[3 \cos(2\pi x) - \epsilon \sin(4\pi x)\right]. \tag{99}$$

Following Researcher T, we incorrectly implement MUSCL(1/3), by applying it to pointwise initial values and supplementing it with a pointwise source term. In this way, the scheme becomes identical to FD-SR(1/3)-SP, which is second-order accurate in the fully nonlinear regime; see Pitfalls 3 and 6. In the weakly nonlinear regime, an interesting phenomenon occurs, as seen in Fig. 6(a). It shows the grid convergence of the L_1 error norm,[6] $L_1^p = (1/n) \sum_{j=1}^{n} |u - u_{ss}|_j$, for FD-SR(1/3)-SP, one other scheme of similar behavior, and a third-order comparison scheme. For the first scheme, we observe that the log-log plot of the error as a function of decreasing cell width h has a slope -3 between grid 1 ($h = 1/30$) and grid 2 ($h = 1/62$), and a slope -2 between grid 4 ($h = 1/254$) and grid 6 ($h = 1/1022$). Between grids 2 and 4, a transition occurs. This behavior can be predicted by a truncation-error analysis similar to the one in Pitfall 3. As derived in the Appendix, for every ϵ, there is a unique value $h_t(\epsilon)$ of the cell width for which the amplitudes of second- and third-order truncation error components are equal; we may define this value as the center of the transition region; we find:

$$h_t(\epsilon) = \frac{\epsilon}{6\pi(1+\epsilon)}, \tag{100}$$

or:

$$\frac{1}{h_t} = 6\pi \left(\frac{1}{\epsilon} + 1\right). \tag{101}$$

The value of ϵ used in the weakly nonlinear test of scheme FD-SR(1/3)-SP is 0.16, yielding $h_t = 1/137$, clearly inside of the transition as Fig. 6 shows.

The other curve with similar behavior in Fig. 6(a) is for scheme FV-FR(1/3,(149))-SP, incorrectly applied to pointwise initial and source values. The scheme is third-order accurate for a linear flux, when applied to pointwise initial and source values (it then is identical to FD-SR(1/3)-SP), but reduces to second order accurate when the problem becomes nonlinear; this is because Eqn. (69) becomes meaningless if all $\overline{u}$-values are replaced by point values. The equation produces a purely nonlinear error $\sim h^2$ in $f_{j+1/2}$, equal to the one for FD-SR(1/3)-SP, but no purely nonlinear error $\sim h^3$ owing to the symmetry of the difference operators in Eqn. (69). The quasilinear truncation error in $f_{j+1/2}$ is again the same as for FD-SR(1/3)-SP. The net effect is that the third-order error for FV-FR(1/3,(149)) is somewhat greater than for FD-SR(1/3)-SP and the transition moves to smaller h, the equation for $h_t(\epsilon)$ becomes:

$$\frac{1}{h_t(\epsilon)} = 6\pi \left(\frac{1}{\epsilon} + \frac{8}{3}\right), \tag{102}$$

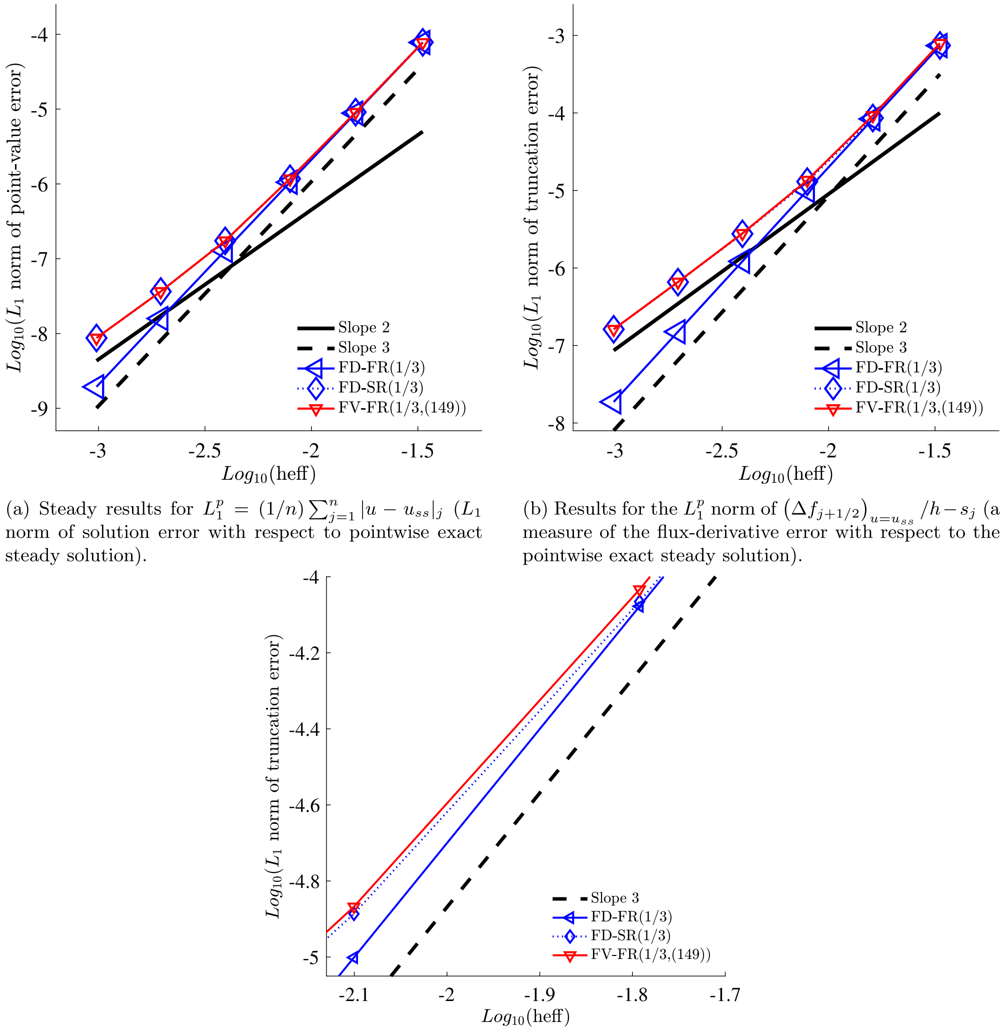
So that $h_t(0.16) = 1/168$, also inside of the scheme's transition region.

The third graph in Fig. 6(a) is for scheme FD-FR(1/3)-SP, properly applied to pointwise initial and source values: it is third-order accurate for the whole range of grids. It has the same quasilinear error as the other two schemes but none of the nonlinear errors. Although the nonlinear error $\sim h^3$ lowers the quasilinear error of scheme FD-SR(1/3)-SP, the nonlinear error $\sim h^2$ negates that effect for all but the coarsest grids, $h < 1/5$. Thus, FD-FR(1/3)-SP is better off without those nonlinear errors and has the lowest error of all three schemes tested.

From the above truncation-error analysis, we would expect the entire curve for FV-FR(1/3,(149))-SP to lie slightly above and to the right of the curve for FD-SR(1/3)-SP, but this is not the case: the curves intersect, and so does the curve for FD-FR(1/3)-SP, which supposedly has the lowest error of the three. This confirms that the error in f_x does not represent the error in u in detail.

To pursue this matter further, we computed the error $\frac{(\Delta f_{j+1/2})_{u=u_{ss}}}{h} - s(x_j)$ on all grids for the three schemes, and in each case obtained the pointwise-error norm $L_1^p = (1/n) \sum_{j=1}^{n} |error|_j$. The results are plotted in Fig. 6(b). The three curves

[6] This norm is more suited to bring out subtle solution differences than the maximum norm; the latter would emphasize local errors that do not necessarily occur in the same location for the different schemes.



(a) Steady results for $L_1^p = (1/n)\sum_{j=1}^{n} |u - u_{ss}|_j$ (L_1 norm of solution error with respect to pointwise exact steady solution).

(b) Results for the L_1^p norm of $\left(\Delta f_{j+1/2}\right)_{u=u_{ss}}/h - s_j$ (a measure of the flux-derivative error with respect to the pointwise exact steady solution).

(c) Enlargement of coarse-grid region of Figure 6(b).

Fig. 6. Grid convergence of solution and flux-derivative errors for some FD schemes applied to the weakly nonlinear steady Burgers problem, with pointwise source term.

are so close together that it is hard to determine their relative position; therefore, an enlargement of the region between the second and third grids is provided in Fig. 6(c). It can now be seen that FV-FR(1/3,(149)) indeed has the largest error, followed by FD-SR(1/3)-SP and FD-FR(1/3)-SP, successively.

5. Conclusions

In the preceding sections, we have done away with misconceptions regarding the accuracy of MUSCL schemes. We have shown by analysis and numerical experiments that the FV MUSCL scheme with $\kappa = 1/3$ and the related FD version both are third-order accurate for all times. The confusion arises because there are some hybrid FD-FV schemes with $\kappa = 1/2$ that are third-order accurate when marched to a steady solution but second-order accurate when marching in time.

We summarize our results by listing all schemes that are third-order accurate when marching in time (Table 3) and all, mostly hybrid, schemes that achieve third-order accuracy only in the steady state (Table 4). Note that for some of the latter kind, the solution discretization in the accurate steady state may have switched from point value to cell average or vice versa. The schemes in Table 3 are assumed to be combined with a third- or higher-order time-marching method; those in Table 4 may be combined with any effective marching method.

A final remark: the above schemes, even when programmed correctly, may produce disappointing results if pre- or post-processing have been done incorrectly. Specifically, for time-accurate FV calculations, the initial values must be cell-averaged, and output must be compared to a cell-averaged exact solution; for time-accurate FD calculations, the initial values and reference solution must be point-valued. For steady-state calculations, the type of initial-value discretization

Table 3
Schemes that are third-order accurate at all times.

Scheme	Comment
FD-FR(1/3)-SP	FD counterpart of MUSCL(1/3)-SA
FD-SR(1/2)-FC-SP	Unique combination of SR and FC in an FD scheme
FD-SR(1/2)-FC-SA-SC	QUICKEST [45], the same as FD-FR(1/3)-SP within $O(h^4)$
FV-SR(1/3)-SA	MUSCL(1/3)-SA
FV-FR(1/3,(149))-SA	Close approximation to basic FV scheme (11)

Table 4
Schemes that are third-order accurate only in steady state.

Scheme	Comment
FD-FR(1/2)-SA	Hybrid; steady point values are third-order accurate.
FD-SR(1/2)-SA	Hybrid, QUICK[31], steady point values are third-order accurate.
FV-SR(1/2)-SA	MUSCL(1/2)-SA, not hybrid; steady point values are third-order accurate; identical to QUICK in steady state.
FD-SR(1/3)-SA	Hybrid; steady cell averages are third-order accurate; identical to MUSCL(1/3)-SA in steady state; the non-hybrid form (with SP instead of SA) is not a κ-scheme.
FV-FR(1/3)-SP	Fluxes computed as $f(\overline{u}_j)$; hybrid; steady point values are third-order accurate; identical to FD-FR(1/3)-SP in steady state; the non-hybrid form (with SA instead of SP) is not a κ-scheme.
FV-SR(1/2)-FC-SP	Flux correction using $f(\overline{u}_j)$; hybrid; steady point values are third-order accurate; identical to FD-SR(1/2)-FC-SP in steady state; the non-hybrid form (with SA instead of SP) is not a κ-scheme.

does not matter, but in the steady state it does. In Table 4, the discretization type that carries the highest accuracy is indicated. For all FV schemes it switches to point values, for one FD scheme it switches to cell averages. The reference solution should be discretized accordingly.

CRediT authorship contribution statement

Bram van Leer: Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Writing – original draft, Writing – review & editing. **Hiroaki Nishikawa:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The second author gratefully acknowledges support from Software CRADLE (part of Hexagon), the U.S. Army Research Office under the contract/grant number W911NF-19-1-0429 with Dr. Matthew Munson as the program manager, and the Hypersonic Technology Project, through the Hypersonic Airbreathing Propulsion Branch of the NASA Langley Research Center, under Contract No. 80LARC17C0004. We would like thank Kyle Anderson (NASA Langley) for reviewing the manuscript.

Appendix. Derivation of a critical mesh size for transition from third- to second-order accuracy in the weakly nonlinear test

We repeat the analysis in Pitfall 3 for the specific case of Burgers' equation

$$f = u^2/2, \quad a = \frac{df}{du}, \quad a(u,v) = \frac{f(v)-f(u)}{v-u} = \frac{u+v}{2}. \tag{103}$$

Isolating the flux $f^{FD}_{j+1/2}$ obtained by direct flux reconstruction leads to:

$$f_{j+1/2} = f^{FD}_{j+1/2} + \frac{1-\kappa}{8}(u^{SR}_{j+1/2} - u_{j-1})(u_j - u_{j-1}) + \frac{1+\kappa}{8}(u^{SR}_{j+1/2} - u_{j+1})(u_{j+1} - u_j), \tag{104}$$

where the superscript SR denotes solution reconstruction:

$$u^{SR}_{j+1/2} = u_j + \frac{\Delta u_j + \Delta u_{j+1}}{4} + \frac{\kappa}{4}\Delta^2 u_{j+1}. \tag{105}$$

The last equation may also be written in the following ways:

$$u^{SR}_{j+1/2} = u_j + \frac{\Delta u_{j+1}}{2} - \frac{1-\kappa}{4}\Delta^2 u_{j+1} \tag{106}$$

$$= u_j + \frac{\Delta u_j}{2} + \frac{1+\kappa}{4}\Delta^2 u_{j+1}. \tag{107}$$

We use Eqns. (106) and (107) to develop the terms in Eqn. (104) proportional to $u_{j+1} - u_j$ and $u_j - u_{j-1}$, respectively, and arrive at the following expression for the purely nonlinear error in $f_{j+1/2}$:

$$\begin{aligned} PNLE_{j+1/2} &= \frac{1+\kappa}{4}\left[-\frac{\Delta u_{j+1}}{4} - \frac{1-\kappa}{8}\Delta^2 u_{j+1}\right]\Delta u_{j+1} + \frac{1-\kappa}{4}\left[\frac{3\Delta u_j}{4} + \frac{1+\kappa}{8}\Delta^2 u_{j+1}\right]\Delta u_j \\ &= \frac{1}{8}\left(\frac{1}{2} - \kappa\right)\left[(\Delta u_{j+1})^2 + (\Delta u_j)^2\right] - \frac{1}{8}\left(1 - \frac{\kappa}{2}\right)\left[(\Delta u_{j+1})^2 - (\Delta u_j)^2\right] \\ &\quad - \frac{1}{32}(1-\kappa^2)(\Delta^2 u_{j+1})^2 \end{aligned} \tag{108}$$

Note that this is the exact discrete purely nonlinear error in $f_{j+1/2}$; we shall further use its continuum representation $PNLE(x)$[7]:

$$PNLE(x_{j+1/2}) = \frac{1}{4}\left(\frac{1}{2} - \kappa\right)(u_x)^2_j h^2 - \frac{1}{4}\left(1 - \frac{\kappa}{2}\right)(u_x u_{xx})_j h^3 + O(h^4). \tag{109}$$

In an FD scheme, the flux derivative, rather than the interface flux, carries the accuracy of the scheme (see Sections 3.2 and 3.13), so we must analyze the error in $\Delta f_{j+1/2}/h$:

$$\frac{\Delta f_{j+1/2}}{h} = \frac{\Delta f^{FD}_{j+1/2}}{h} + (PNLE_x)_j, \tag{110}$$

where:

$$(PNLE_x)_j = \frac{1}{2}\left(\frac{1}{2} - \kappa\right)(u_x u_{xx})_j h^2 - \frac{1}{4}\left(1 - \frac{\kappa}{2}\right)[(u_x u_{xx})_x]_j h^3 + O(h^4). \tag{111}$$

Next we turn to computing the quasilinear error in $\Delta f_{j+1/2}/h$, which also is the full error in $\Delta f^{FD}_{j+1/2}/h$, with:

$$\Delta f^{FD}_{j+1/2} = \Delta f_j + \frac{\Delta^2 f_{j+1} + \Delta^2 f_j}{4} + \frac{\kappa}{4}\Delta^3 f_{j+1}. \tag{112}$$

A straightforward Taylor expansion of f about x_j produces the quasilinear error $QLE(x)$ in $\Delta f^{FD}_{j+1/2}/h$:

$$\begin{aligned} \frac{1}{h}\Delta f^{FD}_{j+1/2} &= (f_x)_j + QLE(x_j) \\ &= (f_x)_j + (1-3\kappa)(f_{xxx})_j \frac{h^2}{12} + \frac{1}{8}(1-\kappa)(f_{xxxx})_j h^3 + O(h^4) \\ &= (f_x)_j + (1-3\kappa)(f_{xxx})_j \frac{h^2}{12} + \frac{1}{8}(1-\kappa)[(uu_x)_{xxx}]_j h^3 + O(h^4). \end{aligned} \tag{113}$$

For the total truncation error $TTE(x)$ in $\Delta f_{j+1/2}/h$, we therefore find:

[7] Here $u(x)$ is to be regarded as a $C^{(3)}$ fit to the discrete data u_j.

$$\begin{aligned} TTE(x_j) &= QLE(x_j) + (PNLE_x)_j \\ &= \left\{ \frac{1-\kappa}{8}[(uu_x)_{xxx}]_j - \frac{1}{4}\left(1-\frac{\kappa}{2}\right)[(u_x u_{xx})_x]_j \right\} h^3 \\ &\quad + \frac{1}{2}\left(\frac{1}{2}-\kappa\right)(u_x u_{xx})_j h^2 + (1-3\kappa)(f_{xxx})_j \frac{h^2}{12} + O(h^4). \end{aligned} \tag{114}$$

The truncation error in $\Delta f^{FD}_{j+1/2}/h$ is related to the error in u_j (not readily available in a steady-state problem) and has the expected structure. To determine where the transition from third- to second-order accuracy occurs, it suffices to consider the balance of terms in TTE.

When the numerical solution has become steady, there are several ways to interpret TTE as a flux-derivative error. The expression $\Delta f_{j+1/2}/h - (\partial f_{ss}/\partial x)_j$ is not one of these, because it vanishes in the steady state:

$$\Delta f_{j+1/2}/h = s_j = (\partial f_{ss}/\partial x)_j \quad \text{in the steady state.} \tag{115}$$

But we may use these relations to obtain:

$$\left(\frac{\partial(f-f_{ss})}{\partial x}\right)_j = -TTE(x_j) = -\{TTE(u(x)=u_{ss}(x))\}_{x=x_j} + O(h^4). \tag{116}$$

And, since Eqn. (114) is valid for any pair of functions u, $f(u)$, we may insert $u = u_{ss}$ and obtain:

$$\frac{\Delta(f-f_{ss})_{j+1/2}}{h} = s_j - \frac{\Delta(f_{ss})_{j+1/2}}{h} = \left(Res_j\right)_{u=u_{ss}} = -\{TTE(u(x)=u_{ss}(x))\}_{x=x_j}. \tag{117}$$

It is seen that these two error measures differ by $O(h^4)$. Henceforth we shall use the second one since it is readily computable. We are now ready to get specific. Into Eqn. (117), we insert the value $\kappa = 1/3$, and u_{ss} according to Eqn. (98). The result is (subscript j omitted):

$$\begin{aligned} -Res(x) = TTE(x) &= (2\pi)^4\epsilon\left[\frac{1}{12}\{1.5\sin(2\pi x) - 4\epsilon\cos(4\pi x)\} + \frac{5}{24}\epsilon\cos(4\pi x)\right]h^3 \\ &\quad - \frac{(2\pi)^3}{24}\epsilon^2\sin(4\pi x)h^2 + O(h^4) \\ &= \frac{(2\pi)^4}{12}\epsilon[1.5(\sin(2\pi x) - \epsilon\cos(4\pi x)]h^3 - \frac{(2\pi)^3}{24}\epsilon^2\sin(4\pi x)h^2 + O(h^4). \end{aligned} \tag{118}$$

The error is built up from two wave forms with amplitudes $\sim h^3$ and $\sim h^2$:

$$\text{Amplitude} \sim h^3 = \frac{(2\pi)^4}{8}\epsilon(1+\epsilon)h^3, \tag{119}$$

$$\text{Amplitude} \sim h^2 = \frac{(2\pi)^3}{24}\epsilon^2 h^2. \tag{120}$$

For every ϵ, there is a unique value $h_t(\epsilon)$ of the cell width for which the amplitudes of the two error components are equal; we may define this value as the center of the transition region; we find:

$$h_t(\epsilon) = \frac{\epsilon}{6\pi(1+\epsilon)}, \tag{121}$$

or:

$$\frac{1}{h_t} = 6\pi\left(\frac{1}{\epsilon}+1\right). \tag{122}$$

References

[1] B. van Leer, Towards the ultimate conservative difference scheme. III. Upstream-centered finite difference schemes for ideal compressible flow, J. Comput. Phys. 23 (1977) 263–275.

[2] B. van Leer, Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection, J. Comput. Phys. 23 (1977) 276–299.

[3] B. van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101–136.

[4] S.K. Godunov, A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics, Mat. Sb. (N. S.) 47(89) (3) (1959) 271–306.

[5] G.D. van Albada, B. van Leer, W.W. Roberts J, A comparative study of computational methods in cosmic gas dynamics, Astron. Astrophys. 108 (1) (1982) 76–84.

[6] B. van Leer, Upwind and high-resolution methods for compressible flow: from donor cell to residual distribution schemes, in: Proc. of 16th AIAA Computational Fluid Dynamics Conference, 2003, AIAA Paper 2003-3559.

[7] B. van Leer, C.H. Tai, K.G. Powell, Design of optimally-smoothing multi-stage schemes for the Euler equations, in: Proc. of 15th AIAA Computational Fluid Dynamics Conference, 1989, AIAA Paper 1989-1933.

[8] C.-W. Shu, S.J. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, J. Comput. Phys. 83 (1989) 32–78.

[9] C.-W. Shu, High order weighted essentially nonoscillatory schemes for convection dominated problems, SIAM Rev. 51 (1) (2009) 82–126.

[10] B. Cockburn, C.W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws: V. Multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.

[11] X. Liu, L. Xuan, H. Luo, Y. Xia, A reconstructed discontinuous Galerkin method for the compressible Navier-Stokes equations on three-dimensional hybrid grids, Comput. Fluids 152 (2017) 217–230.

[12] H. Luo, Y. Xia, S. Spiegel, R. Nourgaliev, Z. Jiang, A reconstructed discontinuous Galerkin method based on a hierarchical WENO reconstruction for compressible flows on tetrahedral grids, J. Comput. Phys. 236 (1) (2013) 477–492.

[13] B. van Leer, Upwind-difference methods for aerodynamic problems governed by the Euler equations, in: Fluid Mechanics, in: Lectures in Applied Mathematics, vol. 22, 1985, pp. 327–335.

[14] J.E. Fromm, A method for reducing dispersion in convective difference schemes, J. Comput. Phys. 3 (1968) 176–189.

[15] W.K. Anderson, J.T. Thomas, B. van Leer, Comparison of finite volume flux vector splitting for the Euler equations, AIAA J. 24 (9) (1986) 1453–1460.

[16] J.T. Thomas, B. van Leer, R.W. Walters, Implicit Flux-Split Schemes for the Euler Equations, AIAA Paper 85-1680, 1985.

[17] P.A. Ullrich, C. Jablonowski, B. van Leer, High-order finite-volume methods for the shallow-water equations on the sphere, J. Comput. Phys. 229 (2020) 6104–6134.

[18] B.P. Leonard, S. Mokhtari, Beyond first-order upwinding: the ultra-sharp alternative for non-oscillatory steady-state simulation of convection, Int. J. Numer. Methods Fluids 30 (1990) 729–766.

[19] H. Wu, L. Wand, G. Sun, Non-existence of third order MUSCL schemes unified construction of ENO schemes and a new discontinuity sharpening technique - stiff source term approach, Comput. Fluid Dyn. Rev. 1998 (1998) 300–317.

[20] M. Liu, Computational study of convective-diffusive mixing in a microchannel mixer, Chem. Eng. Sci. 66 (2011).

[21] S.M. Kresta III, A.W. E, D.S. Dickey, V.A. Atiemo-Obeng (Eds.), Advances in Industrial Mixing: a Companion to the Handbook of Industrial Mixing, John Wiley & Sons, 2015.

[22] B. Koren, A Robust Upwind Discretization Method for Advection, Report NM-R9308, April 1993.

[23] B.P. Leonard, Order of accuracy of QUICK and related convection-diffusion schemes, Appl. Math. Model. 19 (1995) 644–653.

[24] Y. Chen, R.A. Falconer, Modified forms of the third-order convection, second-order diffusion scheme for the advection-diffusion equation, Adv. Water Resour. 17 (1994) 147–170.

[25] R.W. Johnson, R.J. Mackinon, Equivalent versions of the QUICK scheme for finite-difference and finite-volume numerical methods, Commun. Appl. Numer. Methods 8 (1992) 841–847.

[26] H.Q. Yang, R.E. Harris, Development of vertex-centered high-order schemes and implementation in FUN3D, AIAA J. 54 (2016) 3742–3760.

[27] B. Song, R.S. Amano, A high-order bounded discretization scheme, in: R.S. Amano, B. Sudén (Eds.), Computational Fluid Dynamics and Heat Transfer: Emerging Topics, WIT Press, 2011, pp. 3–5.

[28] C.O.E. Burg, Higher Order Variable Extrapolation for Unstructured Finite Volume RANS Flow Solvers, AIAA Paper 2005-4999, 2005.

[29] D. Zhang, C. Jiang, D. Liang, L. Cheng, A review on TVD schemes and a refined flux-limiter for steady-state calculations, J. Comput. Phys. 302 (2015).

[30] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.

[31] R.H. Pletcher, J.C. Tannehill, D.A. Anderson, Computational Fluid Mechanics and Heat Transfer, 3rd ed., Taylor & Francis, 2013.

[32] C. Hirsch, Numerical Computation of Internal and External Flows, vol. 1, 2nd ed., Butterworth-Heinemann, 2007.

[33] P. Wesseling, Principles of Computational Fluid Dynamics, Springer, 2000.

[34] T.K. Sengupta, High Accuracy Computing Methods: Fluid Flows and Wave Phenomena, Cambridge University Press, 2013.

[35] T.J. Barth, P.O. Frederickson, Higher Order Solution of the Euler Equations on Unstructured Grids using Quadratic Reconstruction, AIAA Paper 90-0013, 1990.

[36] A. Jalali, C. Ollivier-Gooch, Higher-order unstructured finite volume RANS solution of turbulent compressible flows, Comput. Fluids 143 (2017) 32–47.

[37] P. Tsoutsanis, Extended bounds limiter for high-order finite-volume schemes on unstructured meshes, J. Comput. Phys. 362 (2018) 69–94.

[38] H. Nishikawa, Economically high-order unstructured-grid methods: clarification and efficient FSR schemes, Int. J. Numer. Methods Fluids (2021), https://doi.org/10.1002/fld.5028, in press, arXiv:2012.08213 [math.NA], 15 December 2020.

[39] F.M. Denaro, Towards a new model-free simulation of high-Reynolds-flows: local average direct numerical, Int. J. Numer. Methods Fluids 23 (1996) 125–142.

[40] A. Aprovitola, P. D'Ambra, F.M. Denaro, D. di Serafino, S. Filippone, SParC-LES: enabling large eddy simulations with parallel sparse matrix computation tools, Comput. Methods Appl. Mech. Eng. 70 (2015) 2688–2700.

[41] B.P. Leonard, A stable and accurate convective modelling procedure based on quadratic upstream interpolation, Comput. Methods Appl. Mech. Eng. 19 (1979) 59–98.

[42] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357–372.

[43] H. Nishikawa, On false accuracy verification of UMUSCL scheme, Commun. Comput. Phys. 30 (2021) 1037–1060.

[44] R. Zhang, M. Zhang, C.-W. Shu, On the order of accuracy and numerical performance of two classes of finite volume WENO schemes, Commun. Comput. Phys. 9 (3) (2011) 807–827.

[45] H. Nishikawa, The QUICK scheme is a third-order finite-volume scheme with point-valued numerical solutions, Int. J. Numer. Methods Fluids 93 (April 2021) 2311–2388.

[46] D. Zhong, C. Sheng, A new method towards high-order WENO schemes on structured and unstructured grids, Comput. Fluids 200 (2020) 104453.

[47] D.C. Dement, S.M. Ruffin, Higher order cell centered finite volume schemes for unstructured Cartesian grids, in: 56th AIAA Aerospace Sciences Meeting, 2018, AIAA Paper 2018-1305, Kissimmee, Florida.

[48] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Rev. 43 (1) (2001) 89–112.