



# Efficient Alternative Finite Difference WENO Schemes for Hyperbolic Systems with Non-conservative Products

Dinshaw S. Balsara<sup>1,2</sup> · Deepak Bhojia<sup>1</sup> · Chi-Wang Shu<sup>3</sup> · Harish Kumar<sup>4</sup>

Received: 17 July 2023 / Revised: 6 December 2023 / Accepted: 19 January 2024  
© Shanghai University 2024

## Abstract

Higher order finite difference Weighted Essentially Non-oscillatory (WENO) schemes for conservation laws represent a technology that has been reasonably consolidated. They are extremely popular because, when applied to multidimensional problems, they offer high order accuracy at a fraction of the cost of finite volume WENO or DG schemes. They come in two flavors. There is the classical finite difference WENO (FD-WENO) method (Shu and Osher in *J. Comput. Phys.* 83: 32–78, 1989). However, in recent years there is also an alternative finite difference WENO (AFD-WENO) method which has recently been formalized into a very useful general-purpose algorithm for conservation laws (Balsara et al. in *Efficient alternative finite difference WENO schemes for hyperbolic conservation laws*, submitted to CAMC, 2023). However, the FD-WENO algorithm has only very recently been formulated for hyperbolic systems with non-conservative products (Balsara et al. in *Efficient finite difference WENO scheme for hyperbolic systems with non-conservative products*, to appear CAMC, 2023). In this paper, we show that there are substantial advantages in obtaining an AFD-WENO algorithm for hyperbolic systems with non-conservative products. Such an algorithm is documented in this paper. We present an AFD-WENO formulation in a fluctuation form that is carefully engineered to retrieve the flux form when that is warranted and nevertheless extends to non-conservative products. The method is flexible because it allows any Riemann solver to be used. The formulation we arrive at is such that when non-conservative products are absent it reverts exactly to the formulation in the second citation above which is in the exact flux conservation form. The ability to transition to a precise conservation form when non-conservative products are absent ensures, via the Lax-Wendroff theorem, that shock locations will be exactly captured by the method. We present two formulations of AFD-WENO that can be used with hyperbolic systems with non-conservative products and stiff source terms with slightly differing computational complexities. The speeds of our new AFD-WENO schemes are compared to the speed of the classical FD-WENO algorithm from the first of the above-cited papers. At all orders, AFD-WENO outperforms FD-WENO. We also show a very desirable result that higher order variants of AFD-WENO schemes do not cost that much more than their lower order variants. This is because the larger number of floating point operations associated with larger stencils is almost very efficiently amortized by the CPU when the AFD-WENO code is designed to be cache friendly. This should have great, and very beneficial, implications for the role of our AFD-WENO schemes in the Peta- and Exascale computing. We apply

Extended author information available on the last page of the article

the method to several stringent test problems drawn from the Baer-Nunziato system, two-layer shallow water equations, and the multicomponent debris flow. The method meets its design accuracy for the smooth flow and can handle stringent problems in one and multiple dimensions. Because of the pointwise nature of its update, AFD-WENO for hyperbolic systems with non-conservative products is also shown to be a very efficient performer on problems with stiff source terms.

**Keywords** Hyperbolic PDEs · Numerical schemes · Conservation laws · Stiff source terms · Finite difference WENO

**Mathematics Subject Classification** 65M06 · 65M22 · 65N06 · 35Q35 · 76T10 · 76M20

## 1 Introduction

In a landmark paper, Harten et al. [27] developed finite volume Essentially Non-Oscillatory (ENO) schemes which could evolve conservation laws with better than second order of accuracy. Soon thereafter, Shu and Osher [43, 44] developed finite difference versions of ENO schemes which were substantially faster than their finite volume counterparts in multi-dimensions. Early versions of ENO schemes suffered from the deficiency that rapid switching of the stencils could diminish the order of accuracy. Weighted Essentially Non-Oscillatory (WENO) schemes overcame this deficiency by making a non-linearly hybridized weighting of the reconstruction polynomials from multiple stencils (Liu et al. [33], Jiang and Shu [30]). The methods were extended to the seventh, ninth, and eleventh orders by Balsara and Shu [11] and much later to seventeenth order by Gerolymos et al. [26]. Some of the early deficiencies of WENO schemes stemmed from a loss of accuracy at critical points, and a way out of this problem was presented in Henrick et al. [28], Borges et al. [12], and Castro et al. [13]. For a comprehensive review of WENO schemes, see Shu [41, 42].

For a long time now, progress on finite difference WENO schemes has focused on improving their performance for hyperbolic systems of conservation laws. The original approach by Shu and Osher [43, 44] was based on making a flux vector splitting of the LLF flux into left- and right-going contributions and then carrying out an upwinded reconstruction of those fluxes. However, in recent years, newer hyperbolic PDE systems have come to be of interest which have non-conservative products. As a result, it is very desirable to have efficient finite difference WENO methods that can evolve hyperbolic systems with non-conservative products. In the literature, the update of PDEs with non-conservative products is mainly done in a fluctuation form, which is quite different from a flux form. Since the original Shu and Osher [43, 44] formulation of the finite difference WENO was based on a flux form, it is understandable that the extension to the fluctuation form must have seemed difficult. As far as we know, the first effort to produce a finite difference WENO scheme for treating hyperbolic systems with non-conservative products was made by Balsara et al. [5]. The method we presented was cast in a fluctuation form and was, therefore, capable of treating PDEs that are in the conservation form as well as PDEs with non-conservative products. The problem with a fluctuation form is that unless one engineers the situation perfectly it is hard to retrieve a flux conservative form from a fluctuation form. Thanks to the high accuracy of the WENO methods, we were able to show that the method in Balsara et al. [5] was effectively conservative for all intents and purposes. In other words, even if a

problem was dominated by strong shocks, the conservation was preserved to a high level of accuracy in all situations where it should have been preserved. However, the method was not exactly conservative down to the machine accuracy. In this paper we precisely engineer the transcription from the flux form to the non-conservative form and vice versa. Consequently, this paper presents the first finite difference WENO methods that are exactly conservative, i.e., they retain a flux form, when the conservation is warranted; but the methods presented here can nevertheless handle PDEs with non-conservative products.

The paper by Shu and Osher [44] contained a thorough description of the classical finite difference WENO (FD-WENO) that has by now become very successful and extremely well known. This well-known FD-WENO formulation is based on the aforementioned flux vector splitting. However, the first paper by Shu and Osher [43] also contained the description of another alternative finite difference WENO (AFD-WENO) algorithm that was not followed up to any significant extent later, until quite recently. A paper by Merriman [34] made some headway in clarifying the AFD-WENO algorithm. Jiang et al. [29] labeled the AFD-WENO algorithm in Shu and Osher [43] as the “alternative formulation of the finite difference WENO scheme”. (To explain further, it is an alternative algorithm to the original, well-known and well-used finite difference WENO algorithm from Shu and Osher [44].) Interest in AFD-WENO has been sporadic (Jiang et al. [29, 31], Zheng et al. [49], Gao et al. [25]). In Balsara et al. [6] we presented an AFD-WENO formulation for hyperbolic conservation laws that overcame its three major weaknesses. First, it was shown that the entire algorithm in the conservation form could be derived by using a computer algebra system, which demystifies the derivation of the equations. Second, AFD-WENO relies on WENO interpolation rather than on the much better known WENO reconstruction. Therefore, in Balsara et al. [6] we provided all the explicit formulae needed for implementing AFD-WENO with the use of the WENO interpolation. Third, the AFD-WENO algorithm requires higher order derivatives of the fluxes to be available at zone boundaries. Since those derivatives are usually obtained by finite differencing the zone-centered fluxes, they can become a source of spurious oscillations when the solution is non-smooth. However, please note that the inclusion of those fluxes is also crucially important for preserving the order property when the solution is smooth. Balsara et al. [6] invented a novel WENO interpolation which takes the first derivatives of the fluxes at zone centers as its inputs and returns the requisite non-linearly hybridized higher order derivatives of flux-like terms at the zone boundaries as its output. This non-linear hybridization stabilizes the evaluation of the higher derivatives of the fluxes at zone boundaries. Removal of these three barriers makes the implementation of AFD-WENO for conservation laws much more accessible to the greater community.

A good formulation of AFD-WENO has many desirable traits. In recent years we have seen many different Riemann solvers emerge which have special attributes that make them very useful in various application areas. Those Riemann solvers do not fit well into the strictures of classical FD-WENO because of its focus on reconstructing the fluxes. The AFD-WENO algorithm is free of such strictures—any type of Riemann solver can be invoked in a pointwise fashion at the zone boundaries. This makes a well-designed AFD-WENO very broadly applicable to many application areas. Classical FD-WENO also does not take well to preserving the free stream condition on curvilinear meshes; whereas AFD-WENO can indeed take well to curvilinear meshes (Jiang et al. [29, 31]). As a result, we see that it would be very desirable to arrive at an AFD-WENO formulation in the fluctuation form that is carefully engineered to retrieve the flux form when that is warranted and nevertheless extends to non-conservative products. The method we arrive at is such that when non-conservative products are absent it reverts exactly to the method in Balsara

et al. [6] which is in the exact conservation form. Arriving at such a formulation for AFD-WENO is indeed the primary *goal* of this paper. Because the method reverts to the algorithm in Balsara et al. [6] when non-conservative products are absent, we do not display any results from conservation laws in this paper. Also note that the ability to transition to a precise flux conservation form when non-conservative products are absent ensures, via the Lax-Wendroff theorem, that shock locations will be exactly captured by the method. For many hyperbolic systems some of the components of the solution vector are in the flux conservation form while other components might be affected by non-conservative products. In such situations, the AFD-WENO algorithm in this paper is so designed that the components of the solution vector that are in the conservation form will indeed be evolved by the scheme in a fully conservative fashion. For this reason, all the examples that we will present in the later sections will consist of hyperbolic systems with at least some non-conservative products. When non-conservative products are completely absent, the AFD-WENO scheme we present here will perform identically to the AFD-WENO scheme in Balsara et al. [6]. In that paper, we show numerous examples that are drawn from several different conservation laws.

Section 2 derives the AFD-WENO for hyperbolic systems with non-conservative products. Section 3 provides further detail for evaluating the non-conservative products in a stable but nevertheless higher order fashion. Section 4 provides a pointwise plan for implementing the algorithm. Section 5 presents the accuracy analysis, showing that the method meets its design accuracy. It also provides a speed comparison of different finite difference WENO formulations. Section 6 shows one-dimensional tests. Section 7 presents multidimensional tests. Section 8 shows that the method takes well to hyperbolic systems that have non-conservative products as well as stiff source terms. Section 9 presents some conclusions.

## 2 Deriving AFD-WENO for Hyperbolic Systems with Non-conservative Products

AFD-WENO for conservation laws, along with the reconstruction strategies that support it, has been described extensively in Balsara et al. [6]. We consider that paper to be a background paper for this paper. A majority of hyperbolic systems with non-conservative products will have some components of the solution vector that are indeed in the conservation form while the remaining components of the solution vector may have non-conservative products. In this section, we wish to arrive at a scheme that retains the conservation form for the conserved components while fully accommodating the non-conservative products. In Sect. 2.1, we give some background on AFD-WENO schemes. In Sect. 2.2, we present a strategy for transitioning from a hyperbolic PDE in the conservation form to a PDE with non-conservative products. In Sect. 2.3, we show how we use this strategy to derive an update strategy for AFD-WENO that is naturally conservative for those components of the hyperbolic PDE that are in the conservation form and can simultaneously accommodate a PDE that might have some non-conservative products.

### 2.1 AFD-WENO for PDEs in Conservation Form

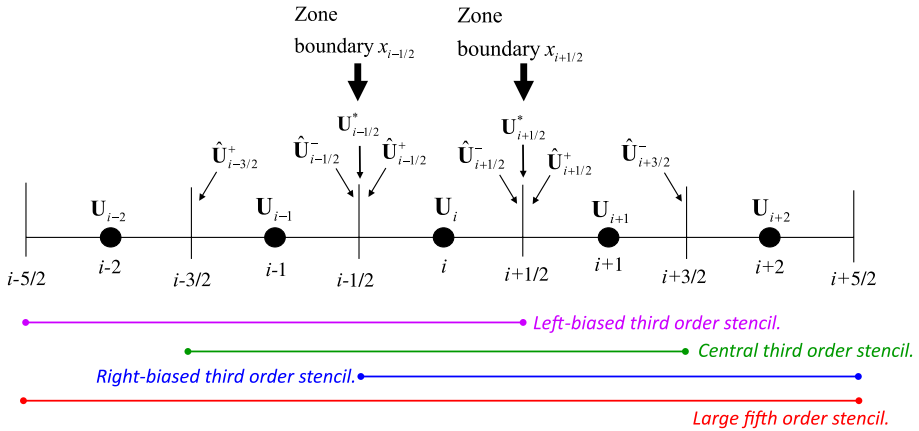
Let us start by recalling that classical FD-WENO of the type described in Jiang and Shu [30] or Balsara and Shu [11] relies on a fundamental trick that was invented in Shu and

Osher [43]. The trick consists of realizing that if one applies reconstruction to a smooth (Lipschitz continuous) flux then the finite difference scheme just reduces to a dimension-by-dimension reconstruction applied to the right- and left-going components of a numerical flux. Traditionally, the Locally Lax-Friedrichs (LLF) flux splitting is used for this. Despite the convenience and simplicity of traditional FD-WENO, it turns out that many popular Riemann solvers are not amenable to such a flux splitting. For this reason, subsequent attention has focused on another less-used idea from Shu and Osher [43] which was subsequently elaborated on by Merriman [34]. The idea consists of realizing that one can use any pointwise Riemann solver at zone boundaries as long as one can add higher order flux derivatives at zone boundaries to restore the high order pointwise accuracy at the zone centers of the overall finite difference WENO scheme. This scheme is referred to as Alternative Finite Difference WENO, or AFD-WENO (Jiang et al. [29]). In AFD-WENO one can apply any type of the Riemann solver at zone boundaries, making the method more flexible for different uses. For example, low cost Riemann solvers, like the HLLI Riemann solver of Dumbser and Balsara [19] which accurately preserves all stationary linearly degenerate discontinuities, can also be used. AFD-WENO schemes can also retain accuracy on body fitted curvilinear meshes where the preservation of the free stream condition seems to be beyond the capabilities of classical FD-WENO (Jiang et al. [31]). A corollary of AFD-WENO is that it is not based on the dimension-by-dimension finite volume *reconstruction* of the upwinded fluxes but rather on a dimension-by-dimension pointwise *interpolation* of the conserved variables. This distinction between reconstruction and interpolation is very important. In Balsara et al. [8] several WENO reconstruction formulae at different orders have been provided that are useful for classical FD-WENO. In Balsara et al. [6] analogous pointwise WENO interpolation formulae have been provided at different orders for use in AFD-WENO.

We start by writing an AFD-WENO scheme for a conservation law. We focus on the solution of a one-dimensional PDE system given by

$$\partial_t \mathbf{U} + \partial_x \mathbf{F} = 0. \quad (1)$$

Let us establish some notation. Please see Fig. 1. It shows a small section of the mesh function in a few adjacent zones. The zones are labeled by “ $i - 1, i, i + 1$ ”, etc. and their zone centers are denoted by “ $x_{i-1}, x_i, x_{i+1}$ ”, etc. The zone boundaries of each zone “ $i$ ” are denoted by “ $x_{i-1/2}$ ” and “ $x_{i+1/2}$ ” with  $\Delta x = x_{i+1/2} - x_{i-1/2}$  being a constant because we have assumed a uniform mesh. The associated mesh functions are specified in the pointwise fashion at the zone centers and are labeled by “ $\mathbf{U}_{i-1}, \mathbf{U}_i, \mathbf{U}_{i+1}$ ”, etc. Here “ $\mathbf{U}$ ” is a vector of primal variables for the hyperbolic PDE that we are considering. For the moment we consider a one-dimensional mesh, but because this is a finite difference scheme, the method can be extended dimension-by-dimension to multiple dimensions. Using our familiar WENO-AO interpolation strategy, as applied to the point values of the mesh function, we can obtain a suitably high order interpolation within each zone. (The “AO” in WENO-AO stands for adaptive order, Balsara et al. [8]. See also Balsara et al. [5, 7, 9] for additional insights.) The interpolation within zone “ $i$ ” gives us interpolated values of the mesh function,  $\hat{\mathbf{U}}_{i+1/2}^-$  and  $\hat{\mathbf{U}}_{i-1/2}^+$ , at the right and left boundaries of the zone being considered, see Fig. 1. We will use a caret to denote such interpolated variables. Note that  $\hat{\mathbf{U}}_{i+1/2}^-$  is available at the left side of the zone boundary  $x_{i+1/2}$  and  $\hat{\mathbf{U}}_{i-1/2}^+$  is available at the right side of the zone boundary  $x_{i-1/2}$ . Doing the same in zone “ $i + 1$ ”, we obtain  $\hat{\mathbf{U}}_{i+1/2}^+$  at the right side of the zone boundary  $x_{i+1/2}$ . Similarly from zone “ $i - 1$ ”, we obtain  $\hat{\mathbf{U}}_{i-1/2}^-$  at the left side of the zone boundary  $x_{i-1/2}$ . We assume that a Riemann solver



**Fig. 1** Part of the mesh around zone “ $i$ ”. The mesh functions are collocated at the zone centers, as shown by the thick dots. The zone boundaries are shown by the vertical lines. The figure also shows the stencils associated with the zone “ $i$ ” for the third- and fifth-order pointwise WENO-AO interpolation strategies. We have three smaller third-order stencils and a large fifth-order stencil. For third-order WENO-AO, only the three smaller stencils are used, whereas the larger stencil is also used for fifth-order WENO-AO. The interpolated variables at the zone boundaries are shown with a caret. The variables with a superscript star are resolved states obtained by the pointwise application of a Riemann solver at the zone boundaries. The process described here can also be adapted for Multiresolution WENO interpolation

with left and right states given by  $\hat{U}_{i+1/2}^-$  and  $\hat{U}_{i+1/2}^+$  is applied at the zone boundary  $x_{i+1/2}$  and it yields a resolved state  $U_{i+1/2}^*$  that overlies the zone boundary, as seen in Fig. 1. Because we wish to eventually work with systems that have non-conservative products, we can use the resolved state, and the structure of the Riemann fan, to obtain left- and right-going fluctuations at that zone boundary. We denote those fluctuations by  $D^{*-}(\hat{U}_{i+1/2}^-, \hat{U}_{i+1/2}^+)$  and  $D^{*+}(\hat{U}_{i+1/2}^-, \hat{U}_{i+1/2}^+)$ , respectively. (When non-conservative products are absent, these fluctuations carry an amount of upwinding-related information that is comparable to a resolved flux which we denote by  $F^*(\hat{U}_{i+1/2}^-, \hat{U}_{i+1/2}^+)$ .) Likewise, we assume that a Riemann solver with left and right states given by  $\hat{U}_{i-1/2}^-$  and  $\hat{U}_{i-1/2}^+$  is applied at the zone boundary  $x_{i-1/2}$  and it produces left- and right-going fluctuations at that zone boundary which are denoted by  $D^{*-}(\hat{U}_{i-1/2}^-, \hat{U}_{i-1/2}^+)$  and  $D^{*+}(\hat{U}_{i-1/2}^-, \hat{U}_{i-1/2}^+)$ , respectively. (When non-conservative products are absent, these fluctuations carry an amount of upwinding-related information that is comparable to a resolved flux which we denote by  $F^*(\hat{U}_{i-1/2}^-, \hat{U}_{i-1/2}^+)$ .) One of the advantages of the AFD-WENO algorithm is that it is agnostic to the type of Riemann solver that is used. This completes our description of Fig. 1.

Let us start from a simpler starting point by considering a hyperbolic conservation law, see Eq. (1). The discrete in space but continuous in time update equation for the AFD-WENO scheme can then be written as

$$\partial_t U_i = -\frac{1}{\Delta x} \left\{ \mathbf{F}^* \left( \hat{\mathbf{U}}_{i+1/2}^-, \hat{\mathbf{U}}_{i+1/2}^+ \right) - \mathbf{F}^* \left( \hat{\mathbf{U}}_{i-1/2}^-, \hat{\mathbf{U}}_{i-1/2}^+ \right) \right\} \\ - \frac{1}{\Delta x} \left\{ \begin{aligned} & \left[ -\frac{1}{24} (\Delta x)^2 \left[ \partial_x^2 \mathbf{F} \right]_{i+1/2} + \frac{7}{5760} (\Delta x)^4 \left[ \partial_x^4 \mathbf{F} \right]_{i+1/2} \right. \\ & \left. - \frac{31}{967680} (\Delta x)^6 \left[ \partial_x^6 \mathbf{F} \right]_{i+1/2} + \frac{127}{154828800} (\Delta x)^8 \left[ \partial_x^8 \mathbf{F} \right]_{i+1/2} \right] \\ & - \left[ -\frac{1}{24} (\Delta x)^2 \left[ \partial_x^2 \mathbf{F} \right]_{i-1/2} + \frac{7}{5760} (\Delta x)^4 \left[ \partial_x^4 \mathbf{F} \right]_{i-1/2} \right. \\ & \left. - \frac{31}{967680} (\Delta x)^6 \left[ \partial_x^6 \mathbf{F} \right]_{i-1/2} + \frac{127}{154828800} (\Delta x)^8 \left[ \partial_x^8 \mathbf{F} \right]_{i-1/2} \right] \end{aligned} \right\}. \quad (2)$$

The above equation was originally set down in Shu and Osher [43] but those who want a simpler computer algebra system-based derivation of it with more explanations can also see Balsara et al. [6]. The first curly bracket of Eq. (2) would yield a scheme whose accuracy is restricted to second order. The higher order flux derivatives are needed for raising the accuracy of the update equation to its design accuracy. The black flux derivative terms ( $[\partial_x^2 \mathbf{F}]_{i+1/2}$  and  $[\partial_x^2 \mathbf{F}]_{i-1/2}$ ) in the above equation yield a third-order scheme. In that case, these second derivatives have to be third-order accurate. If the red flux derivative terms ( $[\partial_x^4 \mathbf{F}]_{i+1/2}$  and  $[\partial_x^4 \mathbf{F}]_{i-1/2}$ ) are also included, in addition to the black terms, the scheme becomes fifth-order accurate. In that case, all the applicable derivatives have to be fifth-order accurate. If the blue flux derivative terms ( $[\partial_x^6 \mathbf{F}]_{i+1/2}$  and  $[\partial_x^6 \mathbf{F}]_{i-1/2}$ ) are also included, in addition to the black and red terms, we get a seventh-order scheme. In that case, all the applicable derivatives have to be seventh-order accurate. If the magenta flux derivative terms ( $[\partial_x^8 \mathbf{F}]_{i+1/2}$  and  $[\partial_x^8 \mathbf{F}]_{i-1/2}$ ) are included, in addition to the black, red, and blue terms, we get a ninth-order scheme. In that case, all the applicable derivatives have to be ninth-order accurate. We will retain this coloring scheme in all subsequent equations that describe the update equation for this scheme. Equation (2) is to be used to derive an AFD-WENO scheme that can include hyperbolic systems with non-conservative products. At this stage of the discussion in this paper, we do not specify how the higher derivatives of the flux are to be obtained. For that reason, we keep the notation for the higher derivatives of the flux somewhat relaxed. We will tighten up the notation once we get to the part where the algorithmic aspects are to be discussed.

Please observe that Eq. (2) is in the flux form and should, therefore, be able to capture shocks accurately. The first line in Eq. (2) comes from applying Riemann solvers at zone boundaries. These Riemann solvers contain the stabilization, i.e., the numerical dissipation, that is needed for handling discontinuous solutions. However, if only the first line in Eq. (2) is used, the scheme would be restricted to the second order of accuracy. The subsequent higher order derivatives of the flux in Eq. (2) are indeed *essential* for restoring higher order accuracy to Eq. (2). It should also be noted that the higher order derivatives can cause Gibbs oscillation and should only be used when the solution is smooth. We see, therefore, that the higher order derivatives of the flux in Eq. (2) constitute a two-edged sword. When the solution is smooth, they are needed for higher order accuracy. But, when the solution is non-smooth, their suppression is needed for the numerical stabilization of the AFD-WENO scheme. In Balsara et al. [6] a variant of WENO interpolation was presented which can take the zone-centered point values of the fluxes as input and return non-linearly hybridized values of the higher derivatives of the fluxes at the zone boundaries. It was found that this interpolation was very useful in retaining the high order accuracy when

it is justified but nevertheless avoiding spurious numerical oscillations from developing in the vicinity of non-smooth solutions. This completes our broad-brush description of the AFD-WENO scheme for conservation laws.

## 2.2 Insight on Transitioning to Hyperbolic PDEs with Non-conservative Products

Let us begin with the same conservation law as in Eq. (1) but write it as

$$\partial_t \mathbf{U} + \partial_x \mathbf{F} = 0 \quad \text{with the flux splitting} \quad \mathbf{F}(\mathbf{U}) = \mathbf{F}_C(\mathbf{U}) + \mathbf{F}_{NC}(\mathbf{U}). \quad (3)$$

The subscripts “C” and “NC” indicate that in the upcoming discussion we will treat the flux  $\mathbf{F}_C(\mathbf{U})$  in the conservation form and we will write  $\mathbf{F}_{NC}(\mathbf{U})$  as if it is only available as a non-conservative product. We can now write the characteristic matrix as

$$\begin{aligned} \mathbf{A}(\mathbf{U}) &= \frac{\partial (\mathbf{F}_C(\mathbf{U}) + \mathbf{F}_{NC}(\mathbf{U}))}{\partial \mathbf{U}} = \mathbf{B}(\mathbf{U}) + \mathbf{C}(\mathbf{U}) \quad \text{with} \\ \mathbf{B}(\mathbf{U}) &\equiv \frac{\partial \mathbf{F}_C(\mathbf{U})}{\partial \mathbf{U}} \quad \text{and} \quad \mathbf{C}(\mathbf{U}) \equiv \frac{\partial \mathbf{F}_{NC}(\mathbf{U})}{\partial \mathbf{U}}. \end{aligned} \quad (4)$$

As a result, the hyperbolic PDE system in Eq. (3) can be written in any of three equivalent forms:

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) = 0 \Leftrightarrow \partial_t \mathbf{U} + \mathbf{A}(\mathbf{U}) \partial_x \mathbf{U} = 0 \Leftrightarrow \partial_t \mathbf{U} + \partial_x \mathbf{F}_C(\mathbf{U}) + \mathbf{C}(\mathbf{U}) \partial_x \mathbf{U} = 0. \quad (5)$$

It is the last form of Eq. (5) that gives us the essential insight because it looks exactly like a general hyperbolic PDE system that has some non-conservative products, which we write as

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) + \mathbf{C}(\mathbf{U}) \partial_x \mathbf{U} = 0. \quad (6)$$

Compared to the last Eq. (5), we have obtained Eq. (6) by just erasing the “C” subscript for the flux. This is going to be our strategy for starting with Eq. (2) and recasting it in a form that accommodates non-conservative products.

## 2.3 Deriving an AFD-WENO Scheme with a Conservative Limit and also Accommodation for Non-conservative Products

There is a good physics-based reason for wanting to write the final numerical update of Eq. (6) in the fluctuation form. It stems from the fact that the matrix “A” does have real eigenvectors and a complete set of eigenvalues; but it cannot be guaranteed that the matrices “B” and “C” have real eigenvalues or a complete set of eigenvectors. As a result, when non-conservative products are present, the fluctuation form is the only way to go. Many Riemann solvers for conservation laws have been formulated so that they can be written in the flux form or an entirely equivalent fluctuation form that respects conservation. Many of those Riemann solvers have been extended so that they can provide a fluctuation form even when non-conservative products are present.

When the hyperbolic PDE is in the conservation form, it can be written in an entirely equivalent fluctuation form. However, the previous paragraph shows us that once we have obtained a fluctuation form, we can use it also for a hyperbolic system with non-conservative products. To that end, let us write the fluctuation form for the left-going fluctuations at the zone boundary “ $i + 1/2$ ” as

$$\begin{aligned} \mathbf{D}^{*-}(\hat{\mathbf{U}}_{i+1/2}^-, \hat{\mathbf{U}}_{i+1/2}^+) &= \mathbf{F}^*(\hat{\mathbf{U}}_{i+1/2}^-, \hat{\mathbf{U}}_{i+1/2}^+) - \mathbf{F}(\hat{\mathbf{U}}_{i+1/2}^-) \\ \Rightarrow \mathbf{F}^*(\hat{\mathbf{U}}_{i+1/2}^-, \hat{\mathbf{U}}_{i+1/2}^+) &= \mathbf{D}^{*-}(\hat{\mathbf{U}}_{i+1/2}^-, \hat{\mathbf{U}}_{i+1/2}^+) + \mathbf{F}_C(\hat{\mathbf{U}}_{i+1/2}^-) + \mathbf{F}_{NC}(\hat{\mathbf{U}}_{i+1/2}^-). \end{aligned} \quad (7)$$

We can also write the fluctuation form for the right-going fluctuations at the zone boundary “ $i - 1/2$ ” as

$$\begin{aligned} \mathbf{D}^{*+}(\hat{\mathbf{U}}_{i-1/2}^-, \hat{\mathbf{U}}_{i-1/2}^+) &= \mathbf{F}(\hat{\mathbf{U}}_{i-1/2}^+) - \mathbf{F}^*(\hat{\mathbf{U}}_{i-1/2}^-, \hat{\mathbf{U}}_{i-1/2}^+) \\ \Rightarrow \mathbf{F}^*(\hat{\mathbf{U}}_{i-1/2}^-, \hat{\mathbf{U}}_{i-1/2}^+) &= -\mathbf{D}^{*+}(\hat{\mathbf{U}}_{i-1/2}^-, \hat{\mathbf{U}}_{i-1/2}^+) + \mathbf{F}_C(\hat{\mathbf{U}}_{i-1/2}^+) + \mathbf{F}_{NC}(\hat{\mathbf{U}}_{i-1/2}^+). \end{aligned} \quad (8)$$

We should recall  $\mathbf{F}(\mathbf{U}) = \mathbf{F}_C(\mathbf{U}) + \mathbf{F}_{NC}(\mathbf{U})$  and its usage in the above two equations. To arrive at our derivation, let us simply replace the resolved flux terms (the ones with the star superscripts) in the first curly bracket Eq. (2) with the fluctuation terms from Eqs. (7) and (8). From Eq. (2), we get

$$\begin{aligned} \partial_t \mathbf{U}_i &= -\frac{1}{\Delta x} \left\{ \mathbf{D}^{*-}(\hat{\mathbf{U}}_{i+1/2}^-, \hat{\mathbf{U}}_{i+1/2}^+) + \mathbf{D}^{*+}(\hat{\mathbf{U}}_{i-1/2}^-, \hat{\mathbf{U}}_{i-1/2}^+) \right\} \\ &\quad - \frac{1}{\Delta x} \left\{ \mathbf{F}_C(\hat{\mathbf{U}}_{i+1/2}^-) - \mathbf{F}_C(\hat{\mathbf{U}}_{i-1/2}^+) \right\} - \frac{1}{\Delta x} \left\{ \mathbf{F}_{NC}(\hat{\mathbf{U}}_{i+1/2}^-) - \mathbf{F}_{NC}(\hat{\mathbf{U}}_{i-1/2}^+) \right\} \\ &\quad - \frac{1}{\Delta x} \left\{ \left[ -\frac{1}{24}(\Delta x)^2 [\partial_x^2 \mathbf{F}]_{i+1/2} + \frac{7}{5760}(\Delta x)^4 [\partial_x^4 \mathbf{F}]_{i+1/2} \right. \right. \\ &\quad \left. \left. - \frac{31}{967680}(\Delta x)^6 [\partial_x^6 \mathbf{F}]_{i+1/2} + \frac{127}{154828800}(\Delta x)^8 [\partial_x^8 \mathbf{F}]_{i+1/2} \right] \right. \\ &\quad \left. - \left[ -\frac{1}{24}(\Delta x)^2 [\partial_x^2 \mathbf{F}]_{i-1/2} + \frac{7}{5760}(\Delta x)^4 [\partial_x^4 \mathbf{F}]_{i-1/2} \right. \right. \\ &\quad \left. \left. - \frac{31}{967680}(\Delta x)^6 [\partial_x^6 \mathbf{F}]_{i-1/2} + \frac{127}{154828800}(\Delta x)^8 [\partial_x^8 \mathbf{F}]_{i-1/2} \right] \right\}. \end{aligned} \quad (9)$$

Now we realize that the  $-\left\{ \mathbf{F}_C(\hat{\mathbf{U}}_{i+1/2}^-) - \mathbf{F}_C(\hat{\mathbf{U}}_{i-1/2}^+) \right\} / \Delta x$  term, along with the fluctuation term  $-\left\{ \mathbf{D}^{*-}(\hat{\mathbf{U}}_{i+1/2}^-, \hat{\mathbf{U}}_{i+1/2}^+) + \mathbf{D}^{*+}(\hat{\mathbf{U}}_{i-1/2}^-, \hat{\mathbf{U}}_{i-1/2}^+) \right\} / \Delta x$ , in the above equation will give us a very nice conservation property for those components of the hyperbolic PDE that are genuinely in the conservation form. For that reason, we leave those terms as they are. However, in our current philosophy, the term  $-\left\{ \mathbf{F}_{NC}(\hat{\mathbf{U}}_{i+1/2}^-) - \mathbf{F}_{NC}(\hat{\mathbf{U}}_{i-1/2}^+) \right\} / \Delta x$  is just a placeholder for a non-conservative product and it will eventually have to be represented in terms of the matrix of non-conservative products “ $\mathbf{C}$ ” in Eq. (9). To that end, we perform a Taylor series expansion for the placeholder term and write it as follows:

$$\begin{aligned}
 & -\frac{1}{\Delta x} \left\{ \mathbf{F}_{\text{NC}} \left( \hat{\mathbf{U}}_{i+1/2}^- \right) - \mathbf{F}_{\text{NC}} \left( \hat{\mathbf{U}}_{i-1/2}^+ \right) \right\} \\
 & \cong -(\partial_x \mathbf{F}_{\text{NC}})_i - \left\{ + \frac{\Delta x^2}{24} (\partial_x^3 \mathbf{F}_{\text{NC}})_i + \frac{\Delta x^4}{1920} (\partial_x^5 \mathbf{F}_{\text{NC}})_i \right. \\
 & \quad \left. + \frac{\Delta x^6}{322560} (\partial_x^7 \mathbf{F}_{\text{NC}})_i + \frac{\Delta x^8}{92897280} (\partial_x^9 \mathbf{F}_{\text{NC}})_i \right\}.
 \end{aligned} \quad (10)$$

The above equation can also be written in terms of the higher order derivatives of  $\mathbf{F}_{\text{NC}}$  that are evaluated at the zone boundaries. This is useful because it will enable us to make a simplification in Eq. (9). We, therefore, write the above equation as

$$\begin{aligned}
 & -\frac{1}{\Delta x} \left\{ \mathbf{F}_{\text{NC}} \left( \hat{\mathbf{U}}_{i+1/2}^- \right) - \mathbf{F}_{\text{NC}} \left( \hat{\mathbf{U}}_{i-1/2}^+ \right) \right\} \\
 & \cong -\mathbf{C}(\mathbf{U}_i) (\partial_x \hat{\mathbf{U}})_i - \frac{1}{\Delta x} \left\{ \left[ \begin{aligned} & \frac{1}{24} (\Delta x)^2 [\partial_x^2 \mathbf{F}_{\text{NC}}]_{i+1/2} - \frac{7}{5760} (\Delta x)^4 [\partial_x^4 \mathbf{F}_{\text{NC}}]_{i+1/2} \\ & + \frac{31}{967680} (\Delta x)^6 [\partial_x^6 \mathbf{F}_{\text{NC}}]_{i+1/2} - \frac{127}{154828800} (\Delta x)^8 [\partial_x^8 \mathbf{F}_{\text{NC}}]_{i+1/2} \end{aligned} \right] \right. \\
 & \quad \left. - \left[ \begin{aligned} & \frac{1}{24} (\Delta x)^2 [\partial_x^2 \mathbf{F}_{\text{NC}}]_{i-1/2} - \frac{7}{5760} (\Delta x)^4 [\partial_x^4 \mathbf{F}_{\text{NC}}]_{i-1/2} \\ & + \frac{31}{967680} (\Delta x)^6 [\partial_x^6 \mathbf{F}_{\text{NC}}]_{i-1/2} - \frac{127}{154828800} (\Delta x)^8 [\partial_x^8 \mathbf{F}_{\text{NC}}]_{i-1/2} \end{aligned} \right] \right\}.
 \end{aligned} \quad (11)$$

In the above two equations we use the “ $\cong$ ” sign instead of the “=” sign because the two sides of the equation are not exactly identical; but they are only identical to the level of discretization error. This also shows us that this is the step where the exact conservation can be lost. However, by localizing this loss of conservation to those components of the solution vector that have non-zero values for  $(\mathbf{C} \partial_x \hat{\mathbf{U}})$ , we see that the components of the solution vector that are in the conservation form will indeed retain the exact flux conservation. This is a very important property because it ensures that when the exact conservation is guaranteed by the PDE, our AFD-WENO scheme will indeed retain the exact flux conservation in a numerical simulation up to machine precision.

Now by amalgamating Eqs. (9) and (11), we get

$$\begin{aligned}
 \partial_t \mathbf{U}_i &= -\frac{1}{\Delta x} \left\{ \mathbf{D}^{*-} \left( \hat{\mathbf{U}}_{i+1/2}^-, \hat{\mathbf{U}}_{i+1/2}^+ \right) + \mathbf{D}^{*+} \left( \hat{\mathbf{U}}_{i-1/2}^-, \hat{\mathbf{U}}_{i-1/2}^+ \right) \right\} - \frac{1}{\Delta x} \left\{ \mathbf{F}_{\text{C}} \left( \hat{\mathbf{U}}_{i+1/2}^- \right) - \mathbf{F}_{\text{C}} \left( \hat{\mathbf{U}}_{i-1/2}^+ \right) \right\} - \mathbf{C}(\mathbf{U}_i) (\partial_x \hat{\mathbf{U}})_i \\
 & - \frac{1}{\Delta x} \left\{ \left[ \begin{aligned} & -\frac{1}{24} (\Delta x)^2 [\partial_x^2 \mathbf{F}_{\text{C}}]_{i+1/2} + \frac{7}{5760} (\Delta x)^4 [\partial_x^4 \mathbf{F}_{\text{C}}]_{i+1/2} \\ & - \frac{31}{967680} (\Delta x)^6 [\partial_x^6 \mathbf{F}_{\text{C}}]_{i+1/2} + \frac{127}{154828800} (\Delta x)^8 [\partial_x^8 \mathbf{F}_{\text{C}}]_{i+1/2} \end{aligned} \right] \right. \\
 & \quad \left. - \left[ \begin{aligned} & -\frac{1}{24} (\Delta x)^2 [\partial_x^2 \mathbf{F}_{\text{C}}]_{i-1/2} + \frac{7}{5760} (\Delta x)^4 [\partial_x^4 \mathbf{F}_{\text{C}}]_{i-1/2} \\ & - \frac{31}{967680} (\Delta x)^6 [\partial_x^6 \mathbf{F}_{\text{C}}]_{i-1/2} + \frac{127}{154828800} (\Delta x)^8 [\partial_x^8 \mathbf{F}_{\text{C}}]_{i-1/2} \end{aligned} \right] \right\}.
 \end{aligned} \quad (12)$$

The above equation still pertains to Eq. (5). We now realize that to transition from Eqs. (5)–(6) we only need to make the transcription  $\mathbf{F}_{\text{C}} \rightarrow \mathbf{F}$ . Therefore, when working with Eq. (6), we have

$$\partial_i \mathbf{U}_i = -\frac{1}{\Delta x} \left\{ \mathbf{D}^{*-}(\hat{\mathbf{U}}_{i+1/2}^-, \hat{\mathbf{U}}_{i+1/2}^+) + \mathbf{D}^{*+}(\hat{\mathbf{U}}_{i-1/2}^-, \hat{\mathbf{U}}_{i-1/2}^+) \right\} - \frac{1}{\Delta x} \left\{ \mathbf{F}(\hat{\mathbf{U}}_{i+1/2}^-) - \mathbf{F}(\hat{\mathbf{U}}_{i-1/2}^+) \right\} - \mathbf{C}(\mathbf{U}_i) \left[ \partial_x \hat{\mathbf{U}} \right]_i$$

$$- \frac{1}{\Delta x} \left\{ \begin{aligned} & \left[ -\frac{1}{24}(\Delta x)^2 \left[ \partial_x^2 \hat{\mathbf{F}} \right]_{i+1/2} + \frac{7}{5760}(\Delta x)^4 \left[ \partial_x^4 \hat{\mathbf{F}} \right]_{i+1/2} \right. \\ & \left. - \frac{31}{967680}(\Delta x)^6 \left[ \partial_x^6 \hat{\mathbf{F}} \right]_{i+1/2} + \frac{127}{154828800}(\Delta x)^8 \left[ \partial_x^8 \hat{\mathbf{F}} \right]_{i+1/2} \right] \\ & - \left[ -\frac{1}{24}(\Delta x)^2 \left[ \partial_x^2 \hat{\mathbf{F}} \right]_{i-1/2} + \frac{7}{5760}(\Delta x)^4 \left[ \partial_x^4 \hat{\mathbf{F}} \right]_{i-1/2} \right. \\ & \left. - \frac{31}{967680}(\Delta x)^6 \left[ \partial_x^6 \hat{\mathbf{F}} \right]_{i-1/2} + \frac{127}{154828800}(\Delta x)^8 \left[ \partial_x^8 \hat{\mathbf{F}} \right]_{i-1/2} \right] \end{aligned} \right\}. \quad (13)$$

Till Eq. (12) we had been willing to accept a relaxed notation for the higher derivatives of the fluxes at the zone boundaries because we were just engaged in a derivation. In Eq. (13) and onwards, we have tightened the notation. Therefore, the caret in  $\left[ \partial_x^2 \hat{\mathbf{F}} \right]_{i+1/2}$ , and terms like it, indicates that is term is to be obtained via some form of the WENO interpolation and the interpolant is to be evaluated at the zone boundary “ $i+1/2$ ”. The exact form of the interpolation will be discussed in the next section. The above equation is potentially very useful in several situations where there is a weak coupling between the conservative terms and the non-conservative products. For example, it could be very useful for large combinations of PDE systems like general relativistic hydrodynamics where there is a clear split between the Einstein field equations and the equations of relativistic hydrodynamics. However, for tightly coupled systems, i.e., systems where the flux terms and non-conservative terms interact strongly with one another, it still has a small deficiency which we explain below.

Please look at Eq. (13) and notice the derivative  $\left[ \partial_x \hat{\mathbf{U}} \right]_i$  in the term  $-\mathbf{C}(\mathbf{U}_i) \left[ \partial_x \hat{\mathbf{U}} \right]_i$ . That derivative will be evaluated using the WENO interpolation. However, the non-linear stabilization in WENO is applied to the interpolant, not to its derivative. Therefore, it is slightly advantageous to write Eq. (13) in an equivalent form given by

$$\partial_i \mathbf{U}_i = -\frac{1}{\Delta x} \left\{ \mathbf{D}^{*-}(\hat{\mathbf{U}}_{i+1/2}^-, \hat{\mathbf{U}}_{i+1/2}^+) + \mathbf{D}^{*+}(\hat{\mathbf{U}}_{i-1/2}^-, \hat{\mathbf{U}}_{i-1/2}^+) \right\} - \frac{1}{\Delta x} \left\{ \mathbf{F}(\hat{\mathbf{U}}_{i+1/2}^-) - \mathbf{F}(\hat{\mathbf{U}}_{i-1/2}^+) \right\}$$

$$- \frac{1}{\Delta x} \mathbf{C}(\mathbf{U}_i) (\hat{\mathbf{U}}_{i+1/2}^- - \hat{\mathbf{U}}_{i-1/2}^+) + \frac{1}{\Delta x} \mathbf{C}(\mathbf{U}_i) \left\{ \begin{aligned} & \frac{1}{24}(\Delta x)^3 \left[ \partial_x^3 \hat{\mathbf{U}} \right]_i + \frac{1}{1920}(\Delta x)^5 \left[ \partial_x^5 \hat{\mathbf{U}} \right]_i \\ & + \frac{1}{322560}(\Delta x)^7 \left[ \partial_x^7 \hat{\mathbf{U}} \right]_i + \frac{1}{92897280}(\Delta x)^9 \left[ \partial_x^9 \hat{\mathbf{U}} \right]_i \end{aligned} \right\}$$

$$- \frac{1}{\Delta x} \left\{ \begin{aligned} & \left[ -\frac{1}{24}(\Delta x)^2 \left[ \partial_x^2 \hat{\mathbf{F}} \right]_{i+1/2} + \frac{7}{5760}(\Delta x)^4 \left[ \partial_x^4 \hat{\mathbf{F}} \right]_{i+1/2} \right. \\ & \left. - \frac{31}{967680}(\Delta x)^6 \left[ \partial_x^6 \hat{\mathbf{F}} \right]_{i+1/2} + \frac{127}{154828800}(\Delta x)^8 \left[ \partial_x^8 \hat{\mathbf{F}} \right]_{i+1/2} \right] \\ & - \left[ -\frac{1}{24}(\Delta x)^2 \left[ \partial_x^2 \hat{\mathbf{F}} \right]_{i-1/2} + \frac{7}{5760}(\Delta x)^4 \left[ \partial_x^4 \hat{\mathbf{F}} \right]_{i-1/2} \right. \\ & \left. - \frac{31}{967680}(\Delta x)^6 \left[ \partial_x^6 \hat{\mathbf{F}} \right]_{i-1/2} + \frac{127}{154828800}(\Delta x)^8 \left[ \partial_x^8 \hat{\mathbf{F}} \right]_{i-1/2} \right] \end{aligned} \right\}. \quad (14)$$

Notice from the above equation that the terms  $\hat{\mathbf{U}}_{i+1/2}^-$  and  $\hat{\mathbf{U}}_{i-1/2}^+$  in  $-\mathbf{C}(\mathbf{U}_i) (\hat{\mathbf{U}}_{i+1/2}^- - \hat{\mathbf{U}}_{i-1/2}^+) / \Delta x$  can now be obtained by a good non-linearly hybridized WENO interpolation process. Having seen that the flux terms can be written as the finite difference approximation of suitable

combinations of higher order derivatives at zone boundaries in the above equation, we want to use the same idea for the higher order derivatives given by  $[\partial_x^3 \hat{\mathbf{U}}]_i$ ,  $[\partial_x^5 \hat{\mathbf{U}}]_i$ ,  $[\partial_x^7 \hat{\mathbf{U}}]_i$ , and  $[\partial_x^9 \hat{\mathbf{U}}]_i$  in Eq. (14). This is a reasonable thing to do because we have the intuition that a finite difference approximation is numerically more stable than the evaluation of higher derivatives at zone centers. For that reason, we rewrite Eq. (14) as

$$\begin{aligned} \partial_t \mathbf{U}_i = & -\frac{1}{\Delta x} \left\{ \mathbf{D}_{i+1/2}^* \left( \hat{\mathbf{U}}_{i+1/2}^-, \hat{\mathbf{U}}_{i+1/2}^+ \right) + \mathbf{D}_{i-1/2}^{*+} \left( \hat{\mathbf{U}}_{i-1/2}^-, \hat{\mathbf{U}}_{i-1/2}^+ \right) \right\} - \frac{1}{\Delta x} \left\{ \mathbf{F} \left( \hat{\mathbf{U}}_{i+1/2}^- \right) - \mathbf{F} \left( \hat{\mathbf{U}}_{i-1/2}^+ \right) \right\} \\ & - \frac{1}{\Delta x} \mathbf{C}(\mathbf{U}_i) \left( \hat{\mathbf{U}}_{i+1/2}^- - \hat{\mathbf{U}}_{i-1/2}^+ \right) \\ & - \frac{1}{\Delta x} \mathbf{C}(\mathbf{U}_i) \left\{ \begin{aligned} & \left[ -\frac{1}{24} (\Delta x)^2 [\partial_x^2 \hat{\mathbf{U}}]_{i+1/2} + \frac{7}{5760} (\Delta x)^4 [\partial_x^4 \hat{\mathbf{U}}]_{i+1/2} \right. \\ & \left. - \frac{31}{967680} (\Delta x)^6 [\partial_x^6 \hat{\mathbf{U}}]_{i+1/2} + \frac{127}{154828800} (\Delta x)^8 [\partial_x^8 \hat{\mathbf{U}}]_{i+1/2} \right] \\ & \left[ -\frac{1}{24} (\Delta x)^2 [\partial_x^2 \hat{\mathbf{U}}]_{i-1/2} + \frac{7}{5760} (\Delta x)^4 [\partial_x^4 \hat{\mathbf{U}}]_{i-1/2} \right. \\ & \left. - \frac{31}{967680} (\Delta x)^6 [\partial_x^6 \hat{\mathbf{U}}]_{i-1/2} + \frac{127}{154828800} (\Delta x)^8 [\partial_x^8 \hat{\mathbf{U}}]_{i-1/2} \right] \end{aligned} \right\} \\ & - \frac{1}{\Delta x} \left\{ \begin{aligned} & \left[ -\frac{1}{24} (\Delta x)^2 [\partial_x^2 \hat{\mathbf{F}}]_{i+1/2} + \frac{7}{5760} (\Delta x)^4 [\partial_x^4 \hat{\mathbf{F}}]_{i+1/2} \right. \\ & \left. - \frac{31}{967680} (\Delta x)^6 [\partial_x^6 \hat{\mathbf{F}}]_{i+1/2} + \frac{127}{154828800} (\Delta x)^8 [\partial_x^8 \hat{\mathbf{F}}]_{i+1/2} \right] \\ & \left[ -\frac{1}{24} (\Delta x)^2 [\partial_x^2 \hat{\mathbf{F}}]_{i-1/2} + \frac{7}{5760} (\Delta x)^4 [\partial_x^4 \hat{\mathbf{F}}]_{i-1/2} \right. \\ & \left. - \frac{31}{967680} (\Delta x)^6 [\partial_x^6 \hat{\mathbf{F}}]_{i-1/2} + \frac{127}{154828800} (\Delta x)^8 [\partial_x^8 \hat{\mathbf{F}}]_{i-1/2} \right] \end{aligned} \right\}. \end{aligned} \quad (15)$$

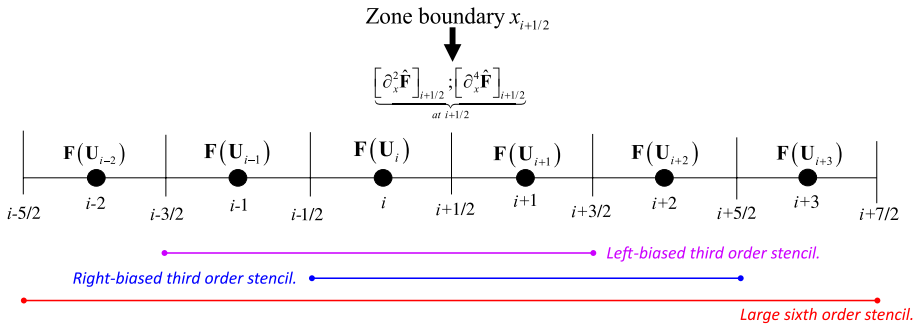
The color coding in Eqs. (9)–(15), as it pertains to order of accuracy, is identical to the color coding in Eq. (2). Equation (15) is the AFD-WENO update equation that we are seeking. It contains all the update terms that will be needed in AFD-WENO schemes that are up to ninth-order accurate. To obtain a third-order AFD-WENO scheme from Eq. (15), please retain only the black terms and eliminate the red, blue, and magenta terms. To obtain a fifth-order AFD-WENO scheme from Eq. (15), please retain only the black, and red terms and eliminate the blue and magenta terms. To obtain a seventh-order AFD-WENO scheme from Eq. (15), please retain only the black, red, and blue terms and eliminate the magenta terms. To obtain a ninth-order AFD-WENO scheme from Eq. (15), please retain all the terms in that equation. The computer algebra system script in “Appendix A” of Balsara et al. [6] can be extended to give even higher orders. In “Appendix A” of this paper, we provide an alternative to Eq. (15) which could very slightly reduce the cost of evaluating the last curly bracket in Eq. (15).

Equation (15) has such a nice structure that when the matrix of non-conservative products is zero, i.e., when we have “ $\mathbf{C} = 0$ ”, then Eq. (15) reduces identically to Eq. (2) which is manifestly the flux conservation form. When only a few rows of the matrix “ $\mathbf{C}$ ” are non-zero, Eq. (15) retains a nice structure which ensures that the remaining rows of the hyperbolic PDE will indeed remain in the flux conservation form. Therefore, Eq. (15) retains the flux conservation form when such a conservation form is present in the governing PDE, and Eq. (15) nevertheless permits the incorporation of non-conservative products

when the PDE has such products. Furthermore, Eq. (15) is in the finite difference form, which makes it very efficient for high-order accurate computations in multiple dimensions. Also notice that Eq. (15) can be used with any Riemann solver that can be cast in the fluctuation form. Since many of the well-known Riemann solvers have a flux form as well as a fluctuation form, they can all be used in Eq. (15). Different Riemann solvers may have some special properties that make them useful in certain fields of study. This makes Eq. (15) broadly applicable to many PDEs in many different fields of study.

To arrive at a non-linearly stabilized version of Eq. (15), we will need three WENO interpolation steps. Notice that the first WENO interpolation is needed to obtain the non-linearly stabilized and interpolated solution at various locations on the mesh. This will give us the first line (i.e., the first three terms) of Eq. (15), which indeed retains up to second order of accuracy. It is very desirable that this interpolation be carried out in the eigenspace of the governing equation. The first three terms of Eq. (15), if they are used by themselves, also guarantee that we will get a very high quality second order scheme if a high quality WENO interpolation is used. The last two curly brackets of Eq. (15) only contain higher order derivatives that are needed for raising the accuracy of the AFD-WENO scheme to the desired accuracy when the solution is smooth. But those higher order derivatives are also a two-edged sword because they can introduce spurious oscillations, thereby damaging the accuracy, when the solution is non-smooth. We seek to stabilize them using a different style of the WENO interpolation from Balsara et al. [6]. The second WENO interpolation is needed for obtaining the non-linearly stabilized higher order derivatives,  $\left[\partial_x^2 \hat{\mathbf{F}}\right]_{i+1/2}$ ,  $\left[\partial_x^4 \hat{\mathbf{F}}\right]_{i+1/2}$ ,  $\left[\partial_x^6 \hat{\mathbf{F}}\right]_{i+1/2}$ , and  $\left[\partial_x^8 \hat{\mathbf{F}}\right]_{i+1/2}$  (as needed) at the zone boundary “ $i+1/2$ ”; with analogous evaluations needed at the zone boundary “ $i-1/2$ ”. The third WENO interpolation is needed for obtaining the non-linearly stabilized higher order derivatives,  $\left[\partial_x^2 \hat{\mathbf{U}}\right]_{i+1/2}$ ,  $\left[\partial_x^4 \hat{\mathbf{U}}\right]_{i+1/2}$ ,  $\left[\partial_x^6 \hat{\mathbf{U}}\right]_{i+1/2}$ , and  $\left[\partial_x^8 \hat{\mathbf{U}}\right]_{i+1/2}$  (as needed) at the zone boundaries. The second and third WENO interpolations can be carried out in the physical space, so that their cost can be minimized. It is also useful to emphasize that, Eq. (15) is not the only formulation that is possible. In some circumstances, and for some PDEs where the non-conservative products are not strongly non-linear, Eq. (13) can yield a simpler formulation that requires only two WENO interpolation steps.

Our experience has been that the WENO interpolation is sufficient for controlling the higher order derivative terms in Eq. (15). The WENO interpolation represents a *numerics-based modulation* of the solution and applies generally to all PDEs. But for a very small number of very stringent hyperbolic systems with non-conservative products a *physics-based modulation* of the interpolation requires the use of a flattener algorithm (Colella and Woodward [17], Balsara [3, 4]). These physics-based flattener algorithms are PDE-specific and they are designed to do nothing when the solution is smooth or only mildly non-smooth, and activate themselves only when the solution is very non-smooth. So, the higher derivative terms in Eq. (15) have to be used with care.

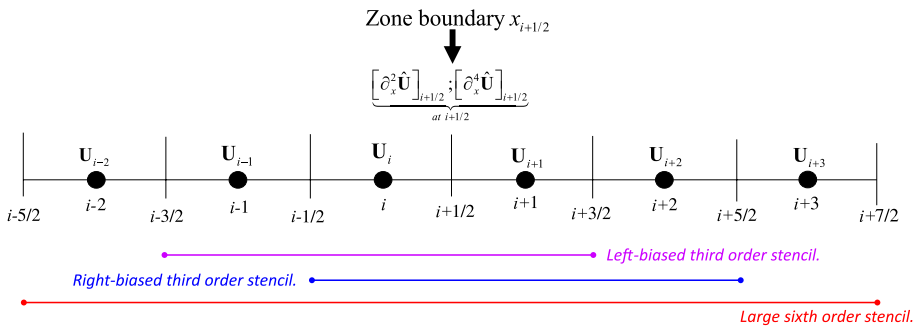


**Fig. 2** Part of the mesh around the zone boundary “ $i + 1/2$ ”. The fluxes are evaluated pointwise at the zone centers, as shown by the thick dots. The zone boundaries are shown by the vertical lines. The figure also shows the stencils associated with the zone boundary “ $i + 1/2$ ” for the third- and fifth-order AFD-WENO schemes. We have two smaller third-order stencils and a large sixth-order stencil. For a third-order AFD-WENO scheme, the two smaller stencils can be non-linearly hybridized. In that case, the second derivatives of the flux can be obtained at the zone boundary when the smoothness in the solution warrants it. For the fifth-order AFD-WENO, the two smaller stencils can be non-linearly hybridized along with the larger stencil. In that case, the second and fourth derivatives of the flux can be obtained at the zone boundary when the smoothness in the solution warrants it. The process described here can be done for Adaptive Order and Multiresolution WENO interpolation

### 3 Detailed Strategy for Obtaining the Higher Order Derivatives in Eq. (15)

Taken by itself, the first three terms of Eq. (15) would only achieve second order of accuracy in space. Equation (15) shows that the final AFD-WENO scheme requires several higher order derivatives to achieve its higher order spatial accuracy. These derivatives are essential if higher order accuracy has to be achieved by the overall scheme when it is used to treat a solution which is smooth on the mesh. However, if the solution is non-smooth on the mesh, these higher order derivatives can even be a source of unphysical oscillations. We seek an automatic process that evaluates these derivatives with sufficient accuracy to meet the design accuracy of the scheme when the solution is smooth. However, this automatic process should also suppress these derivatives when they are likely to trigger spurious oscillations.

In Balsara et al. [6], a novel type of WENO interpolation was designed which does just that. Figure 2, which is modified from Balsara et al. [6], shows how the set of zone-centered fluxes  $\{F(U_i)\}$  is used to obtain the higher order derivatives of the flux at the zone boundaries. But realize from examining Eq. (15) that higher derivatives of  $\{F(U_i)\}$  are needed at the zone boundaries where they provide higher order contributions to the flux terms when the solution is smooth. Section 4 of Balsara et al. [6] provides such a WENO interpolation strategy and provides all the explicit formulae for obtaining the result. This interpolation strategy is illustrated in Fig. 2. For a third-order AFD-WENO scheme,  $[\partial_x^2 \hat{F}]_{i+1/2}$  is needed at the zone boundary “ $i + 1/2$ ” with third order of accuracy. Therefore, the magenta colored left-biased stencil and the blue colored right-biased



**Fig. 3** Part of the mesh around the zone boundary “ $i+1/2$ ”. The state vectors are available pointwise at the zone centers, as shown by the thick dots. The zone boundaries are shown by the vertical lines. The figure also shows the stencils associated with the zone boundary “ $i+1/2$ ” for the third- and fifth-order AFD-WENO schemes. We have two smaller third-order stencils and a large sixth-order stencil. For a third-order AFD-WENO scheme, the two smaller stencils can be non-linearly hybridized. In that case, the second derivatives of the state can be obtained at the zone boundary when the smoothness in the solution warrants it. For the fifth-order AFD-WENO, the two smaller stencils can be non-linearly hybridized along with the larger stencil. In that case, the second and fourth derivatives of the state can be obtained at the zone boundary when the smoothness in the solution warrants it. The process described here can be done for Adaptive Order and Multiresolution WENO interpolation

stencil in Fig. 2 can be non-linearly hybridized to get  $[\partial_x^2 \hat{\mathbf{F}}]_{i+1/2}$ . For a fifth-order AFD-WENO scheme,  $[\partial_x^2 \hat{\mathbf{F}}]_{i+1/2}$  and  $[\partial_x^4 \hat{\mathbf{F}}]_{i+1/2}$  both need to be evaluated at the zone boundary “ $i+1/2$ ” with fifth order of accuracy. To achieve that, the magenta colored left-biased stencil, the blue colored right-biased stencil, and the red colored large sixth-order accurate central stencil in Fig. 2 need to be non-linearly hybridized to get  $[\partial_x^2 \hat{\mathbf{F}}]_{i+1/2}$  and  $[\partial_x^4 \hat{\mathbf{F}}]_{i+1/2}$  at the zone boundaries. The exact details of this novel WENO interpolation are given in Sect. 4 of Balsara et al. [6]. This paragraph, along with Fig. 2, has shown us how the last curly bracket term in Eq. (15) is obtained. It gives us the higher order contributions to the flux terms when the solution is smooth.

Extensive numerical experimentation in Balsara et al. [6] shows that the AFD-WENO schemes in the conservation form are indeed very robust when applied to conservation laws. This is partly because the Lax-Wendroff theorem comes to the rescue. This is also why we wrote the higher order non-conservative products in Eq. (15) in a form that is as close as possible to a finite difference-like form. We see from Fig. 3 that the same strategy that was used in Fig. 2 for obtaining higher order derivatives of the fluxes at the zone boundaries can also be used for obtaining the higher derivatives of the state “ $\mathbf{U}$ ” at the zone boundaries.

This completes our description of the non-linearly hybridized WENO interpolation processes that will be used to extract all the higher derivatives that we see in Eq. (15).

## 4 Pointwise Implementation of Our AFD-WENO Scheme for Hyperbolic PDEs with Non-conservative Products

The AFD-WENO scheme that we present here is based on the WENO interpolation methods described in Sect. 3 of Balsara et al. [6]. In that section, we present Adaptive Order as well as Multiresolution methods for starting from point values at zone centers and using them to obtain interpolated values at the zone boundaries for feeding to a Riemann solver; as shown in Fig. 1. We will also draw on WENO interpolation methods described in Sect. 4 of Balsara et al. [6] where we show that the set of flux terms  $\{\mathbf{F}(\mathbf{U}_i)\}$  that are evaluated at zone centers can have their higher derivatives interpolated to the zone boundaries, as shown in Fig. 2. We will also use similar methods to take the zone-centered state vector “ $\mathbf{U}$ ” and evaluate its higher derivatives at zone boundaries; see Fig. 3. While the WENO interpolation in Sects. 3 and 4 of Balsara et al. [6] is couched in the language of WENO-AO, all the results there can be easily transcribed to the Multiresolution WENO from (Zhu and Shu [51]); as a result we will show results from WENO-AO and Multiresolution WENO in this paper, displaying that in all cases their effective accuracies are almost identical.

The Riemann solver that we will use in this entire work will be the HLLI Riemann solver from Dumbser and Balsara [19] where the philosophy of such Riemann solvers is explained. In Sect. 5 of Balsara et al. [5] we also catalogue results from Dumbser and Balsara [19] in a notation that is more suited for the WENO interpolation shown in Fig. 1.

We provide a pointwise implementation of our AFD-WENO scheme for treating non-linear hyperbolic PDEs that have non-conservative products. Some of the steps are similar to Balsara et al. [6] and some are different. AFD-WENO schemes are always implemented in the dimension-by-dimension fashion, so we only describe one of the dimensional updates here. We realize that the update equation, i.e., Eq. (15), has a lot of terms. The optimal sequence of steps given below is designed so that at the end of each step we catalogue the parts of Eq. (15) that are in hand. Consequently, by the end of these steps, we can finally assemble the entire update equation. The pointwise implementation of our AFD-WENO scheme into a numerical code goes according to the following steps.

- (i) We start with the mesh function as shown in Fig. 1. This means that at each zone center  $x_i$  we have a pointwise value for the conserved variable  $\mathbf{U}_i$ .
- (ii) Starting from the conserved variables in each zone, obtain the primitive variables. Use the conserved and primitive variables, as needed, to obtain the normalized right and left eigenvectors in the conserved variables. Also evaluate  $\mathbf{F}(\mathbf{U}_i)$  and  $\mathbf{C}(\mathbf{U}_i)$  pointwise at the center of each zone “ $i$ ”.
- (iii) As shown in Fig. 1, we use the WENO-AO algorithm from Sect. 3 of Balsara et al. [6]. (We have also used the Multiresolution WENO interpolation from Zhu and Shu [51] and found no difference.) That section includes all closed form expressions that are needed for the WENO interpolation in one dimension. This consists of making a non-linear hybridization between a large high order accurate stencil and smaller lower order accurate stencils. The neighboring zones around zone “ $i$ ” are projected into the characteristic space of zone “ $i$ ”. The third- and fifth-order cases are explicitly shown in Fig. 1. Once the variables in the neighboring zones around zone “ $i$ ” are projected into the characteristic space of zone “ $i$ ”, the WENO-AO interpolation is carried out in the characteristic space. Projecting the interpolated characteristic variables back into the space of right eigenvectors gives us high order accurate  $\hat{\mathbf{U}}_{i+1/2}^-$  and  $\hat{\mathbf{U}}_{i-1/2}^+$  within each zone “ $i$ ”, as shown in Fig. 1. Since this step involves project-

- ing all the zones in all the stencils of interest into the characteristic space of each zone “ $i$ ” using eigenvectors, it is one of the three computationally expensive steps of the algorithm. By the end of this step we should have the WENO interpolation-based  $\hat{\mathbf{U}}_{i+1/2}^-$  and  $\hat{\mathbf{U}}_{i+1/2}^+$  at each zone boundary.
- (iv) At each zone boundary  $x_{i+1/2}$ , use the left and right states  $\hat{\mathbf{U}}_{i+1/2}^-$  and  $\hat{\mathbf{U}}_{i+1/2}^+$  to obtain the left-most and right-most going speeds of the Riemann fan; these are denoted by  $S_{L,i+1/2}$  and  $S_{R,i+1/2}$ . Please note that we are not yet seeking the resolved state within the Riemann fan.
- (v) Now, at each zone boundary  $x_{i+1/2}$ , we hand in the speeds  $S_{L,i+1/2}$  and  $S_{R,i+1/2}$  as well as the states  $\hat{\mathbf{U}}_{i+1/2}^-$  and  $\hat{\mathbf{U}}_{i+1/2}^+$  to the Riemann solver. Unlike the situation with conservation laws (where the Riemann solver returns a resolved flux), we now ask the Riemann solver to return fluctuations  $\mathbf{D}^{*-}(\hat{\mathbf{U}}_{i+1/2}^-, \hat{\mathbf{U}}_{i+1/2}^+)$  and  $\mathbf{D}^{*+}(\hat{\mathbf{U}}_{i+1/2}^-, \hat{\mathbf{U}}_{i+1/2}^+)$  as well as the resolved state  $\mathbf{U}_{i+1/2}^*$  at each zone boundary “ $i + 1/2$ ”. This is shown in Fig. 1. By the end of this step we should have the resolved state  $\mathbf{U}_{i+1/2}^*$  as well as the corresponding fluctuations  $\mathbf{D}^{*-}(\hat{\mathbf{U}}_{i+1/2}^-, \hat{\mathbf{U}}_{i+1/2}^+)$  and  $\mathbf{D}^{*+}(\hat{\mathbf{U}}_{i+1/2}^-, \hat{\mathbf{U}}_{i+1/2}^+)$  at each zone boundary.
- (vi) If one wants to make a characteristic projection of the higher derivatives of the flux variables, we can do that using  $\mathbf{U}_{i+1/2}^*$ . This could be useful in the next step. Therefore, we find the matrices of right and left eigenvectors corresponding to the resolved state  $\mathbf{U}_{i+1/2}^*$  at each zone boundary “ $i + 1/2$ ”. Please notice that if the HLLI Riemann solver is used, then we will naturally be constructing the left and right eigenvectors from the resolved state of that Riemann solver. Therefore, it is worthwhile to derive the maximum use from those eigenvectors.
- (vii) Use the boundary-centered WENO-AO interpolation scheme from Sect. 4 of Balsara et al. [6] and Fig. 2 of this paper to start with the zone-centered flux variables and interpolate their higher derivatives to the zone boundaries. This gives us suitably high order derivatives of the flux variables at each zone boundary. With these high order derivatives of the flux variables, we can evaluate all the higher order flux derivatives that contribute to each zone boundary; see the last term Eq. (15). (Since this step may involve projecting all the zones in all the stencils of interest into the characteristic space of each zone boundary “ $i + 1/2$ ” using eigenvectors, it is the second of the three computationally expensive steps of the algorithm. However, we have found this characteristic projection to be unnecessary, with the result that the WENO interpolation can be applied directly to the flux components.) By the end of this step we should have higher order derivatives of the flux like  $\left[\partial_x^2 \hat{\mathbf{F}}\right]_{i+1/2}$ ,  $\left[\partial_x^4 \hat{\mathbf{F}}\right]_{i+1/2}$ ,  $\left[\partial_x^6 \hat{\mathbf{F}}\right]_{i+1/2}$ , and  $\left[\partial_x^8 \hat{\mathbf{F}}\right]_{i+1/2}$  (as needed) at each of the zone boundaries. (If the PDE is a conservation law, then by this step we have all the pieces in hand that are needed for the assembly of the update equation in Eq. (15).)
- (viii) This step is closely analogous to the previous step; except that we now consider the zone centered state “ $\mathbf{U}$ ” instead of the flux. Use the boundary-centered WENO-AO interpolation scheme from Sect. 4 of Balsara et al. [6] and Fig. 3 of this paper to start with the zone-centered state variables and interpolate their higher derivatives to the zone boundaries. By the end of this step we should have higher order derivatives of the state like  $\left[\partial_x^2 \hat{\mathbf{U}}\right]_{i+1/2}$ ,  $\left[\partial_x^4 \hat{\mathbf{U}}\right]_{i+1/2}$ ,  $\left[\partial_x^6 \hat{\mathbf{U}}\right]_{i+1/2}$ , and  $\left[\partial_x^8 \hat{\mathbf{U}}\right]_{i+1/2}$  (as needed) at each of the zone boundaries.

- (ix) Now realize from the previous steps that we have acquired all the terms that will contribute to Eq. (15). We assemble Eq. (15) which is our update equation. That gives us one spatially higher order update stage of a multistage RK update strategy.
- (x) The above points have only shown one stage of the scheme. It can be coupled with an SSP-RK update strategy, say from Shu and Osher [43] or Spiteri and Ruuth [45, 46], to achieve higher order in time.
- (xi) Some of the PDEs also have stiff source terms; these are usually relaxation terms that enable the system to relax to several useful physical limits. The AFD-WENO method makes it very simple to treat stiff source terms because the source terms are treated pointwise and are collocated at the exact same location as the primal variables. For this reason, when stiff source terms are present, we recommend using the Runge-Kutta IMEX methods from Pareschi and Russo [35]; see also Kupka et al. [32].

We remind the reader that Eq. (15) has been explicitly written out in “Appendix A” as it would pertain to the implementation of third-, fifth-, seventh-, and ninth-order AFD-WENO schemes. When making an implementation in code we advise the computational scientist to first try and implement the first line of Eq. (15) and make it work for several test problems. The subsequent parts of Eq. (15) can be implemented after that.

## 5 Accuracy Analysis and Speed Comparisons

In this section, we do not consider hyperbolic conservation laws. The reason is that the current algorithm reduces exactly to the algorithm in Balsara et al. [6] where we indeed presented many accuracy analyses as applied to several conservation laws. Section 5.1 presents the accuracy analysis for the Baer-Nunziato system. Section 5.2 presents the accuracy analysis for the two-layer shallow water system. Section 5.3 presents the accuracy analysis for the multiphase debris flow model of Pitman and Le. In each instance, the test problems presented here are identical to the ones presented in Balsara et al. [5]. For that reason, we do not describe the test problems in great detail in this paper. In Sect. 5.4 we cross-compare the speeds of classical FD-WENO, conservative AFD-WENO, and the AFD-WENO schemes which include non-conservative products.

Many of the results that follow are tabulated for both the WENO-AO-based interpolation method (see Balsara et al. [8], Balsara et al. [6]) as well as the Multiresolution WENO-based method (Zhu and Shu [51], Zhu and Qiu [50]) showing that in all instances the two interpolation options yield almost identical accuracies.

### 5.1 Accuracy Analysis for the Two-Dimensional Baer-Nunziato Model for Compressible Multi-phase Flows

In this section we focus on the Baer-Nunziato compressible multiphase flow proposed by Baer and Nunziato [2] and extensively studied at the numerical level by Saurel and Abgrall [39], Adrianov and Warnecke [1], Schwendeman et al. [40], Dumbser et al. [24], Tokareva and Toro [47], Coquel et al. [18], and Chiochetti and Müller [16]. A very useful set of eigenvectors has been presented in Tokareva and Toro [47].

The PDE system assumes two phases, a solid phase denoted by the density  $\rho_1$ , the volume fraction  $\phi_1$ , the velocity  $\mathbf{v}_1 = (u_1, v_1, w_1)$ , and a pressure  $p_1$ , and a gas phase denoted by the density  $\rho_2$ , the volume fraction  $\phi_2$ , the velocity  $\mathbf{v}_2 = (u_2, v_2, w_2)$ , and a pressure  $p_2$ . The two

phases have an interfacial pressure  $P_I$  and an interfacial velocity  $\mathbf{V}_I$ , but it is suggested in Baer and Nunziato [2] to set  $P_I = p_2$  and  $\mathbf{V}_I = \mathbf{v}_1$ . The total energy density for phase “ $j$ ” is related to the specific internal energy  $e_j$  by  $\rho_j E_j = \rho_j e_j + \rho_j \mathbf{v}_j^2/2$ ,

$$\begin{aligned}\partial_t(\phi_1 \rho_1) + \nabla \cdot (\phi_1 \rho_1 \mathbf{v}_1) &= 0, \\ \partial_t(\phi_1 \rho_1 \mathbf{v}_1) + \nabla \cdot (\phi_1 (\rho_1 \mathbf{v}_1 \otimes \mathbf{v}_1 + \mathbf{I} p_1)) - P_I \nabla \phi_1 &= 0, \\ \partial_t(\phi_1 \rho_1 E_1) + \nabla \cdot (\phi_1 \mathbf{v}_1 (\rho_1 E_1 + p_1)) + P_I \partial_t \phi_1 &= 0, \\ \partial_t(\phi_2 \rho_2) + \nabla \cdot (\phi_2 \rho_2 \mathbf{v}_2) &= 0, \\ \partial_t(\phi_2 \rho_2 \mathbf{v}_2) + \nabla \cdot (\phi_2 (\rho_2 \mathbf{v}_2 \otimes \mathbf{v}_2 + \mathbf{I} p_2)) - P_I \nabla \phi_2 &= 0, \\ \partial_t(\phi_2 \rho_2 E_2) + \nabla \cdot (\phi_2 \mathbf{v}_2 (\rho_2 E_2 + p_2)) + P_I \partial_t \phi_2 &= 0, \\ \partial_t \phi_1 + \mathbf{V}_I \cdot \nabla \phi_1 &= 0.\end{aligned}$$

The system requires that the phases volume fractions add up to unity,  $\phi_1 + \phi_2 = 1$ . The closure relations for each phase are also given by

$$\rho_j e_j = \frac{p_j + \gamma_j \pi_j}{\gamma_j - 1},$$

where  $\gamma_j$  is the ratio of specific heats and  $\pi_j$  is a constant. For the above EOS, the sound speed  $c_j$  in each phase is given by

$$c_j = \sqrt{\gamma_j \frac{p_j + \pi_j}{\rho_j}}.$$

In Sect. 4 of Dumbser et al. [22] a two-dimensional smooth vortex test problem was designed for the Baer-Nunziato flow. The problem is an analogue of the hydrodynamic vortex. In Balsara et al. [5] the problem was described in detail, so we do not describe it again here. For the fifth, seventh, and ninth orders, we also had to double the size of the computational domain to obtain results that were independent of the exponential fall off in the velocity of the vortex. Table 1 is based on the algorithm described in Eq. (15). It uses three WENO interpolation steps per RK stage and per dimension. In Table 1 we show the accuracy results on using the WENO-AO interpolation and also the Multiresolution WENO interpolation. We see that both algorithms reach their design accuracies and obtain completely comparable results. Table 2 is based on the algorithm described in Eq. (13). It uses two WENO interpolation steps per RK stage and per dimension. We see that it too meets the design accuracy, with the exception that at third order its accuracy degrades by some amount.

## 5.2 Accuracy Analysis for the Two-Dimensional Two-Layer Shallow Water Equations

In this section, we focus on the two-layer shallow water equations from Castro et al. [14]. The PDE system in 2D is given by defining  $h_1, u_1, v_1$  as the height of the upper fluid, its  $x$ -velocity, and its  $y$ -velocity, respectively, and by defining  $h_2, u_2, v_2$  as the height of the lower fluid, its  $x$ -velocity, and its  $y$ -velocity, respectively. The bottom topography is denoted by “ $b$ ” and “ $g$ ” is the gravity. The ratio  $\rho \equiv \rho_1/\rho_2$  denotes the ratio of the fluid densities. The total surface height, therefore, becomes  $\eta = \eta_1 = b + h_1 + h_2$  and the surface elevation of the interior layer is denoted by  $\eta_2 = b + h_2$ . The conservation law is given by

$$\frac{\partial}{\partial t} \begin{pmatrix} h_1 \\ h_1 u_1 \\ h_1 v_1 \\ h_2 \\ h_2 u_2 \\ h_2 v_2 \\ b \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} h_1 u_1 \\ h_1 u_1^2 + gh_1^2/2 \\ h_1 u_1 v_1 \\ h_2 u_2 \\ h_2 u_2^2 + gh_2^2/2 \\ h_2 u_2 v_2 \\ 0 \end{pmatrix} + \frac{\partial}{\partial y} \begin{pmatrix} h_1 v_1 \\ h_1 u_1 v_1 \\ h_1 v_1^2 + gh_1^2/2 \\ h_2 v_2 \\ h_2 u_2 v_2 \\ h_2 v_2^2 + gh_2^2/2 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ gh_1 \partial_x h_2 + gh_1 \partial_x b \\ 0 \\ 0 \\ \rho gh_2 \partial_x h_1 + gh_2 \partial_x b \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ gh_1 \partial_y h_2 + gh_1 \partial_y b \\ 0 \\ 0 \\ \rho gh_2 \partial_y h_1 + gh_2 \partial_y b \\ 0 \end{pmatrix} = 0.$$

All the linearly degenerate eigenvectors in both directions can be evaluated analytically. The non-linear eigenvectors and their eigenvalues have to be found numerically.

In Sect. 4.1 of Dumbser et al. [21] a two-dimensional smooth vortex test problem was designed for the two-layer shallow water flow model. In Balsara et al. [5] the problem was described in detail, so we do not describe it again here. For fifth, seventh, and ninth orders, we also had to double the size of the computational domain to obtain results that were independent of the exponential fall off in the velocity of the vortex. Table 3 is based on the algorithm described in Eq. (15). It uses three WENO interpolation steps per RK stage and per dimension. In Table 3 we show the accuracy results on using the WENO-AO interpolation and also the Multiresolution WENO interpolation. We see that both algorithms reach their design accuracies and obtain comparable results.

### 5.3 Accuracy Analysis for the Two-Dimensional Multiphase Debris Flow Model of Pitman and Le

In this section, we consider the multiphase debris flow model of Pitman and Le [37]. We use the formulation of Pelanti et al. [36] instead of the original formulation by Pitman and Le [37]. The PDE system in 2D is given by defining  $h_s$ ,  $u_s$ ,  $v_s$  as the solid height, the solid  $x$ -velocity, and the solid  $y$ -velocity, respectively, and by defining  $h_f$ ,  $u_f$ ,  $v_f$  as the fluid height, the fluid  $x$ -velocity, and the solid  $y$ -velocity, respectively. We also have  $h_f = (1 - \phi)h$  where  $\phi$  is the solid volume fraction and  $h$  is the total height. The variable “ $b$ ” refers to the bottom topography and is kept constant in all our test problems. The variable “ $g$ ” refers to the gravitational acceleration and “ $\rho \equiv \rho_f / \rho_s$ ” refers to the density ratio of the fluid and the solid. The time evolutionary equations for this model are

$$\frac{\partial}{\partial t} \begin{pmatrix} h_s \\ h_s u_s \\ h_s v_s \\ h_f \\ h_f u_f \\ h_f v_f \\ b \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} h_s u_s \\ h_s u_s^2 + gh_s^2 \rho / 2 + g(1 - \rho)h_s(h_f + h_s)/2 \\ h_s u_s v_s \\ h_f u_f \\ h_f u_f^2 + gh_f^2 / 2 \\ h_f u_f v_f \\ 0 \end{pmatrix} + \frac{\partial}{\partial y} \begin{pmatrix} h_s v_s \\ h_s u_s v_s \\ h_s v_s^2 + gh_s^2 \rho / 2 + g(1 - \rho)h_s(h_f + h_s)/2 \\ h_f v_f \\ h_f u_f v_f \\ h_f v_f^2 + gh_f^2 / 2 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ gh_s \rho \partial_x h_f + gh_s \partial_x b \\ 0 \\ 0 \\ gh_f \partial_x h_s + gh_f \partial_x b \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ gh_s \rho \partial_y h_f + gh_s \partial_y b \\ 0 \\ 0 \\ gh_f \partial_y h_s + gh_f \partial_y b \\ 0 \end{pmatrix} = 0.$$

**Table 1** The accuracy of the two-dimensional Baer-Nunziato Vortex problem using the AFD-WENO algorithm that is described in Eq. (15); the solid volume fraction is shown

WENO-AO-3	$L_1$ error	$L_1$ accuracy	$L_{\text{inf}}$ error	$L_{\text{inf}}$ accuracy
$64^2$	4.880 83E-04		2.662 78E-02	
$128^2$	9.536 35E-05	2.36	9.969 71E-03	1.42
$256^2$	1.660 69E-05	2.52	3.513 89E-03	1.50
$512^2$	2.635 32E-06	2.66	1.205 90E-03	1.54
WENO-AO-(5,3)				
$32^2$	3.367 06E-04		2.873 33E-02	
$64^2$	3.993 96E-05	3.08	3.383 73E-03	3.09
$128^2$	1.334 99E-06	4.90	1.576 72E-04	4.42
$256^2$	4.263 64E-08	4.97	5.366 02E-06	4.88
WENO-AO-(7,3)				
$32^2$	2.524 04E-04		1.260 94E-02	
$64^2$	4.940 39E-06	5.67	4.403 79E-04	4.84
$128^2$	4.350 96E-08	6.83	5.687 63E-06	6.27
$256^2$	3.580 42E-10	6.93	4.769 53E-08	6.90
WENO-AO-(9,3)				
$32^2$	3.253 16E-04		1.975 61E-02	
$64^2$	8.498 05E-07	8.58	7.639 63E-05	8.01
$96^2$	2.647 28E-08	8.56	3.097 00E-06	7.91
$128^2$	2.221 38E-09	8.61	2.700 35E-07	8.48
Order 3 multires WENO				
$64^2$	2.726 29E-04		1.649 69E-02	
$128^2$	5.037 31E-05	2.44	5.606 25E-03	1.56
$256^2$	8.833 83E-06	2.51	1.873 35E-03	1.58
$512^2$	1.358 12E-06	2.70	6.182 62E-04	1.60
Order 5 multires WENO				
$32^2$	6.033 13E-04		3.299 50E-02	
$64^2$	6.903 99E-05	3.13	8.983 65E-03	1.88
$128^2$	8.847 93E-06	2.96	1.526 44E-03	2.56
$256^2$	2.002 59E-07	5.47	6.052 37E-05	4.66
Order 7 multires WENO				
$32^2$	3.229 61E-04		1.110 33E-02	
$64^2$	5.447 78E-06	5.89	5.082 36E-04	4.45
$128^2$	6.548 79E-08	6.38	1.085 83E-05	5.55
$256^2$	3.816 89E-10	7.42	5.207 16E-08	7.70
Order 9 multires WENO				
$32^2$	2.343 61E-04		8.235 76E-03	
$64^2$	8.658 17E-06	4.76	7.153 67E-04	3.53
$96^2$	2.742 30E-08	14.19	3.250 11E-06	13.30
$128^2$	2.162 89E-09	8.83	2.540 77E-07	8.86

The first half of the table shows the results from WENO-AO interpolation with non-linear limiting. The second half shows the same results when multiresolution WENO interpolation is used

**Table 2** The accuracy of the two-dimensional Baer-Nunziato vortex problem using the AFD-WENO algorithm that is described in Eq. (13); the solid volume fraction is shown

WENO-AO-3	$L_1$ error	$L_1$ accuracy	$L_{\text{inf}}$ error	$L_{\text{inf}}$ accuracy
$64^2$	5.140 71E-04		2.664 99E-02	
$128^2$	1.092 75E-04	2.23	1.004 24E-02	1.41
$256^2$	2.189 26E-05	2.32	3.576 57E-03	1.49
$512^2$	4.261 44E-06	2.36	1.246 36E-03	1.52
WENO-AO-(5,3)				
$32^2$	3.917 23E-04		3.504 96E-02	
$64^2$	4.425 81E-05	3.15	3.395 14E-03	3.37
$128^2$	1.774 26E-06	4.64	1.619 84E-04	4.39
$256^2$	8.159 39E-08	4.44	5.825 18E-06	4.80
WENO-AO-(7,3)				
$32^2$	2.114 16E-04		1.141 72E-02	
$64^2$	5.539 98E-06	5.25	4.183 18E-04	4.77
$128^2$	5.654 13E-08	6.61	5.312 57E-06	6.30
$256^2$	6.745 73E-10	6.39	4.963 97E-08	6.74
WENO-AO-(9,3)				
$32^2$	2.374 79E-04		1.731 89E-02	
$64^2$	9.312 42E-07	7.99	7.177 48E-05	7.91
$96^2$	3.020 17E-08	8.46	2.847 21E-06	7.96
$128^2$	2.672 08E-09	8.43	2.551 34E-07	8.39

The table shows the results from WENO-AO interpolation with non-linear limiting

All the linearly degenerate eigenvectors in both directions can be evaluated analytically. The non-linear eigenvectors and their eigenvalues have to be found numerically.

In Sect. 4.2 of Dumbser et al. [21], a two-dimensional smooth vortex test problem was designed for the debris flow model. In Balsara et al. [5] the problem was described in detail, so we do not describe it again here. As previously, to minimize the effect of small jumps in the velocity field at the periodic boundaries, we double the computational domain and stopping time for the fifth-, seventh-, and ninth-order schemes. In Table 4, we show the accuracy results on using the WENO-AO interpolation and also the Multiresolution WENO interpolation. The algorithm uses three WENO interpolation steps per RK stage and per dimension. As before, we see that the scheme reaches its design accuracy.

## 5.4 Speed Comparisons

At this point, it is worthwhile to make estimates of the computational cost of various well-known finite difference WENO algorithms. We do that in this paragraph and in the next two paragraphs that follow. The FD-WENO scheme described in Balsara et al. [5] entails one WENO reconstruction and one call to the Riemann solver. Since the Riemann solvers tend to be light-weight, the majority of the computational cost is in the WENO reconstruction, which absolutely must be done in the characteristic space if the problem has any significant discontinuities. The FD-WENO scheme described in Balsara et al. [5] can handle hyperbolic systems with non-conservative products, but it does not guarantee the existence of a flux conservative form when a conservation law is used. Since the classical FD-WENO

**Table 3** The accuracy of the two-dimensional two-layer shallow water equations vortex problem using the AFD-WENO algorithm that is described in Eq. (15); the variable  $h_1$  is shown

WENO-AO-3	$L_1$ error	$L_1$ accuracy	$L_{\text{inf}}$ error	$L_{\text{inf}}$ accuracy
$64^2$	3.445 83E-03		7.629 37E-02	
$128^2$	7.433 27E-04	2.21	2.132 10E-02	1.84
$256^2$	1.170 72E-04	2.67	5.042 81E-03	2.08
$512^2$	1.662 79E-05	2.82	1.289 14E-03	1.97
WENO-AO-(5,3)				
$64^2$	3.533 68E-04		2.430 11E-02	
$128^2$	3.517 96E-05	3.33	9.021 15E-03	1.43
$256^2$	1.308 11E-06	4.75	3.893 11E-04	4.53
$512^2$	4.207 83E-08	4.96	1.282 75E-05	4.92
WENO-AO-(7,3)				
$64^2$	2.260 27E-04		3.914 91E-02	
$128^2$	3.504 01E-06	6.01	1.381 03E-03	4.83
$256^2$	3.202 43E-08	6.77	1.352 88E-05	6.67
$512^2$	2.630 13E-10	6.93	1.113 19E-07	6.93
WENO-AO-(9,3)				
$32^2$	6.278 01E-03		4.502 78E-01	
$64^2$	2.195 48E-04	4.84	2.128 22E-02	4.40
$128^2$	4.427 86E-07	8.95	1.995 59E-04	6.74
$256^2$	1.077 59E-09	8.68	5.091 73E-07	8.61
Order 3 multires WENO				
$64^2$	2.013 62E-03		5.076 92E-02	
$128^2$	3.776 90E-04	2.41	1.018 69E-02	2.32
$256^2$	5.474 56E-05	2.79	2.661 40E-03	1.94
$512^2$	7.659 31E-06	2.84	7.115 64E-04	1.90
Order 5 multires WENO				
$64^2$	7.388 35E-04		8.029 79E-02	
$128^2$	8.015 88E-05	3.20	1.253 10E-02	2.68
$256^2$	3.341 84E-06	4.58	8.724 71E-04	3.84
$512^2$	6.083 24E-08	5.78	2.787 90E-05	4.97
Order 7 multires WENO				
$64^2$	4.218 98E-04		4.340 15E-02	
$128^2$	1.370 73E-05	4.94	3.321 10E-03	3.71
$256^2$	1.156 96E-07	6.89	7.376 85E-05	5.49
$512^2$	6.287 99E-10	7.52	4.499 76E-07	7.36
Order 9 multires WENO				
$32^2$	6.271 52E-03		4.416 70E-01	
$64^2$	7.007 17E-04	3.16	1.077 67E-01	2.04
$128^2$	6.088 38E-06	6.85	1.529 32E-03	6.14
$256^2$	9.387 93E-09	9.34	5.703 32E-06	8.07

The first half of the table shows the results from WENO-AO interpolation with non-linear limiting. The second half shows the same results when multiresolution WENO interpolation is used

of Shu and Osher [44] also entailed two WENO reconstruction steps in characteristic variables, the FD-WENO scheme from Balsara et al. [5] is expected to cost about half as much as the classical FD-WENO of Shu and Osher [44].

The AFD-WENO scheme developed in Balsara et al. [6] for conservation laws requires two WENO interpolation steps and one call to a (light-weight) Riemann solver. The first of these WENO interpolation steps must be done in characteristic variables if the physical problem has discontinuities. The cost of the second WENO interpolation step may be reduced by doing it in a component by component fashion; this is the choice we make in this section. Since the classical FD-WENO of Shu and Osher [44] entailed two WENO reconstruction steps in characteristic variables, the method in Balsara et al. [6] is expected to cost less. Both methods can only treat conservation laws; however, the AFD-WENO method is much more versatile in terms of the Riemann solvers it can handle and in terms of its ability to work on curvilinear meshes.

The AFD-WENO scheme developed in this paper is more versatile because it can handle hyperbolic systems with non-conservative products. It does this while guaranteeing the variables that are still in the conservation form are updated with a strictly flux conservative update. The method in this paper requires three WENO interpolation steps and one call to a (light-weight) Riemann solver. The first of these WENO interpolation steps must be done in characteristic variables if the physical problem has discontinuities. The cost of the second and third WENO interpolation steps may be substantially reduced by doing them in a component by component fashion; this is the choice that we make here. It could, therefore, be slightly more expensive than the classical FD-WENO of Shu and Osher [44], but it is much more versatile. It too can handle different types of Riemann solvers (as long as they can be written in the fluctuation form) and it too can work on curvilinear meshes.

Lastly, and perhaps most importantly, AFD-WENO schemes have been used in the past with central differences for the higher order derivatives. The interested reader might wish to know whether using central differences for the higher order derivatives would give a scheme that is substantially faster than the current schemes that use the WENO interpolation for their second and third steps? To answer that question, we have also gathered timing statistics for AFD-WENO schemes when only central differences are used for the higher order derivatives. This answers the question of efficiency. One major purpose of scientific computation is to do new problems for which the solution is initially not known. The other major purpose of scientific computation is to use well-tested methods on the newer classes of PDEs that arise in different fields of science and engineering. Therefore, one wants numerical methods for PDEs that have been verified to work on a large range of well-known problems stemming from different types of well-known PDEs. One can, therefore, ask a further question: for general problems with discontinuities, is it effective to use just an AFD-WENO scheme that uses central differences for its higher order derivatives? We will show that it is not effective in the subsequent section.

In light of the expectations that were laid out in the above four paragraphs, it is, therefore, useful to document the actual speeds of all the above-mentioned finite difference WENO formulations on a problem where they are all applicable. For that reason, we took the very popular one-dimensional Sod shock tube problem for the Euler flow and ran it from start to finish on a 1 000 zone mesh that is evolved for 200 timesteps. Third-order accurate in time SSP-RK timestepping was used in all cases. To have a fair comparison across the competing WENO formulations, the LLF Riemann solver was used in all cases. We did this with a code that was stripped of all input and output functions so that we were able to document just the raw number of zones updated per second for this problem for the various algorithms. Also note that characteristic-based interpolation was only used in

**Table 4** The accuracy of the two-dimensional multiphase debris flow model vortex problem using the AFD-WENO algorithm that is described in Eq. (15); the variable  $h_s$  is shown

WENO-AO-3	$L_1$ error	$L_1$ accuracy	$L_{\text{inf}}$ error	$L_{\text{inf}}$ accuracy
$128^2$	6.806 23E-04		2.426 50E-02	
$256^2$	1.407 08E-04	2.27	6.957 20E-03	1.80
$512^2$	2.689 23E-05	2.39	1.772 62E-03	1.97
$1\ 024^2$	4.304 45E-06	2.64	5.524 26E-04	1.68
WENO-AO-(5,3)				
$64^2$	1.277 72E-03		1.131 81E-01	
$128^2$	1.003 27E-04	3.67	1.815 70E-02	2.64
$256^2$	3.638 59E-06	4.79	6.869 42E-04	4.72
$512^2$	1.162 63E-07	4.97	2.244 44E-05	4.94
WENO-AO-(7,3)				
$64^2$	8.294 24E-04		6.914 61E-02	
$128^2$	1.021 60E-05	6.34	2.365 19E-03	4.87
$256^2$	9.135 26E-08	6.81	2.332 27E-05	6.66
$512^2$	7.411 44E-10	6.95	1.946 74E-07	6.90
WENO-AO-(9,3)				
$32^2$	6.152 29E-03		4.404 54E-01	
$64^2$	7.896 05E-04	2.96	5.237 24E-02	3.07
$128^2$	1.360 70E-06	9.18	3.389 21E-04	7.27
$256^2$	3.289 92E-09	8.69	9.669 72E-07	8.45
Order 3 multires WENO				
$64^2$	4.161 05E-04		1.576 17E-02	
$128^2$	7.176 71E-05	2.54	3.822 40E-03	2.04
$256^2$	1.226 07E-05	2.55	8.935 16E-04	2.10
$512^2$	1.894 26E-06	2.69	2.815 95E-04	1.67
Order 5 multires WENO				
$64^2$	1.264 72E-03		7.185 82E-02	
$128^2$	1.684 68E-04	2.91	2.127 44E-02	1.76
$256^2$	7.525 55E-06	4.48	1.093 12E-03	4.28
$512^2$	2.056 92E-07	5.19	4.153 23E-05	4.72
Order 7 multires WENO				
$64^2$	1.068 38E-03		1.107 62E-01	
$128^2$	2.892 78E-05	5.21	4.475 04E-03	4.63
$256^2$	2.748 16E-07	6.72	7.939 13E-05	5.82
$512^2$	1.988 76E-09	7.11	8.115 74E-07	6.61
Order 9 multires WENO				
$32^2$	7.926 09E-03		5.180 75E-01	
$64^2$	1.240 43E-03	2.68	1.402 28E-01	1.89
$128^2$	1.316 60E-05	6.56	1.864 57E-03	6.23
$256^2$	2.339 57E-08	9.14	6.605 66E-06	8.14

The first half of the table shows the results from WENO-AO interpolation with non-linear limiting. The second half shows the same results when Multiresolution WENO interpolation is used

the first step for the algorithm described in this paper as well as the algorithm described in Balsara et al. [6]. This is justified because the additional WENO interpolation steps do not need characteristic decomposition. A single core of a modern Xeon Gold 6248R processor running at 3 GHz was used for all the runs.

In the first four paragraphs of this section, we had catalogued our expectations for the speeds of the various finite difference WENO schemes being considered. The data from Table 5 shows that the WENO scheme from Balsara et al. [5] is the fastest, which bears out our expectations. The classical FD-WENO is somewhat slower than AFD-WENO from Balsara et al. [6] because the latter scheme only uses one characteristic decomposition whereas the former scheme uses two characteristic decompositions. The AFD-WENO scheme from this paper is somewhat slower, but not by much; because it only uses one characteristic decomposition along with two interpolations that are done component by component. However, it provides more capabilities in that it can handle hyperbolic systems with conservative terms as well as non-conservative products. The AFD-WENO scheme from Eq. (13) of this paper, which is expected to be faster than the AFD-WENO described in Eq. (15) of this paper, is also shown. The AFD-WENO scheme that uses only central derivatives is nominally comparable in speed to the AFD-WENO scheme from Eq. (13) of this paper. This is understandable because the major cost in high order WENO interpolation is associated with the largest stencil, and the central differences also have to use the same large stencil that is used in the WENO interpolation!

Table 5 also allows us to cross-compare the speeds of the same algorithm at different orders of spatial accuracy. Our WENO implementations are based on a strip-mining philosophy, where a one-dimensional strip of data is extracted into long one-dimensional arrays and the CPU is asked to work on those strips of data (Colella and Woodward [17], Woodward and Colella [48]). This philosophy is very beneficial for modern CPUs with their large caches because the one-dimensional strips can then be loaded into cache very efficiently and automatically by the CPU. The CPU can then do a lot of work on each strip before releasing it back to the computer's main memory. By scanning the rows of Table 5, we see that from third-order to ninth-order, the increase in cost is very modest. This gives us another insight into the nature of finite difference WENO schemes on modern CPUs with large caches. All the finite difference WENO schemes in Table 5 incur some fixed costs, such as constructing eigenvectors, invoking Riemann solvers, or copying to and from main memory. When the WENO reconstruction or interpolation has been perfectly optimized, the larger stencils associated with higher order WENO schemes do not seem to significantly reduce the speed (i.e., the number of zones updated per second). Instead, it is now the fixed costs that begin to dominate the overall cost of the algorithm. The insight that we gain from scanning the rows of Table 5 is that the larger number of floating point operations associated with larger stencils is almost free of charge if the finite difference WENO code is designed to be cache friendly. Our well-implemented FD-WENO and AFD-WENO schemes come pretty close to the promised land of high performance computing where float point operations are almost free and the dominant costs are the other fixed costs in the implementation!

## 6 One-Dimensional Test Problems

Here we present the same one-dimensional test problems as from Balsara et al. [5] for the Baer-Nunziato system, the two-layer shallow water system, and the multiphase debris flow model. In Sect. 6.1, we focus on the Baer-Nunziato model of compressible

**Table 5** The speed comparison, expressed as number of zones updated per second, for the Sod shock problem on a one-dimensional 1 000 zone mesh that is evolved for 200 time steps

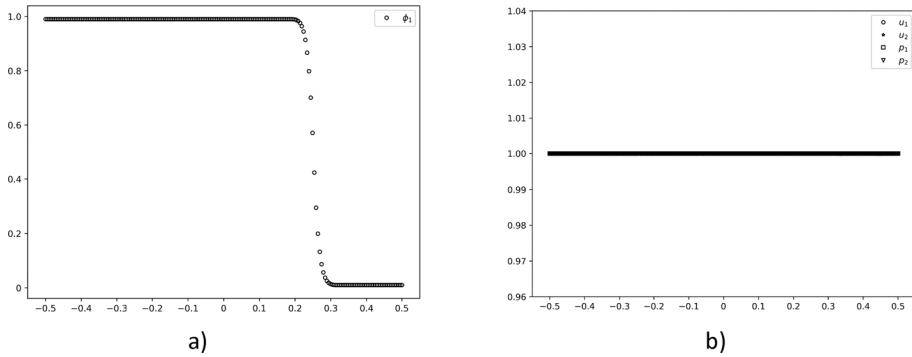
	Third-order (zones/s)	Fifth-order (zones/s)	Seventh-order (zones/s)	Ninth-order (zones/s)
Classical FD-WENO	80 260	72 672	65 374	60 685
WENO B23a	98 644	95 720	92 090	88 684
AFD-WENO B23b	91 786	84 623	78 093	76 741
AFD-WENO this paper	85 396	77 907	69 871	66 751
AFD-WENO 13	92 483	83 159	77 105	72 850
AFD-WENO central derivatives	89 302	85 474	78 654	70 042

A single core of a modern Xeon Gold 6248R processor running at 3 GHz was used for all the runs. The classical FD-WENO scheme (Shu and Osher [44], Jiang and Shu [30]) is denoted “Classical FD-WENO”; the WENO scheme from Balsara et al. [5] is denoted “WENO B23a”; the AFD-WENO scheme from Balsara et al. [6] is denoted “AFD-WENO B23b”; the AFD-WENO scheme from Eq. (15) of this paper is denoted as “AFD-WENO This Paper”. We also show the AFD-WENO algorithm from Eq. (13) of this paper, which we denote as “AFD-WENO 13”. Lastly, we also show the speed of AFD-WENO algorithm from Eq. (15) when only central derivatives are used for the higher order derivatives, and this is denoted as “AFD-WENO Central Derivatives”

multi-phase flows. In Sect. 6.2, we focus on the two-layer shallow water equations. In Sect. 6.3, we present results from the multiphase debris flow model of Pitman and Le. For all the simulations presented in this section, we used a CFL of 0.8 with a third-order SSP-RK scheme.

## 6.1 One-Dimensional Test Problems for the Baer-Nunziato Model for Compressible Multi-phase Flows

For the Baer-Nunziato model described in Sect. 5.1, we first consider the Abgrall problem from Dumbser et al. [22]. The same problem was also used in Balsara et al. [5]. The Abgrall problem, see Saurel and Abgrall [39], is based on the idea that a mixture of the two phases moving in 1D with uniform velocity and pressure should be able to continue moving in this fashion without generating any wiggles in the velocity or pressure. We present such a test problem here on a 200 zone mesh that spans the domain  $[-0.5, 0.5]$ . In the solid phase we set a stiffened EOS with  $\gamma_1 = 3$  and  $\pi_1 = 100$ , whereas in the gas phase we set  $\gamma_2 = 1.4$  and  $\pi_2 = 0$ . The pressures and longitudinal velocities in both phases are set to unity. The problem is initialized with  $\rho_{1,L} = 800$ ,  $\rho_{2,L} = 2$ ,  $\phi_{1,L} = 0.99$  on the left of  $x = 0$  and  $\rho_{1,R} = 1\,000$ ,  $\rho_{2,R} = 1$ ,  $\phi_{1,L} = 0.01$  to its right. As the Abgrall problem contains strong shocks in the volume fractions, therefore we use the flattening algorithm (as described in “Appendix B”) to detect the shock locations and there by suppressing the higher order terms in the simulation. For zone  $i$ , if we have  $\eta_i > 10^{-12}$  then we set higher order terms in Eq. (15), i.e., last two curly bracket terms, to zero. Using the available flattener, we run the problem to a final time of  $t = 0.25$  using the fifth-order accurate LLF-based AFD-WENO scheme. The result for the solid volume fraction is shown in Fig. 4a, and the pressure and velocity profiles are shown in Fig. 4b. We see that the pressure and velocity profiles are absolutely flat, showing that the scheme is able to capture the solution without generating any oscillations in the velocity and pressure profiles. The seventh- and ninth-order AFD-WENO schemes also show identical results, and, therefore, they are not presented here.



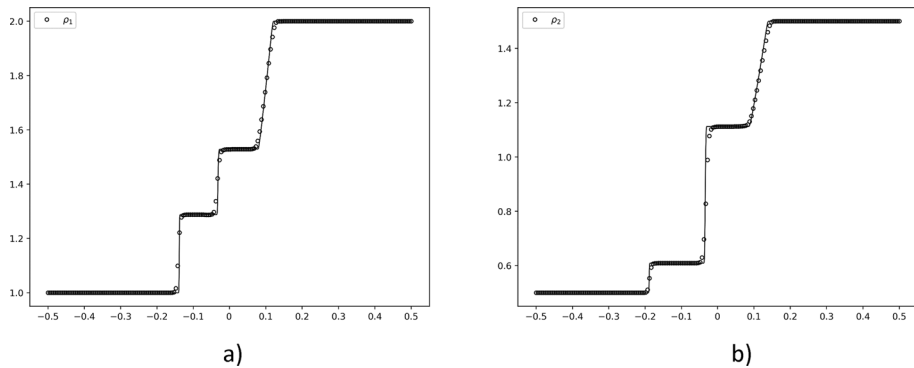
**Fig. 4** Baer-Nunziato: Abgrall problem using the fifth-order accurate LLF-based AFD-WENO scheme with 200 zones. **a** shows the solid volume fraction and **b** shows the velocities and pressures for both the phases. The seventh- and ninth-order AFD-WENO schemes also show identical results and therefore they are not shown here

Next we consider a set of six one-dimensional Riemann problems from Dumbser et al. [22]. The same set of problems was also used in Balsara et al. [5]. Table 6 catalogues the parameters for these six Riemann problems. The results for these six problems using the AFD-WENO scheme are shown in Figs. 5, 6, 7, 8, 9, and 10. All the Baer-Nunziato Riemann problems were run on a 200 zone mesh.

Figures 5a, b show the solid and gas densities for the RP1 test when a fifth-order HLL-based AFD-WENO scheme was used. Figures 6a, b show the solid and gas densities for

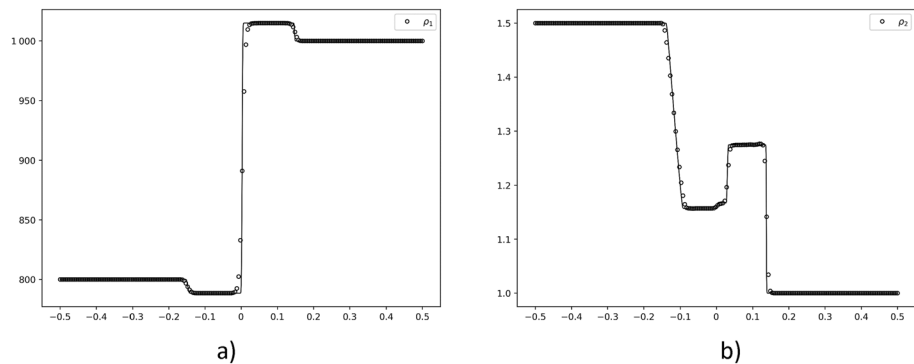
**Table 6** Left and right initial states, computational domain  $[x_L, x_R] = [-0.5, 0.5]$ , and final times ( $t_{\text{end}}$ ) of the six Riemann problems for Baer-Nunziato model

Test Case:	$\rho_1$	$u_1$	$p_1$	$\rho_2$	$u_2$	$p_2$	$\phi_1$	$t_{\text{end}}$
<b>RP1</b>	$\gamma_1 = 1.4, \pi_1 = 0, \gamma_2 = 1.4, \pi_2 = 0$							
Left:	1.0	0.0	1.0	0.5	0.0	1.0	0.4	0.1
Right:	2.0	0.0	2.0	1.5	0.0	2.0	0.8	
<b>RP2</b>	$\gamma_1 = 3.0, \pi_1 = 100, \gamma_2 = 1.4, \pi_2 = 0$							
Left:	800.0	0.0	500.0	1.5	0.0	2.0	0.4	0.1
Right:	1000.0	0.0	600.0	1.0	0.0	1.0	0.3	
<b>RP3</b>	$\gamma_1 = 1.4, \pi_1 = 0, \gamma_2 = 1.4, \pi_2 = 0$							
Left:	1.0	0.9	2.5	1.0	0.0	1.0	0.9	0.1
Right:	1.0	0.0	1.0	1.2	1.0	2.0	0.2	
<b>RP4</b>	$\gamma_1 = 3.0, \pi_1 = 3400, \gamma_2 = 1.35, \pi_2 = 0$							
Left:	1900.0	0.0	10.0	2.0	0.0	3.0	0.2	0.15
Right:	1950.0	0.0	1000.0	1.0	0.0	1.0	0.9	
<b>RP5</b>	$\gamma_1 = 1.4, \pi_1 = 0, \gamma_2 = 1.4, \pi_2 = 0$							
Left:	1.0	0.0	1.0	0.2	0.0	0.3	0.8	0.20
Right:	1.0	0.0	1.0	1.0	0.0	1.0	0.3	
<b>RP6</b>	$\gamma_1 = 1.4, \pi_1 = 0, \gamma_2 = 1.4, \pi_2 = 0$							
Left:	0.2068	1.4166	0.0416	0.5806	1.5833	1.375	0.1	0.1
Right:	2.2263	0.9366	6.0	0.4890	-0.70138	0.986	0.2	

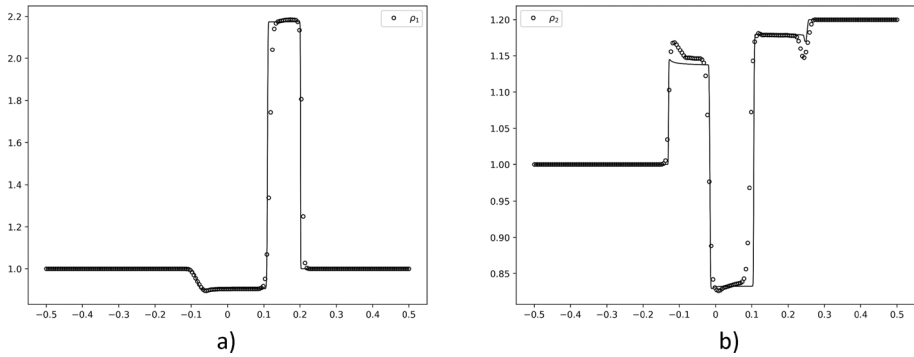


**Fig. 5** Baer-Nunziato: Riemann problem-1 using the fifth-order accurate HLL-based the AFD-WENO scheme with 200 zones. **a** shows the density for the solid phase and **b** shows the density for the gas phase. The seventh- and ninth-order AFD-WENO schemes also show identical results and therefore they are not shown here

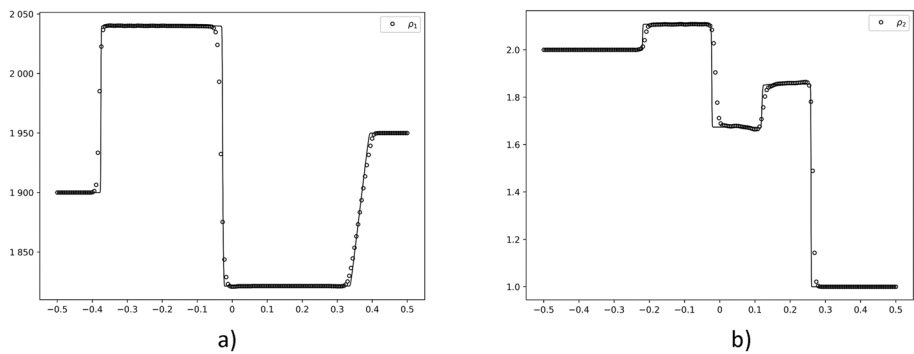
the RP2 test when a seventh-order HLL-based AFD-WENO scheme was used. Figures 7a, b show the solid and gas densities for the RP3 test when a ninth-order HLL-based AFD-WENO scheme was used. Figures 8a, b show the solid and gas densities for the RP4 test when a fifth-order LLF-based AFD-WENO scheme was used. Figures 9a, b show the solid and gas densities for the RP5 test when a seventh-order LLF-based AFD-WENO scheme was used. For this problem (RP5), we notice that the seventh- and ninth-order schemes tend to exaggerate the small oscillations near shocks if a flattener is not used. For this reason, we have invoked the flattener (given in “Appendix B”) to simulate this problem. Figures 10a, b show the solid and gas densities for the RP6 test when a ninth-order LLF-based AFD-WENO scheme was used. The other higher order LLF-based and HLL-based AFD-WENO schemes also show identical results, and therefore they are not presented here. In each of the six Riemann problems a reference solution has been shown in solid lines. The reference solution was obtained using the third-order LLF-based AFD-WENO scheme on



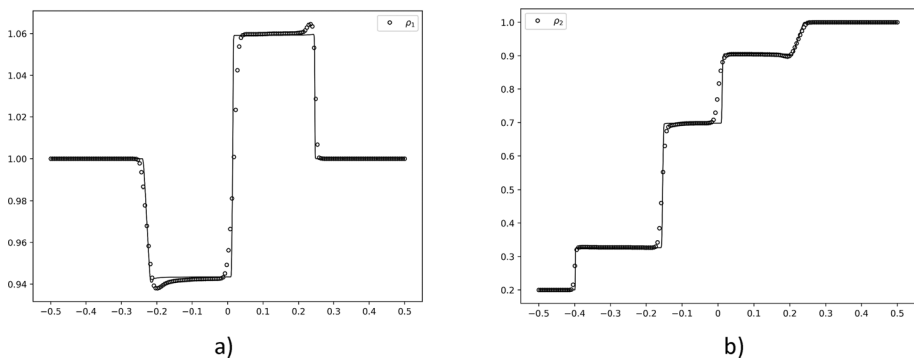
**Fig. 6** Baer-Nunziato: Riemann problem-2 using the seventh-order accurate HLL-based AFD-WENO scheme with 200 zones. **a** shows the density for the solid phase and **b** shows the density for the gas phase. The fifth- and ninth-order AFD-WENO schemes also show identical results and therefore they are not shown here



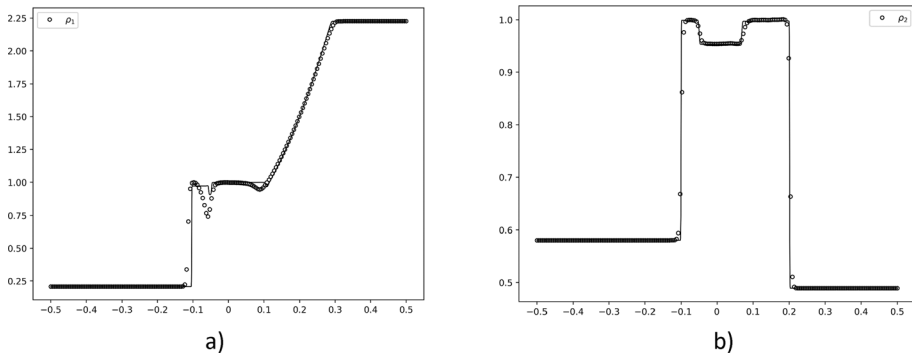
**Fig. 7** Baer-Nunziato: Riemann problem-3 using the ninth-order accurate HLL-based AFD-WENO scheme with 200 zones. **a** shows the density for the solid phase and **b** shows the density for the gas phase. The fifth- and seventh-order AFD-WENO schemes also show identical results and therefore they are not shown here



**Fig. 8** Baer-Nunziato: Riemann problem-4 using the fifth-order accurate LLF-based AFD-WENO scheme with 200 zones. **a** shows the density for the solid phase and **b** shows the density for the gas phase. The seventh- and ninth-order AFD-WENO schemes also show identical results and therefore they are not shown here



**Fig. 9** Baer-Nunziato: Riemann problem-5 using the seventh-order accurate LLF-based AFD-WENO scheme with 200 zones. **a** shows the density for the solid phase and **b** shows the density for the gas phase. We have used flattener for this problem. The fifth- and ninth-order AFD-WENO schemes also show identical results and therefore they are not shown here



**Fig. 10** Baer-Nunziato: Riemann problem-6 using the ninth-order accurate LLF-based AFD-WENO scheme with 200 zones. **a** shows the density for the solid phase and **b** shows the density for the gas phase. The fifth- and seventh-order AFD-WENO schemes also show identical results and therefore they are not shown here

**Table 7** Left and right initial states, computational domain  $[x_L, x_R]$ , location of the discontinuity ( $x_c$ ), and final times ( $t_{\text{end}}$ ) of the three Riemann problems for the two-layer shallow water model

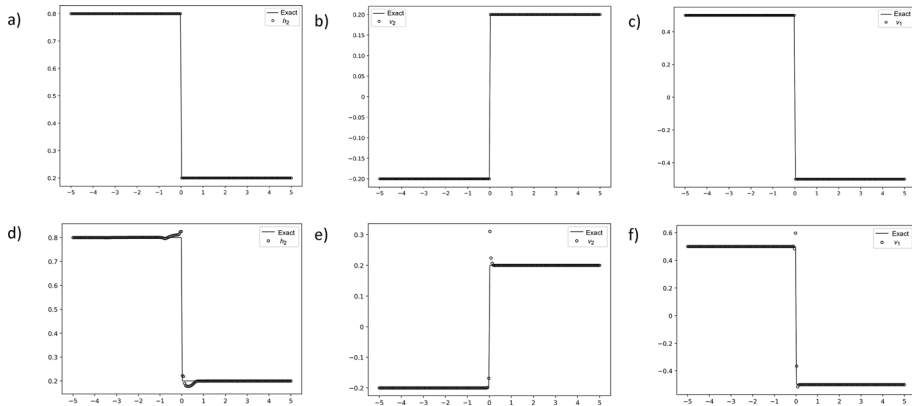
Test Case:		$h_1$	$u_1$	$v_1$	$h_2$	$u_2$	$v_2$	$b$	$x_L$	$x_R$	$x_c$	$t_{\text{end}}$
<b>RP1</b>	Left:	0.5	0.0	0.5	0.8	0.0	-0.2	0.2	-5.0	5.0	0.0	1.0
	Right:	0.5	0.0	-0.5	0.2	0.0	0.2	0.8				
<b>RP2</b>	Left:	0.4	0.0	0.0	0.6	0.0	0.0	0.0	-5.0	5.0	0.0	1.25
	Right:	0.6	0.0	0.0	0.4	0.0	0.0	0.0				
<b>RP3</b>	Left:	1.0	0.0	0.0	1.0	0.0	0.0	0.0	-5.0	5.0	0.0	1.0
	Right:	0.5	0.0	0.0	0.5	0.0	0.0	0.5				

a mesh with 2 000 zones. We can observe that all the results obtained using AFD-WENO schemes provide a clearer representation of the interfaces.

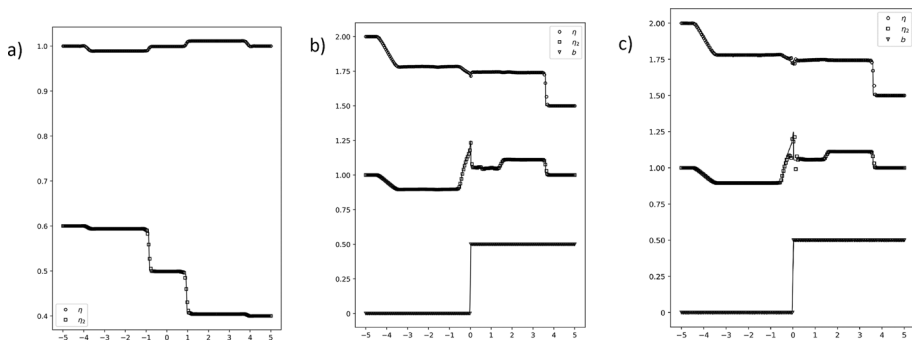
## 6.2 One-Dimensional Test Problems for Two-Layer Shallow Water Equations

For the two-layer shallow water equations described in Sect. 5.2, a set of three one-dimensional Riemann problems has been presented in Dumbser and Balsara [19], Dumbser et al. [21] and Castro et al. [15]. The same set of problems was also used in Balsara et al. [5]. Table 7 gives the relevant information for these Riemann problems. All the two-layer shallow water Riemann problems were run on a 200 zone mesh with  $\rho = 0.8$  and  $g = 9.8$ .

The first Riemann problem from Table 7 illustrates the preservation of stationary jump discontinuities in the linearly degenerate intermediate fields, and the results using the fifth-order HLLI-based AFD-WENO scheme are shown in the upper panel of Fig. 11. The results from the seventh- and ninth-order AFD-WENO schemes are identical, and therefore they are not shown here. Figure 11 shows the ability of our high order schemes to preserve stationary discontinuities in linearly degenerate intermediate fields. Figure 12a shows the results from the second Riemann problem in Table 7 computed using the seventh-order accurate HLLI-based AFD-WENO scheme. The fifth- and ninth-order schemes show the



**Fig. 11** Two-layer shallow water: Riemann problem-1 using the fifth-order accurate HLLI-based AFD-WENO scheme with 200 zones. The seventh- and ninth-order AFD-WENO schemes also show identical results and therefore they are not shown here. The top row shows the results from the scheme developed here. The bottom row of three figures shows the results from the same scheme when central differences are used for the higher order derivatives. We see that using just the central differences results in unphysical oscillations. Similar oscillations would be found if central differences were used for a seventh- or ninth-order scheme



**Fig. 12** Two-layer shallow water: **a** shows the results of Riemann problem-2 using the seventh-order accurate HLLI-based AFD-WENO scheme and **b** shows the results of Riemann problem-3 using the ninth-order accurate HLLI-based AFD-WENO scheme with 200 zones. The fifth- and ninth-order schemes for the Riemann problem-2, and the fifth- and seventh-order schemes for the Riemann problem-3 also show identical results and therefore they are not shown here. **c** shows the results of Riemann problem-3 using the ninth-order scheme when only central differences are used for the higher order derivatives. We clearly see the formation of unphysical oscillations when only central differences are used. Similar oscillations would be found if central differences were used for a fifth- or seventh-order scheme

identical results, and therefore they are not shown here. Figure 12b shows results from the third Riemann problem in Table 7 using the ninth-order accurate HLLI-based AFD-WENO scheme. The fifth- and seventh-order schemes show the identical results, and therefore, they are not shown here. A third-order AFD-WENO scheme was also run on a mesh with 2 000 zones to generate a reference solution, and the reference solution is shown with solid lines in Fig. 12. We see that the results in Fig. 12 closely match the reference solution.

**Table 8** Gives the left and right initial states, computational domain  $[x_L, x_R]$ , location of the discontinuity ( $x_c$ ) and final times ( $t_{\text{end}}$ ) of the three Riemann problems for the multiphase debris flow model

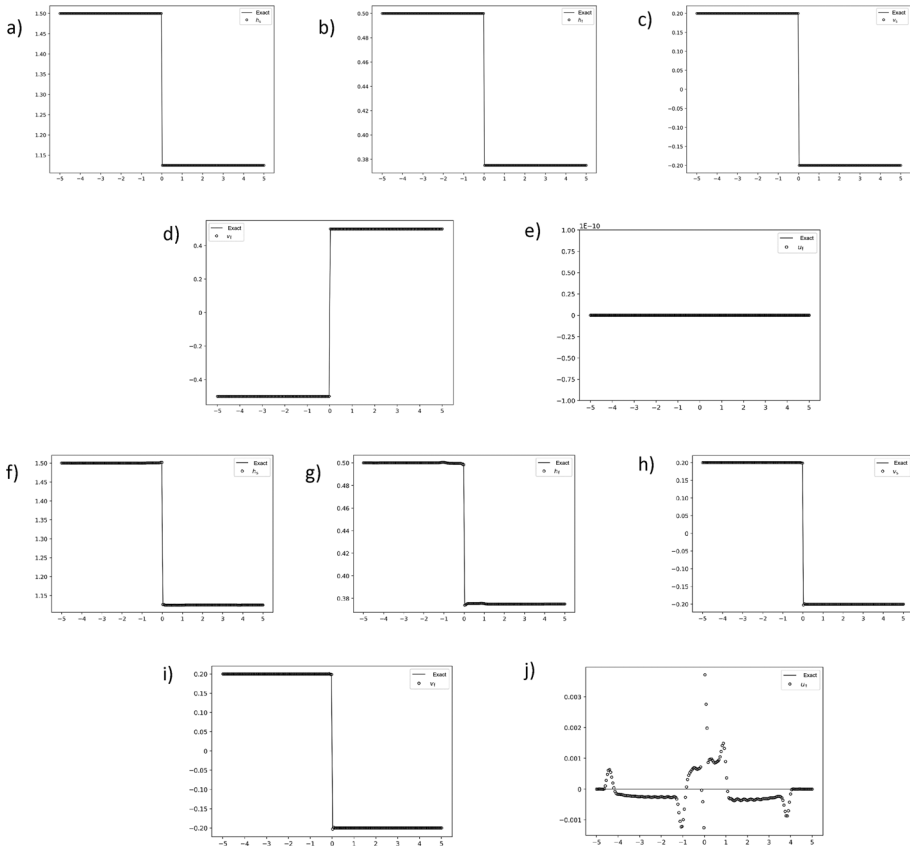
Test Case:		$h_s$	$u_s$	$v_s$	$h_f$	$u_f$	$v_f$	$b$	$x_L$	$x_R$	$x_c$	$t_{\text{end}}$
<b>RP1</b>	Left:	1.5	0.0	0.2	0.5	0.0	-0.5	0.0	-5.0	5.0	0.0	1.0
	Right:	1.125	0.0	-0.2	0.375	0.0	0.5	0.5				
<b>RP2</b>	Left:	2.1	0.0	0.0	0.9	0.0	0.0	0.0	-5.0	5.0	0.0	0.5
	Right:	0.8	0.0	0.0	1.2	0.0	0.0	0.0				
<b>RP3</b>	Left:	2.1	-1.4	0.0	0.9	0.3	0.0	0.0	-5.0	5.0	0.0	0.5
	Right:	0.8	-0.9	0.0	1.2	0.1	0.0	0.0				

Let us now address the topic of the effectiveness of an AFD-WENO scheme that uses only central differences for its higher order derivatives. RP1 was run when only central differences were used for the higher order derivatives and the results are shown in the lower panel of Fig. 11. Similarly, RP3 was run when only central differences were used for the higher order derivatives and the results are shown in Fig. 12c. In each instance, we see that there are unphysical oscillations at the discontinuities. The first key purpose of scientific computation is to do new problems which have not been attempted before and for which the solution is not known. The second key purpose is that the methods that have been developed should be applicable to newer classes of PDEs that may arise in the future in science and engineering. This requires verification of the algorithms on large classes of known solutions emerging from different types of PDEs. Before an algorithm can give reliable results for newer classes of problems, for which the solution is unknown, it is essential that it should reproduce all known problems. The results in Figs. 11 and 12 tell us that an AFD-WENO scheme that only uses central differences for its higher order derivatives would not constitute a properly verified algorithm.

### 6.3 One-Dimensional Test Problems for Multiphase Debris Flow Model of Pitman and Le

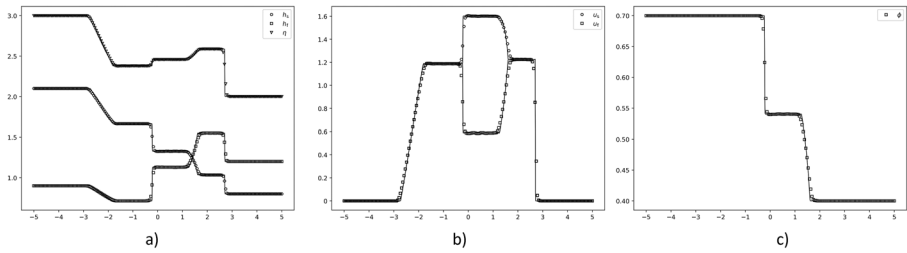
In this section, we present one-dimensional test problems for the multiphase debris flow model of Pitman and Le [37] in the formulation of Pelanti et al. [36]. The equations for the multiphase debris flow model are described in Sect. 5.3. The test problems we present are drawn from Pelanti et al. [36], Dumbser et al. [21], and Rhebergen et al. [38], and they have also been integrated in Dumbser and Balsara [19].

Table 8 describes the three Riemann problems that we present here; they are drawn from Dumbser and Balsara [19]. In all three Riemann problems we use  $\rho = 0.5$  and  $g = 9.8$ . In all cases we use a one-dimensional mesh with 200 zones. The computational domain and the final stopping time are also shown in Table 8. We use the fifth-order accurate HLLI-based AFD-WENO scheme for the first problem from Table 8. The problem corresponds to the superposition of three stationary jump discontinuities in the linearly degenerate fields. The problem was run with the anti-diffusive fluxes turned on for all the linearly degenerate waves. Figures 13a–e show that all the jumps in the linearly degenerate stationary wave families are exactly preserved on the mesh when the full algorithm is used. The seventh- and ninth-order AFD-WENO schemes also show identical results, and therefore they are not shown here. Figures 13f–j show the same results from an AFD-WENO scheme that only used central differences for the higher order

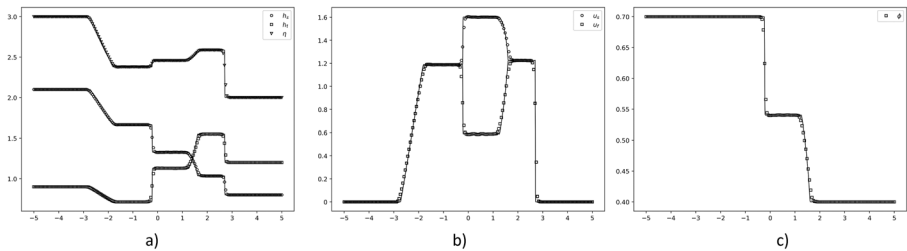


**Fig. 13** **a–e** Debris flow: **a–d** show the jump in the linearly degenerate fields using the fifth-order accurate HLLI-based AFD-WENO scheme with 200 zones. **e** shows the velocity. The seventh- and ninth-order AFD-WENO schemes also show identical results and therefore they are not shown here. **f–j** Debris flow: **f–i** show the jump in the linearly degenerate fields using the fifth-order accurate HLLI-based AFD-WENO scheme with 200 zones when only central differences are used for the higher order derivatives. **j** shows the velocity. The seventh- and ninth-order AFD-WENO schemes also show identical results and therefore they are not shown here

derivatives. We can clearly see that there are spurious oscillations; indicating that such a scheme would not pass all the verification test problems. Figure 14 shows the second Riemann problem from Table 8 obtained using the seventh-order accurate HLLI-based AFD-WENO scheme. The fifth- and ninth-order AFD-WENO schemes also show identical results, and therefore, they are not shown here. Figure 15 shows the third Riemann problem from Table 8 obtained using the ninth-order accurate HLLI-based AFD-WENO scheme. The fifth- and seventh-order AFD-WENO schemes also show identical results, and therefore they are not shown here. A third-order AFD-WENO scheme was also run on a mesh with 2 000 zones in order to generate a reference solution, and the reference solution is shown with solid lines in Figs. 14 and 15. We can observe that the results in Figs. 14 and 15 closely resemble the reference solution.



**Fig. 14** Debris flow: Riemann problem-2 using the seventh-order accurate HLLI-based AFD-WENO scheme with 200 zones. The fifth- and ninth-order AFD-WENO schemes also show identical results and therefore they are not shown here



**Fig. 15** Debris flow: Riemann problem-3 using the ninth-order accurate HLLI-based AFD-WENO scheme with 200 zones. The fifth- and seventh-order AFD-WENO schemes also show identical results and therefore they are not shown here

## 7 Multidimensional Test Problems

In this section, we present the same two-dimensional test problems as from Balsara et al. [5] for the Baer-Nunziato system, the two-layer shallow water system, and the multiphase debris flow model. In Sect. 7.1, we focus on the Baer-Nunziato model of compressible multi-phase flows. In Sect. 7.2, we focus on the two-layer shallow water equations. In Sect. 7.3, we present results from the multiphase debris flow model of Pitman and Le. For all the simulations presented in this section, we used a CFL of 0.4 with a third-order SSP-RK scheme.

### 7.1 Two-Dimensional Test Problems for the Baer-Nunziato Model for Compressible Multi-phase Flows

For the Baer-Nunziato model described in Sect. 5.1, we present two multi-dimensional problems. The first problem is the shock-bubble interaction problem from Dumbser et al. [23]. This problem was also used in Balsara et al. [5]. The second problem is an analog of Euler's Shock-Vortex interaction problem described in Balsara and Shu [11]. Such a problem for the Baer-Nunziato model has been presented in Sect. 9.2 of Balsara et al. [5]. The detailed setup for both problems is given in Balsara et al. [5]; therefore, we do not describe the setup here.

The first problem (shock-bubble interaction problem) consists of a planar, right-going shock propagating into an ambient medium that contains a bubble with a radius of 0.25.

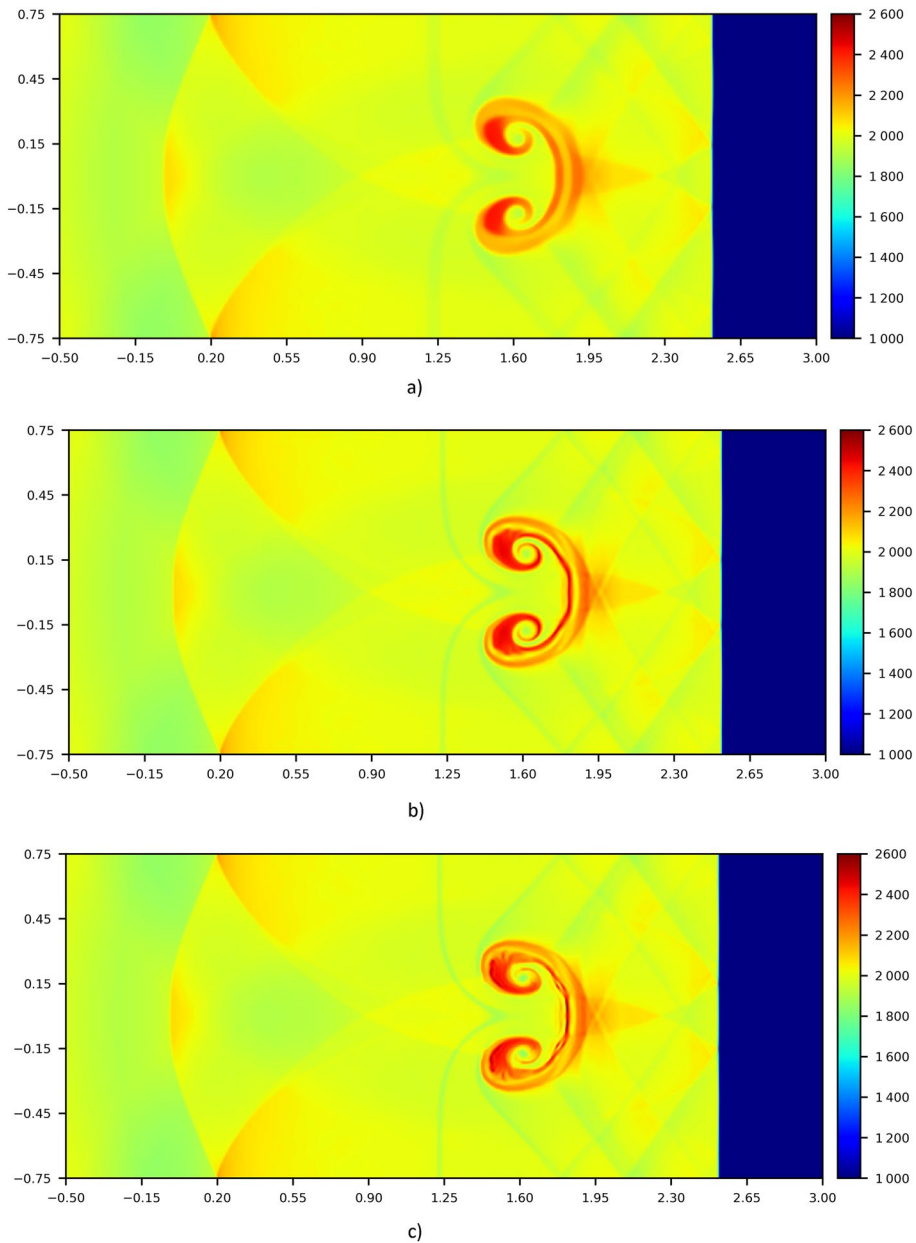
The problem is set up on a domain that spans  $[-0.5, 3.0] \times [-0.75, 0.75]$  and is run to a final time of 0.025 using the third-, fifth-, and seventh-order accurate HLL-based AFD WENO-AO schemes on a mesh with  $700 \times 300$  zones. Figure 16a shows the resulting solid density for the third-order accurate WENO-AO-3 scheme, Fig. 16b shows the resulting solid density for the fifth-order accurate WENO-AO-(5,3) scheme, and Fig. 16c shows the resulting solid density for the seventh-order accurate WENO-AO-(7,5,3) scheme. The ninth-order scheme shows similar results to seventh-order results; therefore, it is not shown here. We observe that the density profile obtained using the third-order scheme lacks sharp resolution. In contrast, the fifth- and seventh-order schemes effectively resolve all flow structures, thus demonstrating the significance of higher order schemes.

Next, we consider the Shock-Vortex interaction problem for the Baer-Nunziato model. A vortex with the mean velocity of the pre-shocked region propagates diagonally into a stationary shock, and as time increases, we see various rearranged structures in the vortex profile. The computational domain for this problem spans  $[-0.5, 1.5] \times [-0.5, 1.5]$ , and the problem was run to a final time of 0.84 using the fifth-order accurate HLL-based AFD WENO-AO-(5,3) scheme on a mesh with  $600 \times 600$  zones. Figures 17a–c show solid volume fractions at times  $t=0.0, 0.23, 0.84$ . The intermediate time of 0.23 corresponds to a time when the vortex has propagated halfway through the shock. Note that the shock does not reveal itself in this variable. In Figs. 17d–f, we show the solid  $x$ -velocity at times  $t=0.0, 0.23, 0.84$ . The shock is visible in this variable. We see that as the vortex propagates through the shock, the structure of the vortex is rearranged by the shock. However, the vortex is not fully destroyed by the shock. As a result, the vortex sheds some of its angular momentum after it has passed through the shock in an attempt to rearrange its structure. This is revealed by the spiral arms that are shed by the vortex in Figs. 17c, f. However, the figure shows us that vortices are very robust flow structures because they carry the angular momentum of the fluid. Because the angular momentum is conserved by the Baer-Nunziato flow, the vortices will not be fully destroyed as they pass through shocks of modest strength. This observation about vortices not being fully destroyed by shocks was previously established for the Euler flow in Balsara and Shu [11]. However, Fig. 17 provides further evidence that this insight also holds true for the Baer-Nunziato flow. The seventh- and ninth-order AFD-WENO schemes also show identical results, therefore, they are not shown here.

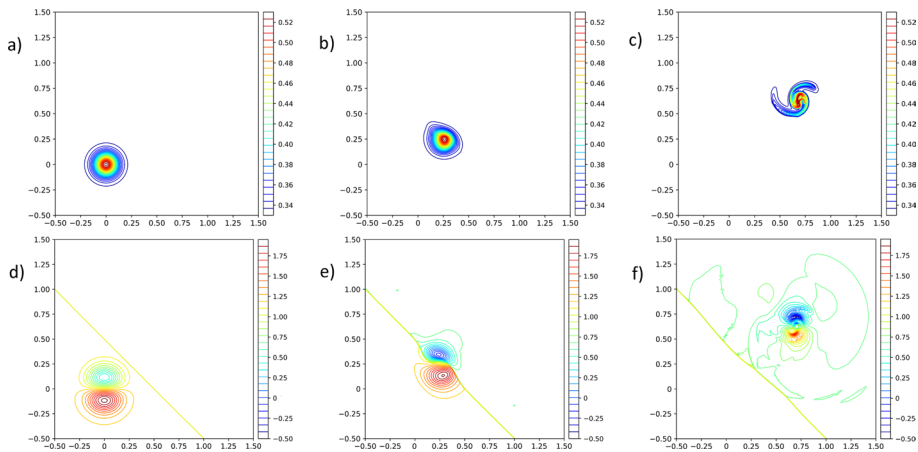
## 7.2 Two-Dimensional Test Problems for the Two-Layer Shallow Water Equations

For the two-layer shallow water equations described in Sect. 5.2, we present two multi-dimensional problems. The first problem is the shock-bubble interaction, and the second problem is the shock-vortex interaction problem. Both problems are presented in great details in Sects. 9.3 and 9.4 of Balsara et al. [5]. As the detailed setup is already given in Balsara et al. [5], we do not describe the setup here.

The first problem (shock-“bubble” interaction problem) consists of a planar, right-going shock propagating into an ambient medium which contains a bubble-like structure. The problem is set up on a domain that spans  $[-0.5, 3.0] \times [-0.75, 0.75]$  and is run to a final time of 0.3 using the third-, fifth-, and seventh-order HLL-based AFD-WENO-AO schemes on a mesh with  $700 \times 300$  zones. Figure 18a shows the resulting height of the lower fluid for the third-order accurate WENO-AO-3 scheme, Fig. 18b shows the resulting height of the lower fluid for the fifth-order accurate WENO-AO-(5,3) scheme, and Fig. 18c shows the resulting height of the lower fluid for the seventh-order accurate



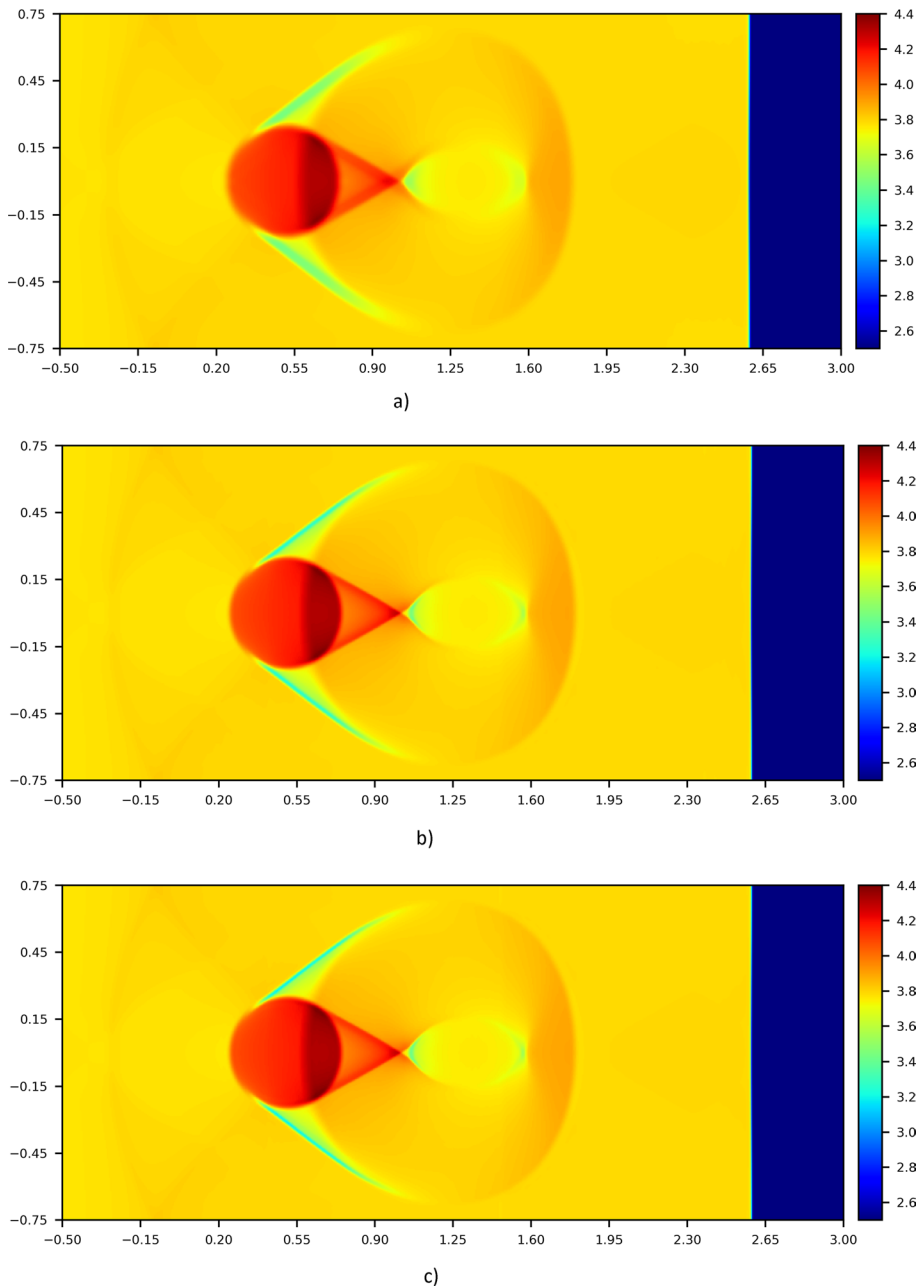
**Fig. 16** **a** Baer-Nunziato model: shock-bubble interaction problem using the third-order accurate HLL-based AFD WENO-AO-3 scheme with  $700 \times 300$  zones. The solid density profiles have been shown. **b** Baer-Nunziato model: shock-bubble interaction problem using the fifth-order accurate HLL-based AFD WENO-AO-(5,3) scheme with  $700 \times 300$  zones. The solid density profiles have been shown. **c** Baer-Nunziato model: shock-bubble interaction problem using the seventh order accurate HLL-based AFD WENO-AO-(7,5,3) scheme with  $700 \times 300$  zones. The solid density profiles have been shown. The ninth-order scheme also shows identical results and therefore it is not shown here



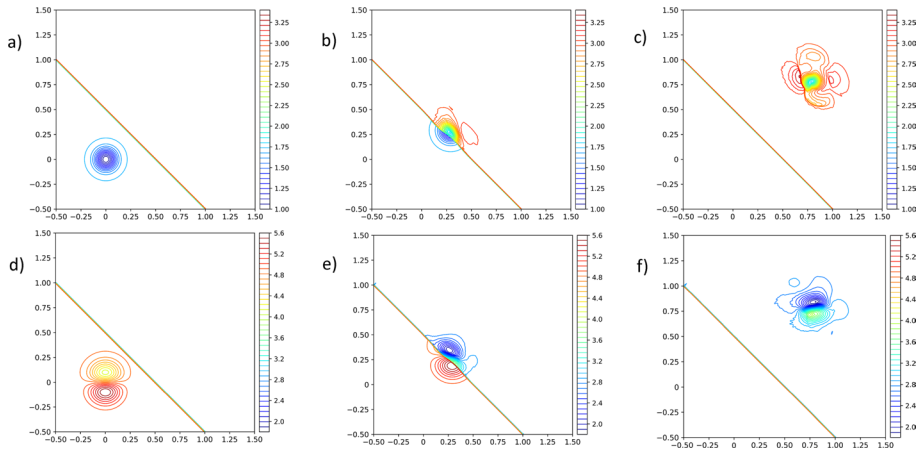
**Fig. 17** Baer-Nunziato: shock-vortex interaction using the fifth-order accurate HLL-based AFD WENO-AO-(5,3) scheme with  $600 \times 600$  zones at time levels  $t=0.0, 0.23$ , and  $0.84$ . **a–c** show the solid volume fraction at times  $t=0.0, 0.23, 0.84$ . **d–f** show the solid  $x$ -velocity at times  $t=0.0, 0.23, 0.84$ . For the solid volume fraction, 30 contours were fit between a range of  $0.33$  and  $0.530$ . For the solid  $x$ -velocity, 30 contours were fit between a range of  $-0.5$  and  $1.95$

WENO-AO-(7,5,3) scheme. The shock propagates into the unshocked medium without perturbation till it encounters the depression. When it encounters the “bubble” it sets up a left-going bow shock and a wake around the bubble (shown in green color in Fig. 18). A very interesting flow structure develops behind the “bubble” as can be seen in Fig. 18. The obtained results match with the reported results in Balsara et al. [5]. In Fig. 18c, we observe that the seventh-order accurate schemes capture the shock front (shown in red) more accurately than the fifth-order accurate results (Fig. 18b). The ninth-order scheme shows the identical result to seventh-order results; therefore, it is not shown here. In conclusion, the higher order simulations show flow structures that are very crisp, highlighting the value of higher order schemes.

Next, we consider the Shock-Vortex interaction problem for the Two-Layer Shallow water equations. The computational domain for this problem spans  $[-0.5, 1.5] \times [-0.5, 1.5]$  and the problem was run to a final time of  $0.24$  using the seventh-order accurate HLL-based AFD WENO-AO-(7,3) scheme on a mesh with  $600 \times 600$  zones. Figures 19a–c show the height of the upper fluid at times  $t=0.0, 0.06, 0.24$ . The intermediate time of  $0.06$  corresponds to a time when the vortex has propagated halfway through the shock and a little bend in the shock front can be observed at this time. Figures 19d–f show the  $x$ -velocity of the upper fluid at times  $t=0.0, 0.06, 0.24$ . Similar to Figs. 19a–c for the height of the upper fluid, a stationary shock is clearly visible in Figs. 19d–f for the  $x$ -velocity of the upper fluid. We observe that as the vortex propagates through the standing shock, the vortex is not fully destroyed by the shock flow. Instead, the vortex sheds some of its angular momentum after it has passed through the shock. In doing so, the vortex has rearranged its form. Analogous to the Baer-Nunziato flow equations, the two-layer shallow water equations preserve the angular momentum. As a result, the core of the vortex is not destroyed as it passes through the mild shock. The obtained results are consistent with the results given in Balsara et al. [5]. The fifth- and ninth-order AFD-WENO schemes also show identical results, therefore, they are not shown here.



**Fig. 18** **a** Two-layer shallow water: shock-bubble interaction problem using the third-order accurate HLL-based AFD WENO-AO-3 scheme with  $700 \times 300$  zones. The solid density profiles have been shown. **b** Two-layer shallow water: shock-bubble interaction problem using the fifth-order accurate HLL-based AFD WENO-AO-(5,3) scheme with  $700 \times 300$  zones. The solid density profiles have been shown. **c** Two-layer shallow water: shock-bubble interaction problem using the seventh-order accurate HLL-based AFD WENO-AO-(7,5,3) scheme with  $700 \times 300$  zones. The solid density profiles have been shown. The ninth-order scheme also shows identical results and, therefore, it is not shown here



**Fig. 19** Two-layer shallow water: shock-vortex interaction using the seventh-order accurate HLL-based AFD WENO-AO-(7,3) scheme with  $600 \times 600$  zones at time levels  $t=0.0, 0.06, 0.24$ . **a–c** Height of the upper fluid at times  $t=0.0, 0.06, 0.24$ . **d–f** show the  $x$ -velocity of the upper fluid at times  $t=0.0, 0.06, 0.24$ . For the height, 40 contours were fit between a range of 1.0 and 3.4. For the velocity, 40 contours were fit between a range of 1.8 and 5.6

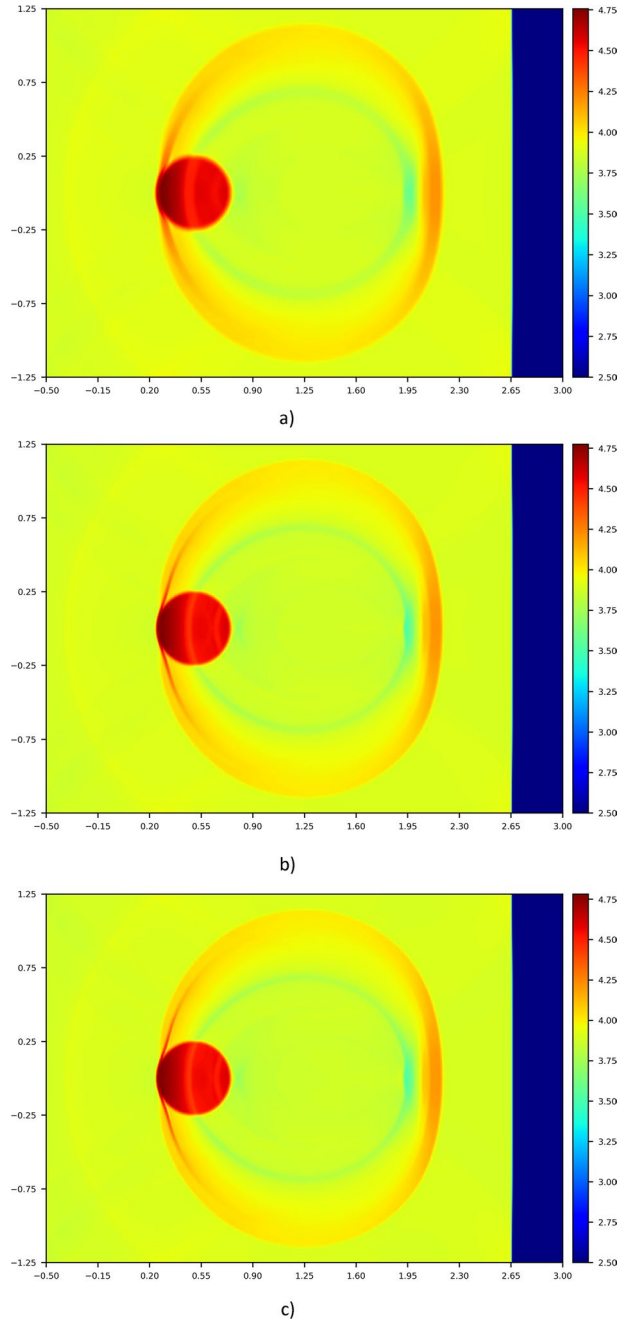
### 7.3 Two-Dimensional Test Problems for the Multiphase Debris Flow

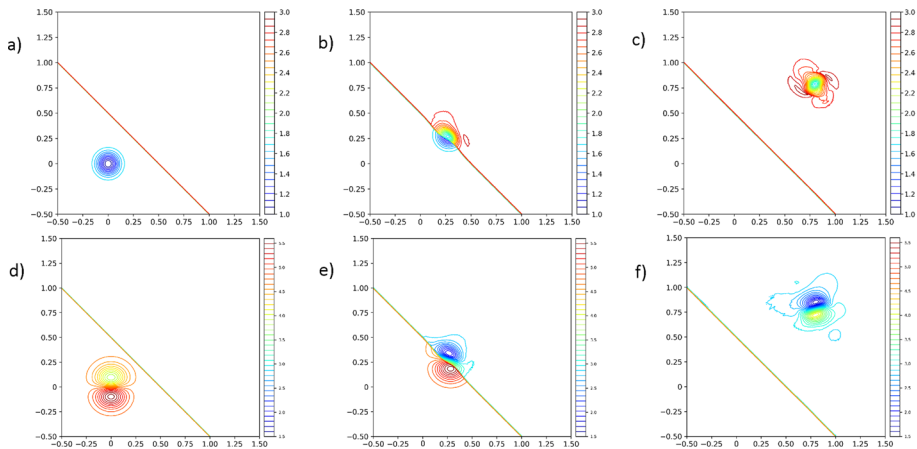
In this section, we present two multi-dimensional problems for the multiphase debris flow. Both problems are described in Balsara et al. [5]. As the detailed setup is already given in Balsara et al. [5], we do not repeat the details here.

The first problem is the shock-“bubble” interaction problem. The problem utilizes the Vortex used in Sect. 5.3. Here we consider a domain that spans  $[-0.5, 3.0] \times [-1.25, 1.25]$ . The problem was run to a final time of 0.3 using the third-, fifth-, and seventh-order HLL-based AFD-WENO-AO schemes on a mesh with  $700 \times 500$  zones. Figure 20a shows the final image of the height of the fluid phase for the third-order accurate WENO-AO-3 scheme, Fig. 20b shows the resulting height of the fluid phase for the fifth-order accurate WENO-AO-(5,3) scheme, and Fig. 20c shows the resulting height of the fluid phase for the seventh-order accurate WENO-AO-(7,5,3) scheme. When the shock encounters the “bubble”, it creates a left-going bow shock. A ring-shaped structure originating from the area of the depression can be observed in Fig. 20. The ninth-order scheme shows the identical result to seventh-order results; therefore, it is not shown here. Once again, we observe that the higher order simulations (fifth and seventh orders) exhibit flow structures that are highly sharp. This emphasizes the significance and advantages of employing higher order schemes.

Next, we consider the shock-vortex interaction problem for the multiphase debris flow model. The computational domain for this problem spans  $[-0.5, 1.5] \times [-0.5, 1.5]$ . The problem was run to a final time of 0.24 using the ninth-order accurate HLL-based AFD WENO-AO-(9,3) scheme on a mesh with  $600 \times 600$  zones. Figures 21a–c show the height of the upper fluid at times  $t=0.0, 0.06, 0.24$ . The intermediate simulation time of 0.06 corresponds to a time when the vortex has propagated halfway through the shock. At this time, a slight reduction in the strength of the solid height can be observed in the vortex. Figures 21d–f show the solid  $x$ -velocity at times  $t=0.0, 0.06, 0.24$ . The standing shock is visible in all of the panels of Fig. 21. The vortex sheds some of its angular momentum

**Fig. 20** **a** Debris flow: shock-bubble interaction problem using the third-order accurate HLL-based AFD WENO-AO-3 scheme with  $700 \times 500$  zones. The solid density profiles have been shown. **b** Debris flow: shock-bubble interaction problem using the fifth-order accurate HLL-based AFD WENO-AO-(5,3) scheme with  $700 \times 500$  zones. The solid density profiles have been shown. **c** Debris flow: shock-bubble interaction problem using the seventh-order accurate HLL-based AFD WENO-AO-(7,5,3) scheme with  $700 \times 500$  zones. The solid density profiles have been shown. The ninth-order accurate scheme also shows identical results and therefore it is not shown here





**Fig. 21** Debris flow: shock-vortex interaction using the ninth-order accurate HLL-based AFD WENO-AO-(9,3) scheme with  $600 \times 600$  zones at time levels  $t=0.0, 0.06$ , and  $0.24$ . **a–c** show the solid height at times  $t=0.0, 0.06, 0.24$ . **d–f** show the solid  $x$ -velocity at times  $t=0.0, 0.06, 0.24$ . For the height, 30 contours were fit between a range of 1.0 and 3.0. For the velocity, 30 contours were fit between a range of 1.5 and 5.6

**Table 9** Left states, right states, and final times of the one-dimensional Riemann problem for the Baer-Nunziato compressible multi-phase flow with stiff source terms

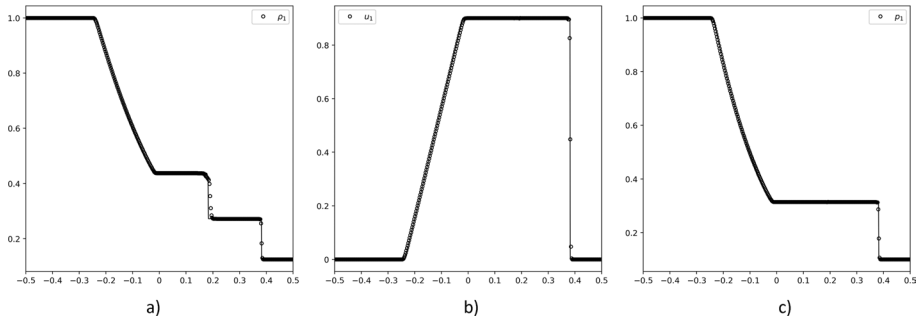
	$\rho_1$	$u_1$	$p_1$	$\rho_2$	$u_2$	$p_2$	$\phi_1$	$t_{\text{end}}$
	$\gamma_1 = 1.4, \pi_1 = 0, \gamma_2 = 1.67, \pi_2 = 0$							
<b>Left:</b>	1.0	0.0	1.0	1.0	0.0	1.0	0.99	0.2
<b>Right:</b>	0.125	0.0	0.1	0.125	0.0	0.1	0.01	

after it has passed through the mild shock. In doing so, the vortex has rearranged its form. The obtained results are consistent with the results given in Balsara et al. [5]. The fifth- and seventh-order AFD-WENO schemes also show identical results, therefore, they are not shown here.

## 8 Test Problems with Stiff Source Terms

AFD-WENO schemes demonstrate a significant advantage when dealing with stiff source terms. Unlike finite volume methods, these schemes operate pointwise, which includes the zone centers where the primal variables are located. Consequently, the enforcement of source terms aligns with the collocation of variables, leading to enhanced accuracy and low computation costs. We present two test problems that are based on the Baer-Nunziato compressible multi-phase flow with stiff source terms, as given in Dumbser and Boscheri [20]. The same set of problems was also used in Balsara et al. [5]. For both the problems, we use a third-order accurate IMEX-SSP3(4,3,3) scheme from Pareschi and Russo [35] for the time integration.

The equations for the Baer-Nunziato compressible multi-phase flow with stiff source terms are given by



**Fig. 22** Baer-Nunziato with stiff source: results for the one-dimensional Riemann Problem using the fifth-order accurate HLL-based AFD WENO-AO-(5,3) scheme with 400 zones. **a–c** show the solid density, solid  $x$ -velocity, and solid pressure

$$\begin{aligned}
 \partial_t(\phi_1 \rho_1) + \nabla \cdot (\phi_1 \rho_1 \mathbf{v}_1) &= 0, \\
 \partial_t(\phi_1 \rho_1 \mathbf{v}_1) + \nabla \cdot (\phi_1 (\rho_1 \mathbf{v}_1 \otimes \mathbf{v}_1 + \mathbf{I} p_1)) - P_I \nabla \phi_1 &= -\lambda(\mathbf{v}_1 - \mathbf{v}_2), \\
 \partial_t(\phi_1 \rho_1 E_1) + \nabla \cdot (\phi_1 \mathbf{v}_1 (\rho_1 E_1 + p_1)) + P_I \partial_t \phi_1 &= -\lambda \mathbf{V}_I \cdot (\mathbf{v}_1 - \mathbf{v}_2), \\
 \partial_t(\phi_2 \rho_2) + \nabla \cdot (\phi_2 \rho_2 \mathbf{v}_2) &= 0, \\
 \partial_t(\phi_2 \rho_2 \mathbf{v}_2) + \nabla \cdot (\phi_2 (\rho_2 \mathbf{v}_2 \otimes \mathbf{v}_2 + \mathbf{I} p_2)) - P_I \nabla \phi_2 &= -\lambda(\mathbf{v}_2 - \mathbf{v}_1), \\
 \partial_t(\phi_2 \rho_2 E_2) + \nabla \cdot (\phi_2 \mathbf{v}_2 (\rho_2 E_2 + p_2)) + P_I \partial_t \phi_2 &= -\lambda \mathbf{V}_I \cdot (\mathbf{v}_2 - \mathbf{v}_1), \\
 \partial_t \phi_1 + \mathbf{V}_I \cdot \nabla \phi_1 &= \mu(p_1 - p_2).
 \end{aligned}$$

The set of equations mentioned above contains stiff terms on the right-hand side, and the stiffness is controlled by two parameters, denoted as  $\lambda$  and  $\mu$ .

The first test problem is a one-dimensional problem that spans the domain  $[-0.5, 0.5]$ . The left and right initial conditions were initialized around  $x = 0$  and are specified in Table 9. The problem uses interphase drag  $\lambda = 10^3$  and pressure relaxation parameter  $\mu = 10^2$ . These values turn this problem into a test problem with a moderately stiff source term. The simulation was run till time  $t=0.2$  on a 400 zone mesh with a CFL of 0.8 using the fifth-order accurate HLL-based AFD WENO-AO-(5,3) scheme. Figures 22a–c show the solid density, solid  $x$ -velocity, and solid pressure profiles, respectively. We see that the results are comparable to those presented in Dumbser and Boscheri [20]. The seventh- and ninth-order AFD-WENO schemes also show identical results, therefore, they are not shown here. The reference solution is also shown as the solid line in Fig. 22 and was obtained using a third-order scheme with 4 000 zones.

The second test problem is a two-dimensional Riemann problem. The computational domain spans  $[-0.5, 0.5] \times [-0.5, 0.5]$ . The initial states are described in Table 10. The problem was run with a CFL of 0.4 to a stopping time of 0.15 on a mesh of  $400 \times 400$  zones. Here we have used  $\lambda = 10^5$  and  $\mu = 10^2$  which corresponds to rather severely stiff source terms. We run the simulation using the seventh-order accurate HLL-based AFD WENO-AO-(7,3) scheme. Figure 23a shows the solid density, Fig. 23b shows the gas density, and Fig. 23c shows the solid volume fraction. We observe that the obtained results are consistent with the solution presented in Dumbser and Boscheri [20]. The fifth- and ninth-order AFD-WENO schemes also show identical results, therefore, they are not shown here.

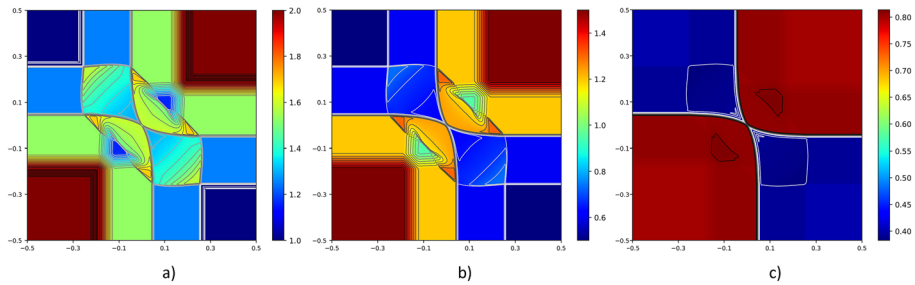
**Table 10** The initial states of the two-dimensional Riemann problem for the Baer-Nunziato compressible multi-phase flow with stiff source terms

$x, y$	$\rho_1$	$u_1$	$v_1$	$p_1$	$\rho_2$	$u_2$	$v_2$	$p_2$	$\phi_1$
	$\gamma_1 = 1.4, \pi_1 = 0, \gamma_2 = 1.67, \pi_2 = 0$								
$x > 0, y > 0$	2.0	0.0	0.0	2.0	1.5	0.0	0.0	2.0	0.8
$x < 0, y > 0$	1.0	0.0	0.0	1.0	0.5	0.0	0.0	1.0	0.4
$x < 0, y < 0$	2.0	0.0	0.0	2.0	1.5	0.0	0.0	2.0	0.8
$x > 0, y < 0$	1.0	0.0	0.0	1.0	0.5	0.0	0.0	1.0	0.4

## 9 Conclusions

Classical FD-WENO schemes have been available for conservation laws since the early papers by Shu and Osher [43, 44], Jiang and Shu [30], and Balsara and Shu [11]. Until very recently, all variants of the finite difference WENO scheme have indeed been restricted to treating only hyperbolic systems that are in the conservation form. The recent emergence of several classes of hyperbolic systems with non-conservative products exposes a dire need for a new class of finite difference WENO schemes that can handle such systems. This has called for a reassessment of the entire design philosophy of finite difference WENO schemes. The first step in such a redesign took place very recently in Balsara et al. [5] where a very efficient set of finite difference WENO schemes was designed that could integrate hyperbolic systems with non-conservative products. The schemes in Balsara et al. [5] were demonstrated to work well for up to ninth order of accuracy. They can truly be thought of as being the analogues of the schemes in Shu and Osher [43, 44] when the hyperbolic system has non-conservative products. The schemes in Balsara et al. [5] were indeed significantly faster than the classical FD-WENO schemes. However, they were written in a fluctuation form which does not simply reduce to a flux conservative form when the system is indeed conservative. The present AFD-WENO schemes are very innovative, and to the best of our knowledge have never been presented in the literature. They are designed to take the field further so that finite difference WENO methods can handle hyperbolic systems with non-conservative products while at the same time respecting the Lax-Wendroff theorem when it is applicable.

The present schemes derive their start from alternative finite difference WENO (AFD-WENO) schemes for conservation laws. A well-developed AFD-WENO scheme offers many extremely desirable advantages over a classical FD-WENO scheme. First, it can work with any type of Riemann solver and different fields of study may have their own special Riemann solver that they find beneficial. Such flexibility is not available in FD-WENO formulations, whereas it is indeed available in AFD-WENO formulations. This flexibility of invoking the Riemann solver at pointwise locations can also be exploited to ensure that the final scheme respects the preservation of free stream conditions on curvilinear meshes. On such curvilinear meshes, the flux reconstruction of classical FD-WENO methods becomes a liability; this enables AFD-WENO methods to come to the fore because they are not restricted by this liability. The stumbling block in the development of AFD-WENO schemes was that general-purpose AFD-WENO schemes that can work with minimal or no changes for large classes of hyperbolic conservation laws were not in hand. This



**Fig. 23** Baer-Nunziato with stiff source: results for the two-dimensional Riemann problem using the seventh-order accurate HLL-based AFD WENO-AO-(7,3) scheme with  $400 \times 400$  zones. **a** shows the solid density, **b** shows the gas density, and **c** shows the solid volume fraction. Thirty equidistant contour lines are shown over the color plots

barrier was removed in Balsara et al. [6] where a general-purpose AFD-WENO scheme was designed for up to ninth order and shown to work well for large classes of hyperbolic conservation laws. The previously mentioned paper set the stage for the work reported in this paper.

If an AFD-WENO scheme could be general-purpose, available at several high orders and work well for large classes of conservation laws, then one is inclined to ask the following question: why is a general-purpose, high order AFD-WENO scheme not available for large classes of hyperbolic systems that have non-conservative products? This question is the very important motivating question that animates this entire paper. In Sect. 2, we show that there is a way to obtain an answer to this very important question. We start with a general flux which we write as having a conservative part and a part which will only be thought of as leading to a non-conservative product. This allows us to write the AFD-WENO update equation in a fluctuation form. Once this door is opened, we carefully track which steps retain flux conservation and which steps in the derivation force us to relinquish flux conservation. The end result is shown in Eq. (15) which presents a general-purpose, high order AFD-WENO scheme that is well-suited for large classes of hyperbolic systems that have non-conservative products. The scheme we derive in Eq. (15) is one which retains a flux conservative update for those components of the PDE that are indeed in the conservation form and judiciously relinquishes it for those components of the PDE that are dominated by non-conservative products. In retaining the conservation form as much as possible, the dictates of the Lax-Wendroff theorem are also preserved as much as possible. Two variants are designed, the first one being documented in Eq. (15) and the second one in Eq. (13). Equation (15) is based on making three WENO interpolation steps; where the first WENO interpolation is done in characteristic space and the next two WENO interpolations are lower cost and can be done component by component. Equation (13) is based on making two WENO interpolation steps; where the first WENO interpolation is done in characteristic space and the next WENO interpolation is lower cost and can be done component by component.

With the formulation in hand, Sect. 3 addresses some issues associated with the non-linear hybridization and stabilization of the higher order derivatives. Section 4 provides point by point details on how the method is implemented.

Section 5 shows accuracy tests. It shows that for three very broad classes of hyperbolic systems with non-conservative products—i.e., the Baer-Nunziato system, the two-layer shallow water system, and the multiphase debris flow model of Pitman and Le—the

method achieves its design accuracies. Equally importantly, in Sect. 5.4 we show a speed comparison of the classical FD-WENO scheme, the WENO scheme from Balsara et al. [5], the AFD-WENO scheme from Balsara et al. [6] which only applies to conservation laws, and the two AFD-WENO schemes from this paper which can be applied to any hyperbolic system with fluxes as well as non-conservative products. The speeds of the schemes conform to our expectations and all schemes are shown to be roughly competitive in performance. The differences in speeds are explained thoroughly in Sect. 5.4. In that section we also provide a discussion on the implementation of finite difference WENO schemes on modern-day CPUs with large caches. We show a very desirable result that higher order variants of finite difference WENO schemes do not cost that much more than their lower order variants. The scientific explanation for that stems from our realization that the larger number of floating point operations associated with larger stencils are almost free of charge if the finite difference WENO code is designed to be cache friendly. This should have great, and very beneficial, implications for the role of finite difference WENO schemes in Peta- and Exascale computing! Simply put: our well-implemented AFD-WENO schemes come pretty close to the promised land of high performance computing where float point operations are almost free and the dominant costs are the other fixed costs in the implementation!

Section 6 shows several stringent one-dimensional test problems drawn from our three PDE systems of interest. Section 7 shows multidimensional test problems, again drawn from our three PDE systems of interest. Section 8 shows test problems with stiff source terms, demonstrating the very favorable handling and features of AFD-WENO in the stiff limit.

## Appendix A

The following variant of Eq. (15) may also be used:

$$\begin{aligned} \partial_t \mathbf{U}_i = & -\frac{1}{\Delta x} \left\{ \mathbf{D}_{i+1/2}^* (\hat{\mathbf{U}}_{i+1/2}^-, \hat{\mathbf{U}}_{i+1/2}^+) + \mathbf{D}_{i-1/2}^* (\hat{\mathbf{U}}_{i-1/2}^-, \hat{\mathbf{U}}_{i-1/2}^+) \right\} - \frac{1}{\Delta x} \left\{ \mathbf{F}(\hat{\mathbf{U}}_{i+1/2}^-) - \mathbf{F}(\hat{\mathbf{U}}_{i-1/2}^+) \right\} \\ & - \frac{1}{\Delta x} \mathbf{C}(\mathbf{U}_i) (\hat{\mathbf{U}}_{i+1/2}^- - \hat{\mathbf{U}}_{i-1/2}^+) \\ & - \frac{1}{\Delta x} \mathbf{C}(\mathbf{U}_i) \left\{ \begin{aligned} & \left[ -\frac{1}{24} (\Delta x)^2 [\partial_x^2 \hat{\mathbf{U}}]_{i+1/2} + \frac{7}{5760} (\Delta x)^4 [\partial_x^4 \hat{\mathbf{U}}]_{i+1/2} \right. \\ & \left. - \frac{31}{967680} (\Delta x)^6 [\partial_x^6 \hat{\mathbf{U}}]_{i+1/2} + \frac{127}{154828800} (\Delta x)^8 [\partial_x^8 \hat{\mathbf{U}}]_{i+1/2} \right] \\ & \left[ -\frac{1}{24} (\Delta x)^2 [\partial_x^2 \hat{\mathbf{U}}]_{i-1/2} + \frac{7}{5760} (\Delta x)^4 [\partial_x^4 \hat{\mathbf{U}}]_{i-1/2} \right. \\ & \left. - \frac{31}{967680} (\Delta x)^6 [\partial_x^6 \hat{\mathbf{U}}]_{i-1/2} + \frac{127}{154828800} (\Delta x)^8 [\partial_x^8 \hat{\mathbf{U}}]_{i-1/2} \right] \end{aligned} \right\} \\ & - \frac{1}{\Delta x} \left\{ \begin{aligned} & \left[ -\frac{1}{24} (\Delta x)^2 [\partial_x (\mathbf{B} \partial_x \hat{\mathbf{U}})]_{i+1/2} + \frac{7}{5760} (\Delta x)^4 [\partial_x^3 (\mathbf{B} \partial_x \hat{\mathbf{U}})]_{i+1/2} \right. \\ & \left. - \frac{31}{967680} (\Delta x)^6 [\partial_x^5 (\mathbf{B} \partial_x \hat{\mathbf{U}})]_{i+1/2} + \frac{127}{154828800} (\Delta x)^8 [\partial_x^7 (\mathbf{B} \partial_x \hat{\mathbf{U}})]_{i+1/2} \right] \\ & \left[ -\frac{1}{24} (\Delta x)^2 [\partial_x (\mathbf{B} \partial_x \hat{\mathbf{U}})]_{i-1/2} + \frac{7}{5760} (\Delta x)^4 [\partial_x^3 (\mathbf{B} \partial_x \hat{\mathbf{U}})]_{i-1/2} \right. \\ & \left. - \frac{31}{967680} (\Delta x)^6 [\partial_x^5 (\mathbf{B} \partial_x \hat{\mathbf{U}})]_{i-1/2} + \frac{127}{154828800} (\Delta x)^8 [\partial_x^7 (\mathbf{B} \partial_x \hat{\mathbf{U}})]_{i-1/2} \right] \end{aligned} \right\}. \end{aligned} \quad (\text{A1})$$

It slightly reduces the cost of the interpolation that is needed in the last curly bracket. But it requires the evaluation of the matrix multiplication  $(\mathbf{B}\partial_x\hat{\mathbf{U}})_i$  at each zone center “ $i$ ” to interpolate its derivatives to the zone boundaries.

## Appendix B Phase-Volume-Based Flattener Function for the Baer-Nunziato Model

Flattening algorithms are used in simulations to detect the location of the strong shocks within the computational domain. Their purpose is to enhance the accuracy of the simulation by reducing numerical artifacts near discontinuities. For the Euler flow, a flattening function that is based on the divergence of the velocity field and characteristic speed has been presented in Colella and Woodward [17], and Balsara [4] and it can also be used here. However, we have found that a sudden change in the volume fraction can also be a source of spurious oscillations for the Baer-Nunziato system. Therefore, we propose a one-dimensional phase-volume based flattener function for the Baer-Nunziato model. The flattener algorithm trivially extends to the two- and three-dimensional cases. The method begins by computing the sum and difference of extremal values (denoted by  $\Delta_i^+\phi$  and  $\Delta_i^-\phi$ , respectively) of the solid volume fractions in cell  $i$  over the neighbouring cells  $i-1$ ,  $i$ , and  $i+1$ . Mathematically, we define the quantities  $\Delta_i^+\phi$  and  $\Delta_i^-\phi$  as follows:

$$\begin{aligned}\Delta_i^+\phi &= \max[\phi_1^{i-1}, \phi_1^i, \phi_1^{i+1}] + \min[\phi_1^{i-1}, \phi_1^i, \phi_1^{i+1}], \\ \Delta_i^-\phi &= \max[\phi_1^{i-1}, \phi_1^i, \phi_1^{i+1}] - \min[\phi_1^{i-1}, \phi_1^i, \phi_1^{i+1}],\end{aligned}$$

where  $\phi_1^i$  is the solid volume fraction within the cell  $i$ . In each zone  $i$ , the flattener function (denoted by  $\eta_i$ ) is defined as

$$\eta_i = \min[1, \max[0, 0.5(a_i - \kappa)]] ,$$

where  $\kappa$  is a positive constant number which works best for us when we choose  $\kappa = 0.1$ . The expression for the quantity  $a_i$  is given by

$$a_i = \max\left[\frac{|\Delta_i^-\phi|}{0.5\Delta_i^+\phi + \epsilon}, \frac{|\Delta_i^-\phi|}{|1 - 0.5\Delta_i^+\phi| + \epsilon}\right], \quad \text{where } \epsilon = 10^{-12} \text{ is a small number.}$$

The flattener function does not modify the reconstruction/interpolation when the flow is smooth or consists of rarefactions, and in that case,  $\eta_i = 0$ . However, the flattener function gradually increases from  $\eta_i = 0$  to  $\eta_i = 1$  when strong shocks are present in the volume fraction.

**Acknowledgements** DSB acknowledges support via NSF grant NSF-AST-2009776, NASA grant NASA-2020-1241 and NASA grant 80NSSC22K0628. DSB and HK acknowledge support from a Vajra award, VJR/2018/00129 and also a travel grant from Notre Dame International. CWS acknowledges support via NSF grant DMS-2309249.

**Funding** The funding has been acknowledged. DSB acknowledges support via the NSF Grants NSF-19-04774, NSF-AST-2009776, NASA-2020-1241, and NASA-80NSSC22K0628. DSB and HK acknowledge support from a Vajra award, VJR/2018/00129. CWS acknowledges support via AFOSR Grant FA9550-20-1-0055 and NSF Grant DMS-2010107.

**Data Availability** All data that were used in the generation of the figures have been stored and available for later use.

## Compliance with Ethical Standards

**Conflict of Interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

**Ethical Approval** Not applicable.

**Informed Consent** Not applicable.

## References

1. Andrianov, N., Warnecke, G.: The Riemann problem for the Baer-Nunziato two-phase flow model. *J. Comput. Phys.* **212**, 434–464 (2004)
2. Baer, M.R., Nunziato, J.W.: A two-phase mixture theory for the deflagration-to-detonation transition (DDT) in reactive granular materials. *Int. J. Multiph. Flow* **12**, 861–889 (1986)
3. Balsara, D.S.: Total variation diminishing algorithm for adiabatic and isothermal magnetohydrodynamics. *Ap. J. Supp.* **116**, 133–153 (1998)
4. Balsara, D.S.: Self-adjusting, positivity preserving high order schemes for hydrodynamics and magnetohydrodynamics. *J. Comput. Phys.* **231**, 7504–7517 (2012)
5. Balsara, D.S., Bhowmik, D., Shu, C.-W., Kumar, H.: Efficient finite difference WENO scheme for hyperbolic systems with non-conservative products. *Commun. Appl. Math. Comput.* (2023). <https://doi.org/10.1007/s42967-023-00275-9>
6. Balsara, D.S., Bhowmik, D., Shu, C.-W., Kumar, H.: Efficient alternative finite difference WENO schemes for hyperbolic conservation laws. (2024). [arxiv:2403.01264v1](https://arxiv.org/abs/2403.01264)
7. Balsara, D.S., Garain, S., Florinski, V., Boscheri, W.: An efficient class of WENO schemes with adaptive order for unstructured meshes. *J. Comput. Phys.* **404**, 109062 (2020)
8. Balsara, D.S., Garain, S., Shu, C.-W.: An efficient class of WENO schemes with adaptive order. *J. Comput. Phys.* **326**, 780–804 (2016)
9. Balsara, D.S., Rumpf, T., Dumbser, M., Munz, C.-D.: Efficient, high-accuracy ADER-WENO schemes for hydrodynamics and divergence-free magnetohydrodynamics. *J. Comput. Phys.* **228**, 2480 (2009)
10. Balsara, D.S., Samantary, S., Subramanian, S.: Efficient WENO-based prolongation strategies for divergence-preserving vector fields. *Commun. Appl. Math. Comput.* **5**(1), 428–484 (2023)
11. Balsara, D.S., Shu, C.-W.: Monotonicity preserving weighted non-oscillatory schemes with increasingly high order of accuracy. *J. Comput. Phys.* **160**, 405–452 (2000)
12. Borges, R., Carmona, M., Costa, B., Don, W.S.: An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. *J. Comput. Phys.* **227**(6), 3101–3211 (2008)
13. Castro, M., Costa, B., Don, W.S.: High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws. *J. Comput. Phys.* **230**, 1766–1792 (2011)
14. Castro, M., Gallardo, J.M., Parés, C.: High order finite volume schemes based on reconstruction of states for solving hyperbolic systems with nonconservative products. Applications to shallow-water systems. *Math. Comput.* **75**(255), 1103–1134 (2006)
15. Castro, M.J., Pardo, A., Parés, C., Toro, E.F.: On some fast well-balanced first order solvers for non-conservative systems. *Math. Comput.* **79**, 1427–1472 (2010)
16. Chiochetti, S., Müller, C.: A solver for stiff finite-rate relaxation in Baer-Nunziato two-phase flow models. (2020). [arxiv: 2002.09381](https://arxiv.org/abs/2002.09381)
17. Colella, P., Woodward, P.R.: The piecewise parabolic method (PPM) for gas-dynamical simulations. *J. Comput. Phys.* **54**(1), 174–201 (1984)
18. Coquel, F., Marmignon, C., Rai, P., Renac, F.: An entropy stable high-order discontinuous Galerkin spectral element method for the Baer-Nunziato two-phase flow model. *J. Comput. Phys.* **431**, 110135 (2021)
19. Dumbser, M., Balsara, D.S.: A new efficient formulation of the HLLEM Riemann solver for general conservative and non-conservative hyperbolic systems. *J. Comput. Phys.* **304**, 275–319 (2016)

20. Dumbser, M., Boscheri, W.: High-order unstructured Lagrangian one-step WENO finite volume schemes for non-conservative hyperbolic systems: applications to compressible multi-phase flows. *Comput. Fluids* **86**, 405–432 (2013)
21. Dumbser, M., Castro, M., Parés, C., Toro, E.F.: ADER schemes on unstructured meshes for non-conservative hyperbolic systems: applications to geophysical flows. *Comput. Fluids* **38**, 1731–1748 (2009)
22. Dumbser, M., Hidalgo, A., Castro, M., Parés, C., Toro, E.F.: FORCE schemes on unstructured meshes II: non-conservative hyperbolic systems. *Comput. Methods Appl. Mech. Engrg.* **199**, 625–647 (2010)
23. Dumbser, M., Hidalgo, A., Zanotti, O.: High-order space-time adaptive ADER-WENO finite volume schemes for non-conservative hyperbolic systems. *Comput. Methods Appl. Mech. Engrg.* **268**, 359–387 (2014)
24. Dumbser, M., Iben, U., Ioriatti, M.: An efficient semi-implicit finite volume method for axially symmetric compressible flows in compliant tubes. *Appl. Numer. Math.* **89**, 24–44 (2015)
25. Gao, Z., Fang, L.-L., Wang, B.-S., Wang, Y., Don, W.S.: Seventh and ninth orders characteristic-wise alternative WENO finite difference schemes for hyperbolic conservation laws. *Comput. Fluids* **202**, 104519 (2020)
26. Gerolymos, G.A., Sénéchal, D., Vallet, I.: Very high order WENO schemes. *J. Comput. Phys.* **228**, 8481–8524 (2009)
27. Harten, A., Engquist, B., Osher, S., Chakravarthy, S.: Uniformly high order essentially non-oscillatory schemes III. *J. Comput. Phys.* **71**, 231–303 (1987)
28. Henrick, A.K., Aslam, T.D., Powers, J.M.: Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points. *J. Comput. Phys.* **207**, 542–567 (2006)
29. Jiang, G.-S., Shu, C.-W.: Efficient implementation of weighted ENO schemes. *J. Comput. Phys.* **126**, 202–228 (1996)
30. Jiang, Y., Shu, C.-W., Zhang, M.: An alternative formulation of finite difference ENO schemes with Lax-Wendroff time discretization for conservation laws. *SIAM J. Sci. Comput.* **35**(2), A1137–A1160 (2013)
31. Jiang, Y., Shu, C.-W., Zhang, M.: Free-stream preserving finite-difference schemes on curvilinear meshes. *Methods Appl. Anal.* **21**(1), 001–030 (2014)
32. Kupka, F., Happenhofer, N., Higuera, I., Koch, O.: Total-variation-diminishing implicit-explicit Runge-Kutta methods for the simulation of double-diffusive convection in astrophysics. *J. Comput. Phys.* **231**, 3561–3586 (2012)
33. Liu, X.-D., Osher, S., Chan, T.: Weighted essentially non-oscillatory schemes. *J. Comput. Phys.* **115**, 200–212 (1994)
34. Merriman, B.: Understanding the Shu-Osher conservative finite difference form. *J. Sci. Comput.* **19**(1/2/3), 309 (2003)
35. Pareschi, L., Russo, G.: Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation. *J. Sci. Comput.* **25**, 129 (2005)
36. Pelanti, M., Bouchut, F., Mangeney, A.: A Roe-type scheme for two-phase shallow granular flows over variable topography. *Math. Model. Numer. Anal.* **42**, 851–885 (2008)
37. Pitman, E.B., Le, L.: A two-fluid model for avalanche and debris flows. *Philos. Trans. R. Soc. A* **363**, 1573–1601 (2005)
38. Rhebergen, S., Bokhove, O., van der Vegt, J.J.W.: Discontinuous Galerkin finite element methods for hyperbolic nonconservative partial differential equations. *J. Comput. Phys.* **227**, 1887–1922 (2008)
39. Saurel, R., Abgrall, R.: A multiphase Godunov method for compressible multifluid and multiphase flows. *J. Comput. Phys.* **150**, 425–467 (1999)
40. Schwendeman, D.W., Wahle, C.W., Kapila, A.K.: The Riemann problem and a high-resolution Godunov method for a model of compressible two-phase flow. *J. Comput. Phys.* **212**, 490–526 (2006)
41. Shu, C.-W.: High order weighted essentially non-oscillatory schemes for convection dominated problems. *SIAM Rev.* **51**, 82–126 (2009)
42. Shu, C.-W.: Essentially non-oscillatory and weighted essentially non-oscillatory schemes. *Acta Numer.* **29**, 701–762 (2020)
43. Shu, C.-W., Osher, S.J.: Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.* **77**, 439–471 (1988)
44. Shu, C.-W., Osher, S.J.: Efficient implementation of essentially non-oscillatory shock capturing schemes II. *J. Comput. Phys.* **83**, 32–78 (1989)
45. Spiteri, R.J., Ruuth, S.J.: A new class of optimal high-order strong-stability-preserving time-stepping schemes. *SIAM J. Numer. Anal.* **40**, 469–491 (2002)
46. Spiteri, R.J., Ruuth, S.J.: Non-linear evolution using optimal fourth-order strong-stability-preserving Runge-Kutta methods. *Math. Comput. Simul.* **62**, 125–135 (2003)

47. Tokareva, S.A., Toro, E.F.: HLLC-type Riemann solver for the Baer-Nunziato equations of compressible two-phase flow. *J. Comput. Phys.* **229**, 3573–3604 (2010)
48. Woodward, P., Colella, P.: The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.* **54**, 115–173 (1984)
49. Zheng, F., Shu, C.-W., Qiu, J.: A high order conservative finite difference scheme for compressible two-medium flows. *J. Comput. Phys.* **445**, 110597 (2021)
50. Zhu, J., Qiu, J.: A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws. *J. Comput. Phys.* **318**, 110–121 (2016)
51. Zhu, J., Shu, C.-W.: A new type of multi-resolution WENO schemes with increasingly higher order of accuracy. *J. Comput. Phys.* **375**, 659–683 (2018)

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## Authors and Affiliations

Dinshaw S. Balsara<sup>1,2</sup> · Deepak Bhooriya<sup>1</sup> · Chi-Wang Shu<sup>3</sup> · Harish Kumar<sup>4</sup>

✉ Dinshaw S. Balsara  
dbalsara@nd.edu

<sup>1</sup> Physics Department, University of Notre Dame, Notre Dame, USA

<sup>2</sup> ACMS Department, University of Notre Dame, Notre Dame, USA

<sup>3</sup> Division of Applied Mathematics, Brown University, Providence, USA

<sup>4</sup> Department of Mathematics, Indian Institute of Technology, Delhi, India