



# A single-step third-order temporal discretization with Jacobian-free and Hessian-free formulations for finite difference methods

Youngjun Lee, Dongwook Lee \*

Department of Applied Mathematics, The University of California, Santa Cruz, CA, United States



## ARTICLE INFO

### Article history:

Received 29 May 2020

Received in revised form 1 December 2020

Accepted 5 December 2020

Available online 11 December 2020

### Keywords:

Picard integration formulation

Jacobian-free

Hessian-free

Cauchy-Kowalewski procedure

High-order method

Finite difference method

## ABSTRACT

In this paper, we present an algorithmic extension of the method called the Picard Integration Formulation (PIF) that belongs to temporal updates based on the Lax-Wendroff procedure. The new extension is called the system-free (SF) approach, which furnishes ease of calculating the Jacobian and the Hessian terms necessary for third-order temporal accuracy in the original PIF method. In contrast to the analytical calculations of the Jacobian and the Hessian tensor terms in the original PIF method, our new SF approach utilizes finite difference approximations that replace the analytical calculations of the Jacobian and Hessian terms with Jacobian-free and Hessian-free approximations in a way commonly adopted in the context of iterative methods. The resulting SF approach enables our new PIF method to be a computationally efficient single-step, third-order accurate temporal scheme, whose computational performance is twice faster than the three-stage SSP-RK3 method with the same accuracy.

© 2020 Elsevier Inc. All rights reserved.

## 1. Introduction

The evolution of high-performance computing (HPC) drives the advent of high-order discrete methods in computational fluid dynamics (CFD) research area. As the hardware progression of the memory capacity per compute core has become gradually saturated [1,2], the HPC community has been compelled to find more efficient ways that can best exercise computing resources in pursuing computer simulations. To meet this end, modern practitioners have relentlessly delved into advancing high-arithmetic-intensity models that can increase numerical accuracy per degree of freedom, while at the same time, can operate with reduced memory requirements and data transfers in HPC architectures. One such computing paradigm is to promote high-order methods in which high arithmetic intensity is achieved by putting to use an increasing number of higher order terms. High-order methods (third-order or higher) avail themselves of an improved figure-of-merit in terms of reaching a target numerical error per computational time faster than the other counterpart, i.e., lower-order methods.

Within the broad framework of finite difference method (FDM) and finite volume method (FVM) discretization schemes, it has been proven that high-order data interpolations and reconstructions play a key role to reduce numerical errors for solving discrete partial differential equations (PDEs) on a given grid resolution. As seen in the success of the Piecewise Parabolic Method (PPM) [3] and Weighted Essentially Non-Oscillatory (WENO) method [4], to name a few, high-order in-

\* Corresponding author.

E-mail addresses: [ylee109@ucsc.edu](mailto:ylee109@ucsc.edu) (Y. Lee), [dlee79@ucsc.edu](mailto:dlee79@ucsc.edu) (D. Lee).

terpolation/reconstruction schemes produce more accurate numerical solutions with a faster rate of convergence-to-solution on a limited computing resource. It is of great advantage to adopting the increased intensity of floating-point operations in scientific simulations on modern HPC architecture, because otherwise increasing grid resolutions to reach the desired solution convergence will require a larger memory footprint which is typically bounded for most HPC systems.

To acquire highly accurate numerical solutions, a high-order temporal discretization scheme ought to be considered alongside the high-order spatial data interpolation/reconstruction. In the context of FDM and FVM formulations, numerical errors arise from both spatial and temporal discretizations since the solution lies on the spatio-temporal plane. There are two different regimes where the dominance of truncation errors can come from either the temporal solver or the spatial solver. In case that the accuracy of both spatial and temporal solvers are comparable (i.e.,  $m \sim n$ ) and the grid resolution is not highly refined, the temporal error of an  $m$ -th order time integrator is typically smaller than the spatial error of an  $n$ -th order spatial solver i.e.,  $\mathcal{O}(\Delta t^m) < \mathcal{O}(\Delta s^n)$ , where  $\Delta s = \min_{d=x,y,z} \{\Delta d\}$ . This, however, flips to the other end in which the temporal error becomes more dominant over the spatial error,  $\mathcal{O}(\Delta t^m) > \mathcal{O}(\Delta s^n)$ , particularly when the temporal accuracy is much lower than the spatial accuracy (e.g.,  $m = 2$  and  $n = 5$ ) and/or the computational need necessitates a finer grid resolution to resolve smaller physical length scales.

For decades, the strong stability preserving Runge-Kutta (SSP-RK) time integrator [5–7] has been considered to be the *standard* strategy for an extensive range of high-order numerical schemes for PDE solvers. The SSP-RK schemes have proven its high fidelity which guarantees not only high-order accuracy but also numerical stability with total variation diminishing (TVD) property. The chief objective of SSP-RK schemes is to maintain the strong stability property (SSP) at high-order accuracy by sequentially applying convex combinations of the first-order forward Euler method as a building-block at each sub-stage [6]. The desired total variation diminishing (TVD) property is achieved if each of the sub-stage is TVD. The Runge-Kutta schemes with this TVD property are known as SSP-RK schemes. Designed this way, the SSP-RK procedure requires  $s$  multiple sub-stages to advance the solution by a single time step  $\Delta t$  at  $m$ -th accuracy.

Of the most popular choice among various SSP-RK solvers is the three-stage, third-order RK3, given that, in practice, spatial accuracy is often considered to carry more weights than temporal accuracy in designing higher accurate spatial models [8,9]. In contrast, devising a fourth-order SSP-RK4 is more involved to meet the favorable SSP property. Several theoretical studies have shown that a fourth-order SSP-RK4 cannot be formulated with just four sub-stages [5], meaning that the classical four-stage, fourth-order RK4 is not SSP. Other authors have demonstrated that SSP-RK4 with positive coefficients could be constructed with an increasing number of sub-stages from five up to eight [10,11]. We remark that, to provide stable, non-oscillatory predictions in the presence of shocks, it is safer to employ SSP-RK schemes than to use non SSP-RK schemes to prevent unphysical oscillations near shocks and discontinuities as they propagate [7]. However, such SSP-RK schemes experience theoretical barriers in attaining the maximum available order and the maximum CFL coefficient [12,13]. Besides, the spatial interpolation/reconstruction, as well as boundary conditions, should be applied at each sub-stage, which makes the overall procedure of SSP-RK computationally expensive. In parallel simulations, these operations also increase the footprint of data movements between node communications as the number of sub-stages grows. This very nature of SSP-RK makes it less efficient for massively parallel simulations particularly when the level of adaptive mesh refinement (AMR) progressively builds up around interesting features in simulations.

As a means of circumventing the said issues in SSP-RK, practitioners have taken a different route of providing a high-order temporal updating strategy for solving numerical PDEs. The core design principle lies in formulating a conservative temporal integrator which works for nonlinear PDEs in multiple spatial dimensions with the equivalent high-order accuracy as in SSP-RK, but, this time, in a single-stage, single-step update. The effort in this direction has resulted in what is now widely known as Arbitrary high order DERivative (ADER) method, which was first introduced in [14] for linear equations. Since then, ADER schemes have gone through several generations of breakthrough by numerous authors with the common goal of meeting the high-order requirement in a compact single-step update.

In the developments led by Toro and his collaborators, the single-step, high-order ADER solution update is attained by solving a series of generalized (or high-order) Riemann problems (GRPs) [15–17] to compute the coefficients corresponding to the high-order terms in the power series expansion of each conservative variable in time [18,19,14]. Solving GRPs involves two steps. The first step is to solve one set of classical nonlinear Riemann problems for the leading term. The second step is to solve another set of classical – but linearized – Riemann problems for the remaining higher derivative terms via the Cauchy-Kowalewski procedure, which exercises a sequence of full couplings between spatial and temporal derivatives.

The ADER formulation has been further taken to a more modern direction. Dumbser et al. [20] extended the original ADER scheme with multiple quadrature points to an efficient quadrature-free ADER scheme with a space-time averaged numerical flux, and the authors applied the scheme to integrate a discontinuous Galerkin (DG) solver. Balsara et al. [21] presented a new compact ADER framework that replaces the usual Cauchy-Kowalewski procedure in the original ADER formalism with a local continuous space-time Galerkin formulation up to fourth-order, and called the new approach ADER-CG (CG for continuous Galerkin). A comprehensive review on the recent developments of ADER-CG has been reported in [22]. In [23], the ADER-CG schemes are shown to be approximately twice faster than the RK methods at the same order of accuracy. It is also shown that ADER is highly adaptable to the AMR grid configuration [24]. The ADER formulation has been further extended to compressible dissipative flows in both hydrodynamics and magnetohydrodynamics in the ADER-DG framework [25] using the MOOD method [26–30] as a posteriori detection to enhance the shock-capturing capability; a comparison study of the space-time predictor calculation based on primitive, conservative, and characteristic variables [31];

a new approach of using the so-called automatic differentiation (or the differential transform method) to reduce the computational cost of ADER schemes called the ADER-Taylor method [32–34], etc.

Unlike the broad usage of SSP-RK in various discrete PDE solvers, the developments of ADER mentioned above have been exclusively applied to FV and DG methods, but FD methods. This is mainly because the fundamental principle of obtaining high-order accuracy in the original ADER scheme relies on solving generalized (or high-order) Riemann problems, which are the characteristic building blocks of FV and DG methods. In this regard, perhaps there could be a pathway to devise an ADER scheme for FDM within a context of a new FDM formulation called FD-Prim [35–37] in which solving Riemann problems is fully utilized as a new way of forming high-order FD fluxes.

Categorically, ODE solvers can be classified as members of multi-stage multi-derivative methods [38,39]. For example, RK schemes are of the multi-stage type as they attain high-order accuracy via utilizing multi-stages as a primary mechanism, whereas ADER methods are of the multi-derivative type since they explore multi-derivatives as a central engine. In this paper, we are interested in constructing – for general FD solvers – a single-stage, single-step temporal updating algorithm that belongs to the class of multi-derivative methods. We follow the recent approach of the Picard integral formulation (PIF) method [40,41] for achieving high-order temporal accuracy in FDM, where the authors designed a high-order temporal discretization by constructing a high-order approximation to the *time-averaged fluxes* over  $[t^n, t^{n+1}]$ . The PIF discretization introduced in [40] attains third-order temporally accurate numerical fluxes by computing the coefficients of the Taylor expansion of the averaged fluxes up to third-order via the Lax-Wendroff (or Cauchy–Kowalewski) procedure which converts the high-order temporal derivatives terms in the Taylor expansion to spatial derivative terms.

As the resulting PIF scheme incorporates the needed high-order discretization directly into the numerical fluxes, the governing hyperbolic conservation law,  $\partial_t \mathbf{U} + \nabla \cdot \mathcal{F}(\mathbf{U}) = 0$ , can be inherently satisfied at least on a simple uniform Cartesian geometry. In contrast, in the one-step temporal discretization methods reported previously in [42,43] for FDM, which are analogously based on the Lax-Wendroff procedure, the high-order expansion was performed on conserved variables with the Lax-Wendroff conversion of temporal derivatives to spatial derivatives. The mechanism that made these approaches conservative is set by discretizing those spatial derivatives via central differencing, which is less intrinsic than that of PIF in terms of constructing a conservative update in a discrete sense. Accordingly, in PIF, there are a few more benefits of working on the time-averaged fluxes. As shown in [41], an extra manipulation of numerical fluxes (e.g., a hybrid numerical flux as a linear combination of a low- and high-order flux to preserve positivity) does not break the hyperbolic conservation law. In addition, as will be seen throughout the present study, the high-order handling of the time-averaged fluxes can serve as an extra standalone step, apart from the rest of the conventional FDM discretization steps such as the flux splitting, high-order flux reconstruction, as well as the final temporal update. As a consequence, the PIF step can readily replace any existing SSP-RK updating step, boosting the overall performance gain by a factor of two.

The primary objective of the current work lies in providing increased flexibility and ease of computation for approximating the needed flux Jacobian and Hessian tensor terms in the original third-order PIF method [40,41]. To do that, we offer an extended algorithmic development that permits the equivalent single-step, third-order PIF method that does not involve the analytic evaluations of the Jacobian and Hessian terms as described in [40,41]. We call our new method a system-free PIF method (or SF-PIF in short) to reflect the added capability of the proposed Jacobian-free and Hessian-free approach.

We organize the paper as follows. In Section 2, we give an overview of the general principles in the PIF-based FD discretization. The new concept of our system-free approach to approximating the flux Jacobian and Hessian tensor terms is introduced in Section 3. We test the new SF-PIF method on a series of 1D and 2D benchmark problems in the system of the Euler equations as well as the shallow water equations. The test results are presented in Section 4. We conclude our paper with a brief summary in Section 5.

## 2. PIF method

We are interested in solving the general conservation laws of hyperbolic PDEs in 1D and 2D, predicting numerical solutions with the target third-order accuracy in time. To present a mathematical foundation of our method, we begin with the following general system of equations in 2D,

$$0 = \frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathcal{F}(\mathbf{U}) = \frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y}. \quad (1)$$

Denoted by  $\mathbf{U}$  is a vector of conservative variables,  $\mathbf{F}$  and  $\mathbf{G}$  are the flux functions in  $x$ - and  $y$ -directions, respectively. Here, we apply the Picard Integral Formulation (PIF) [40] by taking a time average of Eq. (1) within a single timestep  $\Delta t$  over an interval  $[t^n, t^n + \Delta t] = [t^n, t^{n+1}]$ ,

$$\mathbf{U}^{n+1} = \mathbf{U}^n - \Delta t \left( \frac{\partial \mathbf{F}^{avg}}{\partial x} + \frac{\partial \mathbf{G}^{avg}}{\partial y} \right), \quad (2)$$

where  $\mathbf{F}^{avg}$  and  $\mathbf{G}^{avg}$  represent the time-averaged fluxes in  $x$ - and  $y$ -direction respectively,

$$\mathbf{F}^{avg}(\mathbf{x}) \equiv \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \mathbf{F}(\mathbf{U}(\mathbf{x}, t)) dt, \quad \mathbf{G}^{avg}(\mathbf{x}) \equiv \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \mathbf{G}(\mathbf{U}(\mathbf{x}, t)) dt, \quad \mathbf{x} = (x, y) \in \mathbb{R}^2. \quad (3)$$

In order to get a spatially discretized solution  $\mathbf{U}_{i,j} \equiv \mathbf{U}(\mathbf{x}_{ij})$ , we wish to express Eq. (2) with numerical fluxes  $\hat{\mathbf{f}}$  and  $\hat{\mathbf{g}}$  at cell interfaces defined as,

$$\frac{\partial \mathbf{F}^{avg}}{\partial x} \Big|_{\mathbf{x}=\mathbf{x}_{ij}} = \frac{1}{\Delta x} \left( \hat{\mathbf{f}}_{i+\frac{1}{2},j} - \hat{\mathbf{f}}_{i-\frac{1}{2},j} \right) + \mathcal{O}(\Delta x^p + \Delta t^q), \quad \mathbf{x}_{ij} = (x_i, y_j). \quad (4)$$

The numerical flux in  $y$ -direction,  $\hat{\mathbf{g}}$ , is defined in a similar fashion. Now we are able to express Eq. (1) in a fully discretized form as,

$$\mathbf{U}_{i,j}^{n+1} = \mathbf{U}_{i,j}^n - \frac{\Delta t}{\Delta x} \left( \hat{\mathbf{f}}_{i+\frac{1}{2},j} - \hat{\mathbf{f}}_{i-\frac{1}{2},j} \right) - \frac{\Delta t}{\Delta y} \left( \hat{\mathbf{g}}_{i,j+\frac{1}{2}} - \hat{\mathbf{g}}_{i,j-\frac{1}{2}} \right). \quad (5)$$

We remark that the derived governing form in Eq. (5) for PIF is something in between those of FVM and FDM. It is different from that of FVM that it does not carry any spatial average but the temporal average. It is also different from that of FDM that it does involve the temporal average in the fluxes in Eq. (3) to which the numerical fluxes  $\hat{\mathbf{f}}$  and  $\hat{\mathbf{g}}$  approximate.

We aim to get a high-order approximated solution for  $\mathbf{f}$  and  $\mathbf{g}$ , *both in space and time*, by which the update in Eq. (5) enables us to update the solution  $\mathbf{U}_{i,j}^{n+1}$  in a single step at the desired  $p$ -th and  $q$ -th accuracy in space and time. To do this, we first follow the standard convention in FDM in which we treat the pointwise  $x$ -flux  $\mathbf{F}(x, y_j)$  as a 1D cell average of an auxiliary function  $\hat{\mathbf{F}}$  in 1D,

$$\mathbf{F}(x, y_j) = \frac{1}{\Delta x} \int_{x - \frac{\Delta x}{2}}^{x + \frac{\Delta x}{2}} \hat{\mathbf{F}}(\xi, y_j) d\xi. \quad (6)$$

Then the analytic derivative of  $\mathbf{F}$  at  $x = x_i$  in  $x$ -direction becomes

$$\frac{\partial \mathbf{F}}{\partial x} \Big|_{x=x_i} = \frac{1}{\Delta x} \left( \hat{\mathbf{F}}(x_{i+\frac{1}{2}}, y_j) - \hat{\mathbf{F}}(x_{i-\frac{1}{2}}, y_j) \right). \quad (7)$$

Comparing Eq. (7) and Eq. (4), our goal is to be achieved if we can define the numerical flux  $\hat{\mathbf{f}}$  that satisfies the following relationship with  $\hat{\mathbf{F}}$ ,

$$\hat{\mathbf{f}}_{i \pm \frac{1}{2},j} = \hat{\mathbf{F}}(x_{i \pm \frac{1}{2}}, y_j) + \mathcal{O}(\Delta x^p + \Delta t^q). \quad (8)$$

Mathematically speaking, the inverse problem of Eq. (6) is exactly the same as the conventional 1D reconstruction problem in FVM, the operation of which is specifically designed to find the primitive function value  $\hat{\mathbf{F}}$  at a certain location (mostly,  $x_{i \pm 1/2}$ ) in the  $i$ -th cell, given the integral-averaged (or volume-averaged) values  $\mathbf{F}$  at nearby stencil points as input. Namely, this can be written as

$$\hat{\mathbf{F}}(\xi, y_j) = \mathcal{R}(\mathbf{F}_{i-r,j}, \dots, \mathbf{F}_{i+r,j}) + \mathcal{O}(\Delta x^p), \quad \xi \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}], \quad (9)$$

where  $\mathcal{R}$  is a  $p$ -th order accurate reconstruction operator that takes inputs of the pointwise fluxes  $\mathbf{F}$  at the  $x$  neighboring stencil  $[x_{i-r,j}, \dots, x_{i+r,j}]$  with fixed  $y = y_j$ .

We can split the task in Eq. (8) and Eq. (9) in two different stages in the following sequential order. The first step is to attain  $q$ -th order temporal accuracy, followed by the second step that provides  $p$ -th order spatial accuracy:

**Step 1** Approximation of the time-averaged flux – Obtain a  $q$ -th order approximated solution  $\mathbf{F}^{appx}(\mathbf{x})$  to Eq. (3):

$$\mathbf{F}^{avg}(\mathbf{x}) \equiv \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \mathbf{F}(\mathbf{U}(\mathbf{x}, t)) dt = \mathbf{F}^{appx}(\mathbf{x}) + \mathcal{O}(\Delta t^q). \quad (10)$$

**Step 2** Approximation of the high-order spatial reconstruction – Use a traditional high-order finite difference method with the *approximated time-averaged* fluxes  $\mathbf{F}_{i,j}^{appx}$  from **Step 1**, instead of the conventional choice of pointwise fluxes, i.e.,  $\mathbf{F}_{i,j}$  at  $t = t^n$ :

$$\hat{\mathbf{f}}_{i+\frac{1}{2},j} = \mathcal{R} \left( \mathcal{FS} \left( \mathbf{F}_{i-r,j}^{appx}, \dots, \mathbf{F}_{i+r+1,j}^{appx} \right) \right) + \mathcal{O}(\Delta x^p). \quad (11)$$

Here,  $\mathcal{FS}(\cdot)$  represents a characteristic flux splitting in the positive and negative directions, and  $\mathcal{R}(\cdot)$  is a high-order reconstruction scheme which is also used for FVM formulations.

In this study, we adopt the fifth-order WENO method (or WENO5) [4] for the reconstruction in **Step 2**, combined with the Rusanov Lax–Friedrichs flux splitting described in [9], in order to attain high-order accuracy in space domain.

Before we discuss **Step 1** in details, we give a brief description on the flux splitting  $\mathcal{FS}(\cdot)$ , given the fact that we perform the splitting on the time-averaged approximated fluxes,  $\mathbf{F}_{\ell,j}^{appx}$ ,  $i-r \leq \ell \leq i+r+1$ , whose temporal state is not at  $t=t^n$  but at an average state over  $[t^n, t^{n+1}]$ . This is different from the conventional FDM splitting which performs on the pointwise fluxes  $\mathbf{F}$  at  $t=t^n$ . Following [9], the Rusanov Lax–Friedrichs splitting projects the temporally averaged fluxes  $\mathbf{F}_{\ell,j}^{appx}$  from **Step 1** to the left- and the right-going parts according to the characteristic decomposition of the Jacobian matrix,

$$\frac{\partial \mathbf{F}}{\partial \mathbf{U}} \Big|_{\mathbf{U}_{i+\frac{1}{2},j}^n} = \mathbf{R}_{i+\frac{1}{2},j} \Lambda_{i+\frac{1}{2},j} \mathbf{L}_{i+\frac{1}{2},j}, \quad \mathbf{U}_{i+\frac{1}{2},j}^n = \frac{\mathbf{U}_{i,j}^n + \mathbf{U}_{i+1,j}^n}{2}, \quad (12)$$

where  $\mathbf{R}$  and  $\mathbf{L}$  are the corresponding matrices of right and left eigenvectors, and  $\Lambda$  is the diagonal matrix whose diagonal entries are eigenvalues. The projection proceeds to construct  $s$  different left-going (–) and right-going (+) characteristic states of the averaged-fluxes, denoted as  $\mathbf{V}_{(i+\frac{1}{2}):s,j}^{k,\pm}$ , to the cell interface  $(i+\frac{1}{2}, j)$  as

$$\mathbf{V}_{(i+\frac{1}{2}):s,j}^{k,+} = \mathcal{FS}^+(\mathbf{F}_{s,j}^{appx}) = \frac{1}{2} \mathbf{L}_{i+\frac{1}{2},j}^k \cdot (\mathbf{F}_{s,j}^{appx} + \alpha^k \mathbf{U}_{s,j}), \quad \mathbf{V}_{(i+\frac{1}{2}):s,j}^{k,-} = \mathcal{FS}^-(\mathbf{F}_{s',j}^{appx}) = \frac{1}{2} \mathbf{L}_{i+\frac{1}{2},j}^k \cdot (\mathbf{F}_{s',j}^{appx} - \alpha^k \mathbf{U}_{s',j}), \quad (13)$$

where the sub-index  $s$  ranges from  $i-r, \dots, i+r$ , while at the same time,  $s' = 2i-s+1$ . The super-index  $k$  represents each characteristic field. The coefficient  $\alpha^k$  is chosen to be the maximum absolute value of the  $k$ -th characteristic speed over the entire computational domain, resulting in the so-called *global* Lax–Friedrichs flux splitting. Here, we point out that, in conventional FDM, the temporal states of the flux and conservative vectors, the left eigenvectors, and  $\alpha^k$  in Eq. (13) are all consistently at  $t=t^n$ . For PIF, though, the flux vector  $\mathbf{F}_{s,j}^{appx}$  is at the averaged temporal state, while the others are at  $t=t^n$ .

Ideally, one can also choose to compute a similar temporally averaged conservative vectors  $\mathbf{U}_{s,j}^{appx}$ , use them to compute  $\mathbf{L}_{i+\frac{1}{2},j}^k$  and  $\alpha^k$ , in order to provide a consistent temporal handling across all the quantities in Eq. (13). However, we have found that the level of numerical stability as well as the solution accuracy (e.g., preserving the expected symmetries in some symmetry-preserving problems) tend to become less maintained in that approach. This observation has made us to decide that our flux splitting uses theoretically inconsistent temporal states between  $\mathbf{F}_{s,j}^{appx}$  and the other quantities. As will be shown in Section 4, this treatment does not affect the overall solution accuracy and stability. The same choice has been also made in the original PIF method [40]. The rest of the flux splitting procedure follows to compute the numerical fluxes  $\hat{\mathbf{f}}_{i\pm\frac{1}{2},j}$  via a high-order FDM reconstruction operator,  $\mathcal{R}(\cdot)$ . More specifically, we write,

$$\hat{\mathbf{f}}_{i+\frac{1}{2},j} = \sum_k \left( \hat{\mathbf{v}}_{i+\frac{1}{2},j}^{k,+} + \hat{\mathbf{v}}_{i+\frac{1}{2},j}^{k,-} \right) \mathbf{R}_{i+\frac{1}{2},j}^k, \quad (14)$$

which is a linear combination (over the  $k$  characteristic fields) of the coefficients  $\hat{\mathbf{v}}_{i+\frac{1}{2},j}^{k,\pm}$  obtained by solving a high-order reconstruction,

$$\hat{\mathbf{v}}_{i+\frac{1}{2},j}^{k,\pm} = \mathcal{R} \left( \mathbf{V}_{(i+\frac{1}{2}):s,j}^{k,\pm} \right) = \mathcal{R} \left( \mathcal{FS}^+(\mathbf{F}_{s,j}^{appx}), \mathcal{FS}^-(\mathbf{F}_{s',j}^{appx}) \right), \quad s = i-r, \dots, i+r, \quad \text{with } s' = 2i-s+1. \quad (15)$$

In this paper we use  $r=2$  for the fifth-order WENO5 method [4]. We assume that readers are familiar with the WENO5 procedure in Eq. (15); hence it is omitted in this paper. Interested readers are encouraged to refer to [4,9].

Let us now focus on **Step 1**, our main interest in this study, which comprises achieving a high-order temporal accuracy of the numerical fluxes  $\mathbf{F}_{i,j}^{appx}$ . Traditionally, this is accomplished by a multi-stage time integrator, such as an  $s$ -stage,  $m$ -th order RK method. However, as briefly discussed in Section 1, a multi-stage method is computationally expensive, and in practice, is not suitable on an adaptive mesh refinement (AMR) grid configuration due to the increasing amount of calculation and data movements as the refinement level grows in simulations. On the other hand, by being a single-stage, single-step method, a predictor-corrector-type of time integration schemes that are based on the Lax–Wendroff (or Cauchy–Kowalewski) procedure has an advantage to reduce a plenty of computational costs and is favorable to be implemented on an AMR grid configuration while maintaining a high-order accuracy.

The PIF method is to be considered as an approach that belongs to this time integration category, featuring a new formulation of involving the concept of time-averaged fluxes. Using a time-averaged quantity in a predictor step has been widely used in FVM formulations, including the original Godunov scheme [44], piecewise-type methods [3,45] and arbitrary derivative (ADER) schemes [18,19,21]. Recently, Seal et al. [40,41] extended this approach to the FDM formulation, by way of introducing a Picard integration as described in Eq. (2), in which the time-averaged fluxes in Eq. (3) are approximated through Taylor expansion of the integrand around  $t^n$ . For example, in the third-order temporal PIF method, the time-averaged flux  $\mathbf{F}^{avg}(\mathbf{x})$  is approximated by an integral of the Taylor expansion of the pointwise flux  $\mathbf{F}(\mathbf{x}, t)$  around  $t^n$ ,

$$\begin{aligned}
\mathbf{F}^{avg}(\mathbf{x}) &= \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \mathbf{F}(\mathbf{x}, t) dt \\
&= \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \left[ \mathbf{F}(\mathbf{x}, t^n) + (t - t^n) \frac{\partial \mathbf{F}(\mathbf{x}, t)}{\partial t} \Big|_{t=t^n} + \frac{(t - t^n)^2}{2!} \frac{\partial^2 \mathbf{F}(\mathbf{x}, t)}{\partial t^2} \Big|_{t=t^n} + \dots \right] dt \\
&= \mathbf{F}(\mathbf{x}, t^n) + \frac{\Delta t}{2!} \frac{\partial \mathbf{F}(\mathbf{x}, t)}{\partial t} \Big|_{t=t^n} + \frac{\Delta t^2}{3!} \frac{\partial^2 \mathbf{F}(\mathbf{x}, t)}{\partial t^2} \Big|_{t=t^n} + \mathcal{O}(\Delta t^3).
\end{aligned} \tag{16}$$

Dropping the error term  $\mathcal{O}(\Delta t^3)$  in Eq. (16), we achieve a third-order temporally accurate approximation to the averaged-flux  $\mathbf{F}^{avg}(\mathbf{x})$ , which is what we aimed for.

The temporal derivatives can be cast in terms of spatial derivatives through the Cauchy–Kowalewski procedure, satisfying the governing system of equations. We adopt an approach conceptually slightly different from [40] to convert the temporal derivatives in Eq. (16). To wit, in [40] the temporal derivatives of the fluxes in Eq. (16) inadvertently treats  $\mathbf{F}$  as a function (through  $\mathbf{U}$ ) that depends only on  $t$  but not on  $\mathbf{x}$ . The authors then followed a consistent logical pathway to convert the temporal derivatives to the corresponding spatial derivatives via the Cauchy–Kowalewski procedure. Although the final expression of such converted terms in [40] is found to be equivalent to our results presented below, we introduce a new self-consistent approach to converting the temporal *partial* derivatives in Eq. (16) to the related spatial derivatives.

To this end, we first apply the chain rule to  $\frac{\partial \mathbf{F}}{\partial t}$  which leads us to write an evolution equation of the  $x$ -flux  $\mathbf{F}$ ,

$$\frac{\partial \mathbf{F}}{\partial t} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial t} = -\frac{\partial \mathbf{F}}{\partial \mathbf{U}} \left( \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} \right). \tag{17}$$

From Eq. (17), the higher temporal derivatives are acquired in a successive manner. For example, the second derivative can be computed as,

$$\frac{\partial^2 \mathbf{F}}{\partial t^2} = -\frac{\partial}{\partial t} \left( \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \right) \cdot \left( \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} \right) - \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \cdot \frac{\partial}{\partial t} \left( \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} \right), \tag{18}$$

where

$$\frac{\partial}{\partial t} \left( \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \right) = \frac{\partial^2 \mathbf{F}}{\partial \mathbf{U}^2} \cdot \frac{\partial \mathbf{U}}{\partial t} = -\frac{\partial^2 \mathbf{F}}{\partial \mathbf{U}^2} \cdot \left( \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} \right), \tag{19}$$

and

$$\begin{aligned}
\frac{\partial}{\partial t} \left( \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} \right) &= \frac{\partial}{\partial x} \left( \frac{\partial \mathbf{F}}{\partial t} \right) + \frac{\partial}{\partial y} \left( \frac{\partial \mathbf{G}}{\partial t} \right) \\
&= \frac{\partial}{\partial x} \left( -\frac{\partial \mathbf{F}}{\partial \mathbf{U}} \cdot \left( \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} \right) \right) + \frac{\partial}{\partial y} \left( -\frac{\partial \mathbf{G}}{\partial \mathbf{U}} \cdot \left( \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} \right) \right) \\
&= -\frac{\partial^2 \mathbf{F}}{\partial \mathbf{U}^2} \cdot \frac{\partial \mathbf{U}}{\partial x} \cdot \left( \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} \right) - \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \cdot \left( \frac{\partial^2 \mathbf{F}}{\partial x^2} + \frac{\partial^2 \mathbf{G}}{\partial x \partial y} \right) \\
&\quad - \frac{\partial^2 \mathbf{G}}{\partial \mathbf{U}^2} \cdot \frac{\partial \mathbf{U}}{\partial y} \cdot \left( \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} \right) - \frac{\partial \mathbf{G}}{\partial \mathbf{U}} \cdot \left( \frac{\partial^2 \mathbf{F}}{\partial x \partial y} + \frac{\partial^2 \mathbf{G}}{\partial y^2} \right).
\end{aligned} \tag{20}$$

Here,  $\frac{\partial \mathbf{F}}{\partial \mathbf{U}}$  represents an  $N \times N$  flux Jacobian matrix with  $\mathbf{U} \in \mathbb{R}^N$ , while  $\frac{\partial^2 \mathbf{F}}{\partial \mathbf{U}^2}$  represents an  $N \times N \times N$  flux Hessian tensor. A dot product between the Hessian tensor and a vector is to be understood as a tensor contraction,<sup>1</sup> thus a double dot product between the Hessian tensor and two vectors, i.e.,  $\frac{\partial^2 \mathbf{F}}{\partial \mathbf{U}^2} \cdot (\cdot) \cdot (\cdot)$ , yields a vector of dimension  $N$ .

We approximate the spatial derivatives with the following 5-point central differencing formulae that are fourth-order,

$$\frac{\partial \mathbf{F}_i}{\partial x} = \frac{\mathbf{F}_{i-2} - 8\mathbf{F}_{i-1} + 8\mathbf{F}_{i+1} - \mathbf{F}_{i+2}}{12\Delta x} + \mathcal{O}(\Delta x^4), \tag{21}$$

$$\frac{\partial^2 \mathbf{F}_i}{\partial x^2} = \frac{-\mathbf{F}_{i-2} + 16\mathbf{F}_{i-1} - 30\mathbf{F}_i + 16\mathbf{F}_{i+1} - \mathbf{F}_{i+2}}{12\Delta x^2} + \mathcal{O}(\Delta x^4). \tag{22}$$

<sup>1</sup> For  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ , the tensor contraction  $\frac{\partial^2 \mathbf{F}}{\partial \mathbf{U}^2} \cdot \mathbf{x} \cdot \mathbf{y}$  is defined in such a way that its  $k$ -th component is given as  $\left[ \frac{\partial^2 \mathbf{F}}{\partial \mathbf{U}^2} \cdot \mathbf{x} \cdot \mathbf{y} \right]_k = \sum_{i,j=1}^N \frac{\partial^2 \mathbf{F}_k}{\partial \mathbf{U}_i \partial \mathbf{U}_j} \mathbf{x}_i \mathbf{y}_j$ .

Theoretically speaking, in Eq. (16), we are able to retain the third-order temporal accuracy for  $\mathbf{F}^{\text{avg}}(\mathbf{x})$  as long as we approximate  $\partial\mathbf{F}/\partial t$  with an alternative spatial difference approximant whose leading error is  $\mathcal{O}(\Delta x^2)$  at least (e.g., a second-order or higher finite difference approximation of the flux derivative terms in Eq. (17)) and  $\partial^2\mathbf{F}/\partial t^2$  with  $\mathcal{O}(\Delta x)$  at least, assuming  $\Delta t = \mathcal{O}(\Delta x)$  provided by the hyperbolic Courant condition. In this regard, our design choice with the fourth-order approximations in Eqs. (21) and (22) is indeed surfeited. In addition, between Eqs. (21) and (22), it is sufficient to retain one order lower in Eq. (22) than in Eq. (21) because of the extra  $\Delta t$  factor for the second derivative term in Eq. (16). In fact, in case with  $\partial^2\mathbf{F}_i/\partial x^2 = \mathcal{O}(\Delta x^2)$ , some of our test results have indicated that the convergence rate drops slightly without affecting the expected third-order solution accuracy. In practice, the use of any lower order approximation in Eq. (21) or Eq. (22) does not gain the overall computational performance; hence we stick to the fourth-order approximations.

For the cross derivatives, the simple second order central differencing is sufficient to retain the third-order temporal accuracy,

$$\frac{\partial^2\mathbf{F}}{\partial x\partial y} = \frac{\mathbf{F}_{i+1,j+1} - \mathbf{F}_{i-1,j+1} - \mathbf{F}_{i+1,j-1} + \mathbf{F}_{i-1,j-1}}{4\Delta x\Delta y} + \mathcal{O}(\Delta x^2, \Delta y^2). \quad (23)$$

The use of the finite central differencing formulae above leads us to approximate all the terms described in Eq. (16) with sufficient accuracy, and the time-averaged flux  $\mathbf{F}^{\text{avg}}$  is ready to be reconstructed by the WENO5 method. The  $y$ -directional flux  $\mathbf{G}^{\text{avg}}$  can be obtained in a similar way, and this finalizes the general PIF procedure introduced in [40].

What remains to discuss is the system-free methodology, an approach newly introduced in Section 3, to provide a computationally affordable and efficient strategy for Jacobian and Hessian calculations. As can be seen in [40], the straight analytical calculations of the Jacobian matrix  $\frac{\partial\mathbf{F}}{\partial\mathbf{U}}$  and the Hessian tensor  $\frac{\partial^2\mathbf{F}}{\partial\mathbf{U}^2}$  make hurdles to implement the PIF into a code in practice. The situation becomes even worse when extending the current method to a fourth-order or higher, in which case higher derivatives will require high dimensional tensors whose size,  $N^D$  with  $D > 3$ , grows dramatically. Although the Jacobian and the Hessian calculations can be easily obtained with the aid of symbolic manipulators such as SymPy, Mathematica, or Maple, it still demands complicated coding/debugging efforts and ample memory consumption. Furthermore, as the Jacobian/Hessian calculations highly depend on the type of the governing system under consideration, it is required to re-derive the Jacobian/Hessian terms analytically every time we need to solve a new system, e.g., shallow water equations, or magnetohydrodynamics (MHD) equations, to name a few.

With this in mind, in the following section, we propose a new way to attain a high-order approximation for the time-averaged fluxes without the need for direct calculation of Jacobians or Hessians. The method is to be considered as a system-free PIF method, which will be referred to as SF-PIF in what follows.

### 3. System-free approach

Our goal in this section is to provide a new alternate formulation of computing the multiplications of Jacobian-vector and Hessian-vector-vector terms in Eqs. (18) – (20). The new approach will replace the necessity for analytical derivations of these Jacobian and Hessian terms in the original PIF method that are system-dependent, with a new system-independent formulation, based on the so-called “Jacobian-free” method which is widely used for Newton-Krylov-type iterative schemes [46–49]. We also extend it to a “Hessian-free” method. The resulting PIF method, called the system-free PIF (SF-PIF), approximates the Jacobian and Hessian terms via the Jacobian-free and Hessian-free approach.

#### 3.1. Jacobian approximation

To begin with our discussion, we consider a Taylor expansion for the flux vector at a small displacement from  $\mathbf{U}$ ,

$$\mathbf{F}(\mathbf{U} + \varepsilon\mathbf{V}) = \mathbf{F}(\mathbf{U}) + \varepsilon \frac{\partial\mathbf{F}}{\partial\mathbf{U}} \cdot \mathbf{V} + \frac{1}{2}\varepsilon^2 \frac{\partial^2\mathbf{F}}{\partial\mathbf{U}^2} \cdot \mathbf{V} \cdot \mathbf{V} + \mathcal{O}(\varepsilon^3), \quad (24a)$$

$$\mathbf{F}(\mathbf{U} - \varepsilon\mathbf{V}) = \mathbf{F}(\mathbf{U}) - \varepsilon \frac{\partial\mathbf{F}}{\partial\mathbf{U}} \cdot \mathbf{V} + \frac{1}{2}\varepsilon^2 \frac{\partial^2\mathbf{F}}{\partial\mathbf{U}^2} \cdot \mathbf{V} \cdot \mathbf{V} + \mathcal{O}(\varepsilon^3), \quad (24b)$$

where  $\mathbf{V}$  is an arbitrary vector that has the same number of components as  $\mathbf{U}$ , and  $\varepsilon$  is a small scalar perturbation. By subtracting Eq. (24b) from Eq. (24a), we get an expression of a central differencing that is of second-order in  $\varepsilon$ ,

$$\frac{\partial\mathbf{F}}{\partial\mathbf{U}} \cdot \mathbf{V} = \frac{\mathbf{F}(\mathbf{U} + \varepsilon\mathbf{V}) - \mathbf{F}(\mathbf{U} - \varepsilon\mathbf{V})}{2\varepsilon} + \mathcal{O}(\varepsilon^2). \quad (25)$$

Alternatively, the first-order forward differencing or the backward differencing can be used here. However, we choose the above second-order central differencing so that the order of accuracy of the entire system-free approach consistently scales with  $\mathcal{O}(\varepsilon^2)$ , given that the Hessian approximation described in the next subsection is to be bounded by  $\mathcal{O}(\varepsilon^2)$ . With the system-free approximation of Jacobian, all the Jacobian-vector products in Eqs. (17) – (20) are to be replaced with the central differencing in Eq. (25).

### 3.2. Hessian approximation

The flux Hessian tensor is unavoidable in designing the third-order PIF method in time. Below, we derive two different approximations with  $\mathcal{O}(\varepsilon^2)$  accuracy to estimate the two different types of the tensor appearing in Eqs. (18)–(20). Namely, in the first type the Hessian tensor contracts with the same vector, e.g.,  $\mathbf{V}$ , twice, and in the second type the tensor contracts with two different vectors, e.g.,  $\mathbf{V}$  and  $\mathbf{W}$ .

For the first type, we use a Taylor expansion analogous to Eq. (24) to approximate the Hessian-vector-vector product with a central differencing of order  $\mathcal{O}(\varepsilon^2)$ ,

$$\frac{\partial^2 \mathbf{F}}{\partial \mathbf{U}^2} \cdot \mathbf{V} \cdot \mathbf{V} = \frac{\mathbf{F}(\mathbf{U} + \varepsilon \mathbf{V}) - 2\mathbf{F}(\mathbf{U}) + \mathbf{F}(\mathbf{U} - \varepsilon \mathbf{V})}{\varepsilon^2} + \mathcal{O}(\varepsilon^2). \quad (26)$$

Using a simple vector calculus, the second type can be derived from the first type in Eq. (26) by exploring a symmetric property of the Hessians,

$$\frac{\partial^2 \mathbf{F}}{\partial \mathbf{U}^2} \cdot \mathbf{V} \cdot \mathbf{W} = \frac{1}{2} \left( \frac{\partial^2 \mathbf{F}}{\partial \mathbf{U}^2} \cdot (\mathbf{V} + \mathbf{W}) \cdot (\mathbf{V} + \mathbf{W}) - \left( \frac{\partial^2 \mathbf{F}}{\partial \mathbf{U}^2} \cdot \mathbf{V} \cdot \mathbf{V} + \frac{\partial^2 \mathbf{F}}{\partial \mathbf{U}^2} \cdot \mathbf{W} \cdot \mathbf{W} \right) \right) + \mathcal{O}(\varepsilon^2). \quad (27)$$

The Hessian approximations derived here are now ready to be substituted in Eqs. (18)–(20).

### 3.3. Proper choices of $\varepsilon$

In the previous subsections, we approximated the Jacobian-vector product and the Hessian-vector-vector product with a small perturbation  $\varepsilon$ . The choice of  $\varepsilon$  has to be considered carefully as it affects the solution accuracy and stability. On one hand, we need to minimize  $\varepsilon$  to improve the approximated solution accuracy, the quality of which will scale as the truncation error of  $\mathcal{O}(\varepsilon^2)$ . On the other hand, if it is too small the solution would be contaminated by the floating-point roundoff error which is bounded by the machine accuracy  $\varepsilon_{\text{mach}}$  [47]. Therefore,  $\varepsilon$  is to be determined judiciously to provide a good balance between the two types of error.

A recent study by An et al. [50] presents an effective analysis of choosing  $\varepsilon$  in the context of the Jacobian-free Newton-Krylov iterative framework. The authors have shown how to compute an ideal value of  $\varepsilon$  which minimizes the error of the central differencing in the Jacobian-vector approximation. We follow the same idea to obtain ideal values of  $\varepsilon$  for each Jacobian-vector and Hessian-vector-vector approximation.

The main idea in [50] is to find a good balance between the truncation error  $\mathcal{O}(\varepsilon^2)$  of each Jacobian-free approximation in Eq. (25) and Hessian-free approximation in Eq. (27), and the intrinsic floating-point roundoff error  $\delta \mathbf{F}(\mathbf{U})$  when calculating the target exact function value  $\mathbf{F}(\mathbf{U})$  with an approximate value  $\mathbf{F}(\mathbf{U}) + \delta \mathbf{F}(\mathbf{U})$ . The perturbation  $\delta \mathbf{F}(\mathbf{U})$  may include any errors characterized in computer arithmetic such as roundoff errors, and is assumed to be bounded by the machine accuracy, i.e.,  $\|\delta \mathbf{F}(\mathbf{U})\| \leq \varepsilon_{\text{mach}} \sim 2.2204 \times 10^{-16}$ . In this section, we only display the outcome of this analysis which will provide a good optimal estimation of  $\varepsilon$ , and leave a brief derivation of obtaining  $\varepsilon^{op}$  in Appendix A. More details of the relevant study are also found in [50,47].

Let  $\varepsilon^{op}$  denote the optimal value of  $\varepsilon$ , and in particular, we consider computer arithmetic in a 64-bit machine with double precision. Following the analysis in [50], we obtain  $\varepsilon^{op}$  for each Jacobian-free and Hessian-free approximation,

$$\begin{aligned} \varepsilon_{\text{jac}}^{op} &= \arg \min_{\sigma > 0} \left( \frac{\sigma^2}{2} + \frac{\varepsilon_{\text{mach}}}{2\sigma} \right) = \left( \frac{\varepsilon_{\text{mach}}}{2} \right)^{\frac{1}{3}} \approx 4.8062 \times 10^{-6}, \\ \varepsilon_{\text{hes}}^{op} &= \arg \min_{\sigma > 0} \left( \frac{\sigma^2}{3} + \frac{\varepsilon_{\text{mach}}}{\sigma^2} \right) = (3\varepsilon_{\text{mach}})^{\frac{1}{4}} \approx 1.6065 \times 10^{-4}, \end{aligned} \quad (28)$$

where  $\varepsilon_{\text{jac}}^{op}$  and  $\varepsilon_{\text{hes}}^{op}$  represent the optimal  $\varepsilon$  value for the Jacobian-free and Hessian-free approximations, respectively.

However, direct use of  $\varepsilon^{op}$  as the displacement step size in the central differencing schemes in Eq. (25) and Eq. (27) is not a good idea for stability reasons. Usually, the vector  $\mathbf{V}$  in Eq. (25) and Eq. (26) could have an enormous value in a strong shock region, so it is safer to use a smaller step size to preserve the needed stability. To meet this, we normalize the ideal value,  $\varepsilon^{op}$ , by the magnitude of the vector  $\mathbf{V}$ . There are several prescriptions available in the Jacobian-free Newton-Krylov literatures [47,46] to help finalize our decision of choosing a proper value of  $\varepsilon$  as a function of  $\varepsilon^{op}$ . Nonetheless, we have seen that, in a suite of test problems in Section 4, a simple approach of taking a square root of  $\varepsilon^{op}$  with a simple normalization is sufficient to attain the desired accuracy and stability, which is given as,

$$\bar{\varepsilon} = \frac{\sqrt{\varepsilon^{op}}}{\|\mathbf{V}\|_2}. \quad (29)$$

Lastly, we complete the  $\varepsilon$  estimation by taking the minimum value between  $\bar{\varepsilon}$  and  $\Delta t$ ,

$$\varepsilon = \min(\bar{\varepsilon}, \Delta t), \quad (30)$$

considering the fact that the Jacobian-free and Hessian-free terms will be multiplied by (at least)  $\Delta t$  after all. As a consequence, the choice of  $\varepsilon$  in Eq. (30) ensures the third-order temporal accuracy in the overall SF-PIF scheme.

We conclude this section with one important observation. In our numerical experiments, an improper choice of  $\varepsilon$  could lead to unphysical states of physical variables, or failures of preserving the assumed symmetry in simulations. As will be seen in Section 4, our choice of  $\varepsilon$  in this section plays a significant role in preserving the anticipated symmetries of each test problem without experiencing any numerical instabilities. Our prescription of  $\varepsilon$  has shown to provide a good balance between numerical accuracy and stability.

## 4. Results

In this section, we present numerical results of the SF-PIF algorithm on a suite of well-known benchmark problems in 1D and 2D using two different systems of equations. We first demonstrate the test results of SF-PIF on the system of 1D and 2D Euler equations, and compare them with the test results of RK3 and the original PIF method. The SF-PIF is also tested on the system of 2D shallow water equations to illuminate the system-free nature of the SF-PIF algorithm.

### 4.1. 1D Euler equations

In this section we test SF-PIF to simulate numerical problems in 1D Euler equations, defined as,

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = 0, \quad (31)$$

where the conservative variables and the corresponding fluxes are given as,

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{bmatrix}. \quad (32)$$

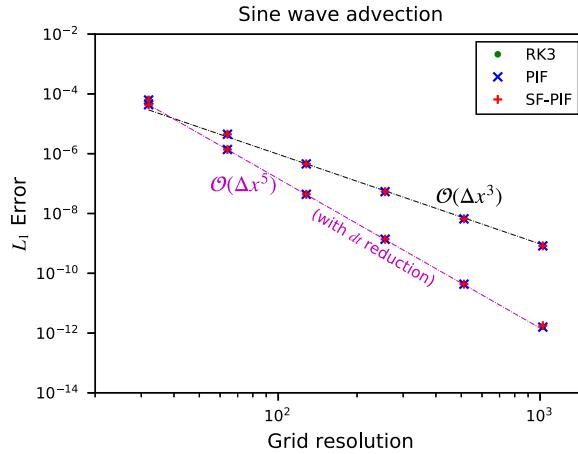
We adopt the spatially fifth-order WENO5 method [4] for all test problems, combined with three different time integrators: the traditional three-stage third-order Runge-Kutta method (RK3) [5], the original third-order Picard Integration Formulation (PIF) [40], and our third-order system-free PIF (SF-PIF) method presented in this paper. In all three combinations the nominal solution accuracy is expected to be fifth-order in space  $\mathcal{O}(\Delta x^5)$  and third-order in time  $\mathcal{O}(\Delta t^3)$ . We use a fixed Courant number,  $C_{\text{cfl}} = 0.7$  in all 1D cases.

#### 4.1.1. Sine wave advection

We start with a simple convergence test to see if the desired solution accuracy is retrieved in smooth flows. To that end, we choose a simple passive advection on a smooth flow configuration following the same setup as in [45]. The density profile is initialized with a sinusoidal wave,  $\rho(x) = 1.5 - 0.5 \sin(2\pi x)$ . The  $x$ -velocity and the pressure are set as constant values of  $u = 1$  and  $p = 1/\gamma$  with the specific heat ratio,  $\gamma = 5/3$ . Albeit solved using the nonlinear Euler equations, the problem is solved in a linear regime, viz., the velocity and pressure remain constant for all  $t \geq 0$  so that the initial sinusoidal density profile is purely advected by the constant velocity  $u = 1$  without any nonlinear dynamics such as a formation of shocks and rarefactions.

The simulation domain is defined on an interval  $[0, 1]$  with the periodic boundary condition on the both ends. The density profile will propagate for one period through the simulation domain and will return to its initial position at  $t = 1$ . On return, by being a linear problem, any deformation of the density profile from the initial shape can be considered as a numerical error associated with phase errors or numerical diffusions. The accuracy of the numerical solutions is measured by computing the  $L_1$  error between the solution at  $t = 1$  and the initial profile on a different number of cells  $N_x = 32, 64, 128, 256, 512$  and  $1024$ . The results are depicted in Fig. 1 for all three different temporal integrating schemes, RK3, PIF, and SF-PIF.

There are two types of result demonstrated in Fig. 1. In the first type, the discrete solutions of three different temporal schemes advanced with timesteps computed from the Courant condition with  $C_{\text{cfl}} = 0.7$ . Interestingly, we observe that the numerical solutions from SF-PIF as well as the other two schemes converge at third-order, indicating that the overall leading error of the simulation is dominated by the third-order accuracy of the temporal schemes despite the use of the more accurate fifth-order spatial WENO5 solver. As also briefly mentioned in Section 1, at low-mid resolutions, solution accuracy of simulations is mostly governed by the spatial accuracy until the leading error of the solution is caught up by the temporal error as computational grids get further refined to higher resolutions. So, in this problem, one may expect that the numerical solutions would pick up a faster convergence, faster than third-order and closer to fifth-order, at least at the lower end of the grid resolutions. Unfortunately, this doesn't happen in this test and the third-order temporal accuracy quickly takes over the control throughout the entire range of the grid resolutions tested herein. This solution behavior supports strongly the importance of integrating spatially interpolated/reconstructed solutions with a temporal scheme whose accuracy is sufficiently high enough to be well comparable to that of the spatial solver.



**Fig. 1.** Convergence test for the 1D sine wave advection problem. The errors are calculated in  $L_1$  sense against the initial density profile resolved on the computational grids refined from 32 to 1024 by a factor of 2. All numerical solutions follow the theoretical third-order convergence rate (the black-dotted line) when using the timesteps computed from the Courant condition. Also plotted are the solutions of using reduced timesteps, which follows the fifth-order convergence rate represented in the pink-dotted line. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

In the second type, however, timesteps are restricted in order to match up the lower third-order temporal accuracy of  $\mathcal{O}(\Delta t^3)$  with the higher fifth-order spatial accuracy of  $\mathcal{O}(\Delta x^5)$ .<sup>2</sup> We follow the usual trick of timestep reduction (e.g., see [9]) to manually adjust the timestep  $\Delta t_N$  on a grid size of  $N$  to an adjusted value, satisfying the equal rate of change between the spatial and temporal variations. As a result, the restricted (or reduced) timestep is defined by,

$$\Delta t_N = \Delta t_0 \left( \frac{\Delta x_N}{\Delta x_0} \right)^{\frac{5}{3}}. \quad (33)$$

The sub-indices “0” and “ $N$ ” refer to the time and grid scales on a nominal coarse and a fine resolution, respectively. For instance, in the current configuration,  $\Delta x_0$  is the grid scale on  $N_x = 32$  and  $\Delta t_0$  is the corresponding timestep subject to the Courant condition with  $C_{\text{cfl}} = 0.7$ . With the reduction, the overall leading error of the simulation (from both spatial and temporal) is matched with the fifth-order spatial accuracy of WENO5, and the numerical solution of SF-PIF follows the fifth-order rate as expected. Moreover, the SF-PIF solution compares indistinctly well with the other two solutions, almost in the same pattern. Even though the timestep reduction helps improve the numerical convergence rate, particularly when a temporal solver has a lower accuracy than a spatial solver, such simulations suffer not only from a much longer computational time to reach a target final time  $t = t_{\text{max}}$  but also from a degradation of solution effectiveness in resolving small scales due to an extended simulation time [51,52].

In all cases, the solution of SF-PIF compares equally well with the solutions of the original PIF and RK3 in both quantitatively and qualitatively.

#### 4.1.2. Sod's shock tube problem

The Sod's shock tube problem [53] is the one of the most famous hydrodynamics test problems for testing a numerical scheme's capability to handle discontinuities and shocks. The initial condition is given as,

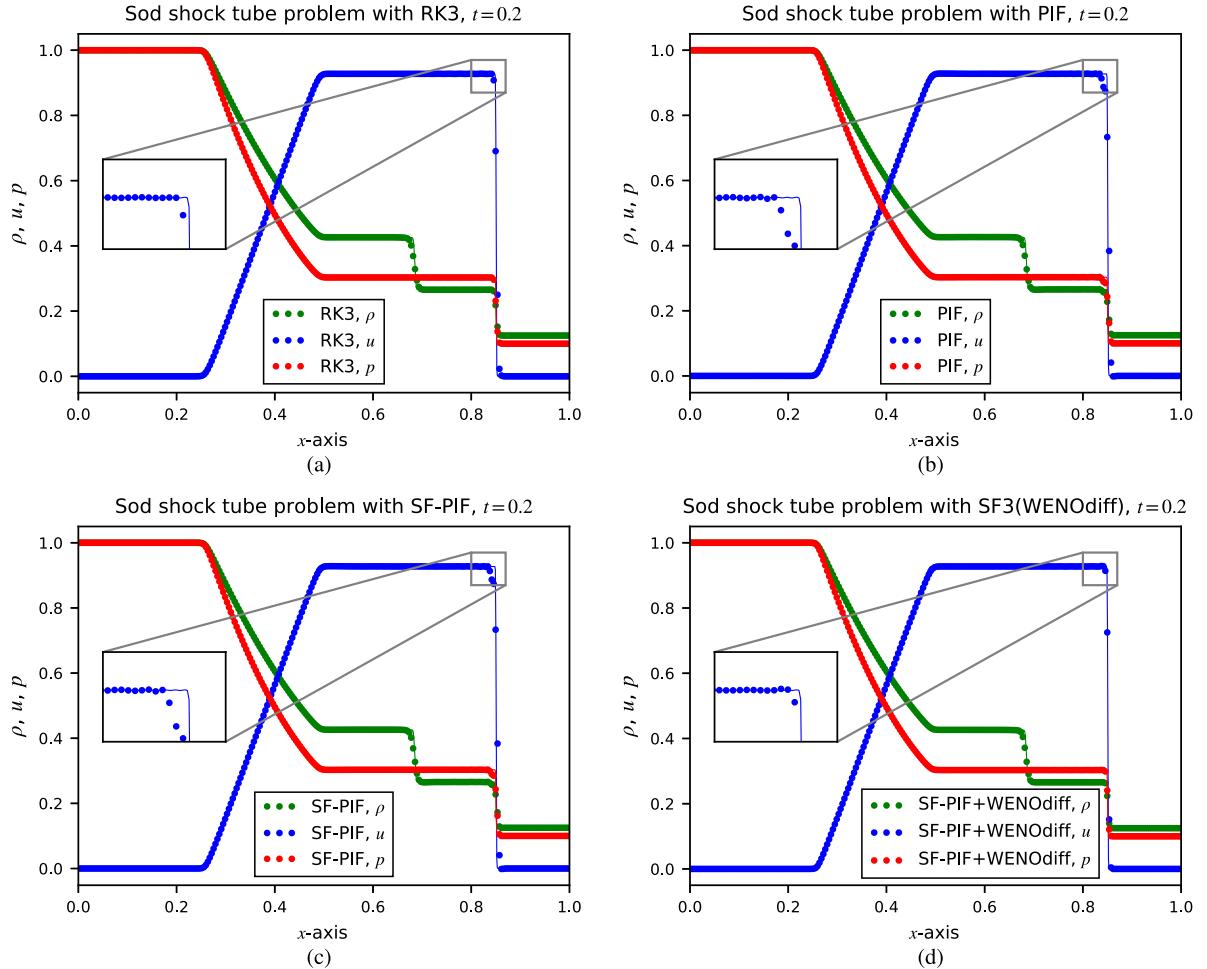
$$(\rho, u, p) = \begin{cases} (1, 0, 1) & \text{for } x \leq 0.5, \\ (0.125, 0, 0.1) & \text{for } x > 0.5, \end{cases} \quad (34)$$

in a simulation box of  $[0, 1]$ , with outflow boundary conditions on both ends at  $x = 0$  and  $x = 1$ .

The results with the grid size of  $N_x = 256$  at  $t = 0.2$  are plotted as symbols in Fig. 2. The solid lines on each panel represent the reference solution resolved on a finer grid size,  $N_x = 1024$ , by using WENO5+RK3. As seen in Fig. 2, the solutions of SF-PIF agree with the reference solution and the two solutions from RK3 and PIF. We see that, in PIF and SF-PIF, there is a slight oscillation in the  $x$ -velocity immediately behind the shock front. This small oscillation is originated from the use of the central differencing formulae in Eqs. (21) – (23) for both SF-PIF and PIF.

One way to resolve such oscillations at sharp gradients is to employ a nonlinear switch that can detect the strength of shocks and discontinuities. To meet this, we have implemented a WENO-like difference operator [4] to discretize Eq. (21) on a 5-point stencil. See Appendix B for details. As displayed in the velocity profile in Fig. 2d, the use of the WENO-differencing

<sup>2</sup> In numerical PDEs, it is a common practice to combine a spatial solver whose accuracy is higher than the accuracy of a temporal solver, with an exception of ADER schemes in which the same order of accuracy is preserved.



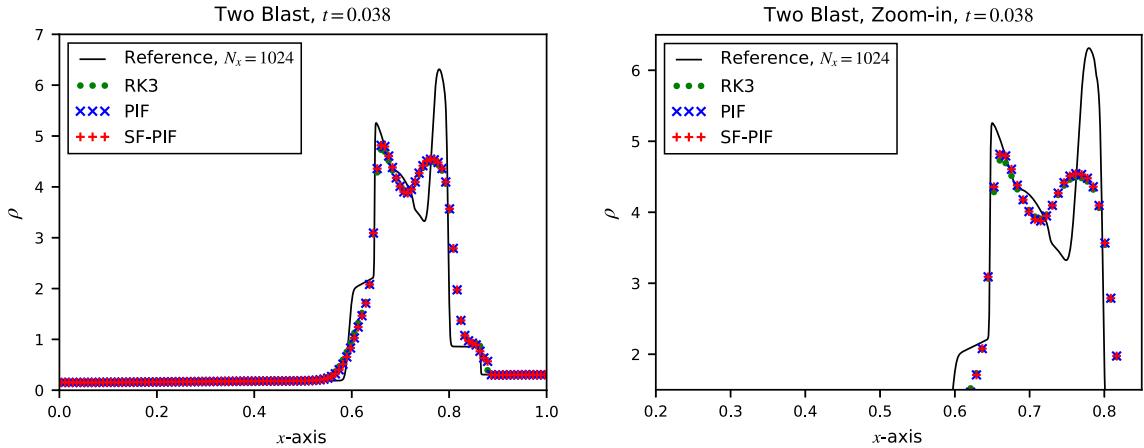
**Fig. 2.** Sod's shock tube problem at  $t = 0.2$ . The reference solutions are over-plotted as solid lines in each panel, which are resolved on a grid resolution of  $N_x = 1024$  with RK3. The symbols in each panel represent the solution resolved on  $N_x = 256$  grid cells with (a) RK3, (b) PIF, and (c) SF-PIF. In (d), the solution is resolved with SF-PIF method combining with a new WENO-like numerical differentiator operator.

helps to improve the performance of SF-PIF at the shock, suppressing the post-shock oscillations that are observed in Fig. 2b and Fig. 2c. With this fix, the SF-PIF results in Fig. 2d is almost identical to the RK results in Fig. 2a. We have found that it suffices to discretize Eq. (21) using the WENO-like differencing, retaining the central differencing for Eqs. (22) and (23). Computationally, the WENO-like differencing adds extra floating point operations, which consequently slow down the overall performance of SF-PIF. For this reason, the WENO-like discretization was employed only on the Sod's shock-tube test as a guidance, while it was opt-out on the rest of the test problems where there has not been observed any unphysical shock/discontinuity oscillations.

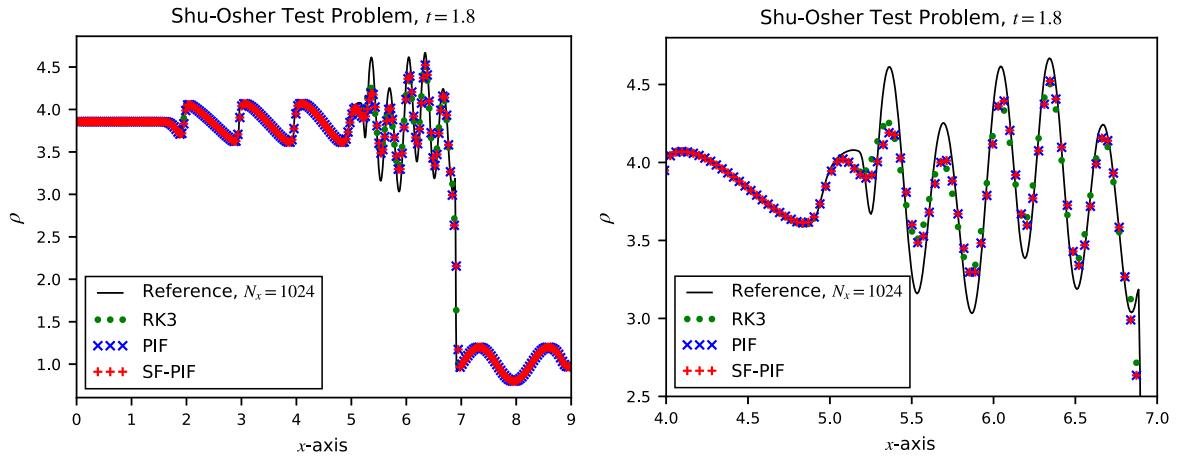
#### 4.1.3. Two-blast wave

Since originally introduced by Woodward and Colella [54], this test problem has been chosen by many practitioners to examine codes' capability of capturing the correct dynamics of shock-shock interactions. Initially, two strong shocks are developed at each end of the simulation box  $[0, 1]$ . The reflecting boundary conditions are used at both ends,  $x = 0$  and  $x = 1$ . They are driven to each other, leading to a highly compressive collision at the middle part of the domain at  $t \sim 0.028$  by the initial pressure discontinuities. The shock-shock collision produces an extremely high and narrow density peak. One of the main points of interest is to demonstrate how well numerical solutions predict the density profile and its peak amplitude particularly at  $t = 0.038$ , one hundredth of a second after the strong collision.

In Fig. 3, the density profile with SF-PIF is plotted against two other density profiles with RK3 and PIF on a grid resolution of  $N_x = 128$ . Over-plotted is the reference solution (the black solid curve) integrated with RK3 on a grid size  $N_x = 1024$  for comparison. We see that all three methods produce an acceptable quality of solutions and they are all in good agreement. As observed in the close-up view in the right panel of Fig. 3, there are slight improvements in the solutions of SF-PIF and PIF over the RK3 solution in resolving the peak density amplitudes following the sharp gradients over  $0.63 \leq x \leq 0.8$ . It is not surprising to see that the two solutions are nigh equivalent since the two methods share the same common ground in



**Fig. 3.** Two-blast wave problem at  $t = 0.038$ . The symbols depict the numerical solutions resolved on a resolution of  $N_x = 128$ , and the solid line represents the reference solution on a finer grid,  $N_x = 1024$ , with RK3 method. Right: A close-up view of the shock interaction region.



**Fig. 4.** The Shu-Osher test problem at  $t = 1.8$ . The reference solution, plotted as a black solid curve, is obtained on  $N_x = 1024$  grid resolution with RK3 method. Other solutions are resolved on  $N_x = 256$  grid resolution, represented as symbols. Right: A close-up view of the doubled frequency region to illustrate methods' numerical diffusivity.

mathematical formulation. On the other hand, the equivalency of the two methods affirms that our system-free approach described in Section 3 not only provides ease of code implementation but also guarantees high fidelity in solution accuracy compared against the analytical approach of the original PIF method.

#### 4.1.4. Shu-Osher problem

The Shu-Osher problem [55] describes an interaction of a Mach 3 shock and a smooth density profile. Initially, a right-going Mach 3 shock propagates into the low density profile with sinusoidal perturbations. As the shock proceeds, it is superposed with the sinusoidal density profile, leaving two different trails of density behind the right-propagating shock.

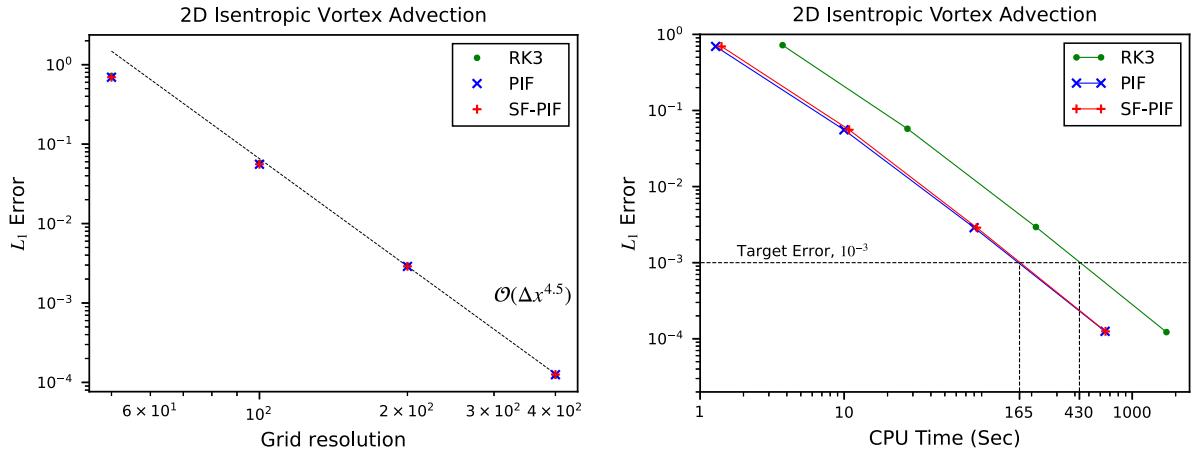
The initial sine wave gets compressed by the penetrating shock, doubling the perturbation frequency in the downstream of the immediate post-shock region. From the tail of the high frequency region to the farther left, the perturbation returns to the original frequency, but at the same time, the profile experiences a shock-steepening which leads the initial smooth profile into a sequence of sharp density gradients further away to the downstream.

Our main focus on this test problem is to check whether the SF-PIF scheme is able to advance the discrete solutions in a stable and accurate manner, capturing both the smooth and discontinuous profiles in the double-frequency region as well as at the shock front. Fig. 4 shows the SF-PIF result at  $t = 1.8$  with the other two methods of RK3 and PIF. All three results are resolved on  $N_x = 256$  grid cells, while the reference solution is obtained on a much higher resolution of  $N_x = 1024$  with RK3 method, displayed in a black solid curve. The results illustrated in the close-up view on the right show that the two single-stage methods, SF-PIF and PIF, are able to produce density peak amplitudes in the double-frequency region slightly higher than the solution from the multi-stage RK3 method. As before, the two solutions of SF-PIF and PIF are found to be almost indistinguishable, assuring the validity of the system-free formulations of the Jacobian and Hessian terms in SF-PIF compared with the analytic calculations of the same terms in the original PIF scheme.

**Table 1**

The  $L_1$  errors, the rates of convergence, and the relative computation times for the vortex advection test. Here, we display the comparison between SSP-RK3 and SF-PIF only since the difference between SF-PIF and PIF is indistinguishable. All the performance results (measured in seconds) are averaged over 10 simulation runs which are conducted on a Coffee Lake quad-core i7 Intel CPU with a clock speed of 2.7 GHz, Turbo Boost up to 4.5 GHz, utilizing four parallel threads.

$N_x = N_y$	RK3				SF-PIF			
	$L_1$ error	$L_1$ order	CPU Time	Speedup	$L_1$ error	$L_1$ order	CPU Time	Speedup
50	$7.22 \times 10^{-1}$	–	3.73 s	1.0	$6.95 \times 10^{-1}$	–	1.41 s	0.38
100	$5.76 \times 10^{-2}$	3.65	27.51 s	1.0	$5.58 \times 10^{-2}$	3.64	10.82 s	0.39
200	$2.94 \times 10^{-3}$	4.29	214.44 s	1.0	$2.89 \times 10^{-3}$	4.27	83.21 s	0.39
400	$1.22 \times 10^{-4}$	4.59	1727.71 s	1.0	$1.26 \times 10^{-4}$	4.52	652.18 s	0.38



**Fig. 5.** Left: The  $L_1$  errors of the isentropic vortex advection problem with respect to the grid resolutions, 50, 100, 200, and 400. All simulations use WENO5 for spatial reconstruction with  $C_{\text{cfl}} = 0.4$ . Right: The  $L_1$  errors of the isentropic vortex advection problem as a function of computational time.

#### 4.2. 2D Euler equations

In this section, we implement the SF-PIF scheme to solve the two-dimensional Euler equations, defined by Eq. (1) with,

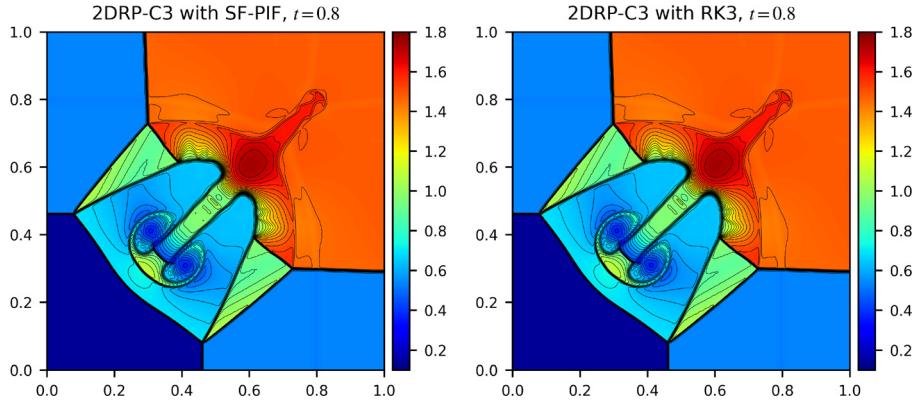
$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(E + p) \end{bmatrix}, \quad \mathbf{G}(\mathbf{U}) = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(E + p) \end{bmatrix}. \quad (35)$$

As before, WENO5 is used as the spatial method in all 2D test problems in this section so that we can focus on differing numerical effects only from the temporal methods. We present results from several 2D test problems and compare them to address the performance of the SF-PIF scheme. We choose the CFL number  $C_{\text{cfl}} = 0.4$  in all problems.

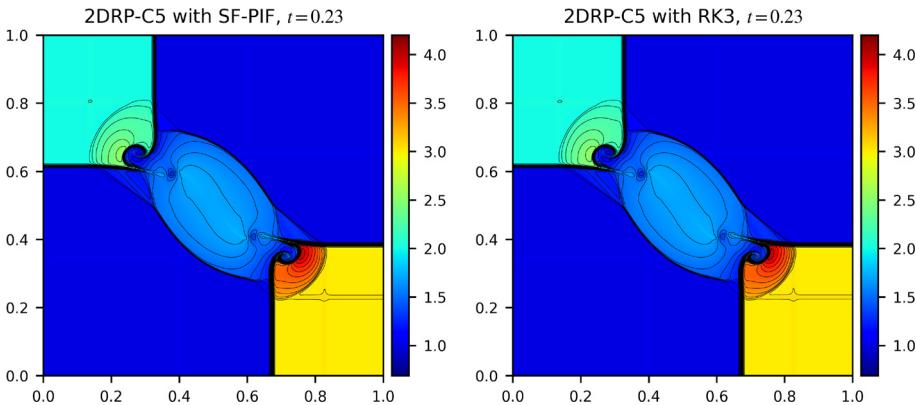
##### 4.2.1. Nonlinear isentropic vortex advection

Our first 2D problem considers SF-PIF's accuracy and speedup in two-dimensional Euler equations. We conduct the nonlinear isentropic vortex advection problem originally introduced by Shu [56]. Here, we modify the problem as presented in [57] where the size of the periodic domain  $[0, 20] \times [0, 20]$  is twice larger in each direction than the original setup in [56] to prevent self-interactions of the vortex across the periodic domain. The same consideration was also found in [45,37]. Initially positioned at the center of the domain, the isentropic vortex starts to advect in the positive diagonal direction and returns to the initial location after one periodic time,  $t = 20$ . The  $L_1$  error is calculated through the direct comparison between the initial condition and the numerical solution at  $t = 20$  to examine numerical convergence rates on four different grid resolutions,  $N_x = N_y = 50, 100, 200$ , and 400.

The results of the convergence test are summarized in Table 1 and depicted on the left panel in Fig. 5. We see a good comparable match in magnitudes of the  $L_1$  errors across all three methods. Between SF-PIF and PIF, there is no significant distinction in their accuracy and performance, as demonstrated on both panels in Fig. 5. All three methods converge at 4.5th order towards the initial condition at each grid resolution. Unlike the 1D linear sine advection test in Section 4.1.1, we do not see the full dominance in error from the third-order temporal discretization (at least over the range of the grid resolutions tested herein), which could potentially reduce the overall convergence rate down to third-order as in the sine advection case. Concurrently, we also see that the solution does not converge at full fifth-order either, the rate of which is due from the use of WENO5. This can be explained as a nonlinear effect in which the overall leading error term of the



**Fig. 6.** The density profile of Configuration 3 with SF-PIF (left) and with RK3 (right). The color map ranges from 0.1 to 1.8, and 40 evenly-spaced contour lines are over-plotted with the same range.



**Fig. 7.** The density profile of Configuration 5 with SF-PIF (left) and with RK3 (right). The color map ranges from 0.69 to 4.2, and 40 evenly-spaced contour lines are over-plotted with the same range.

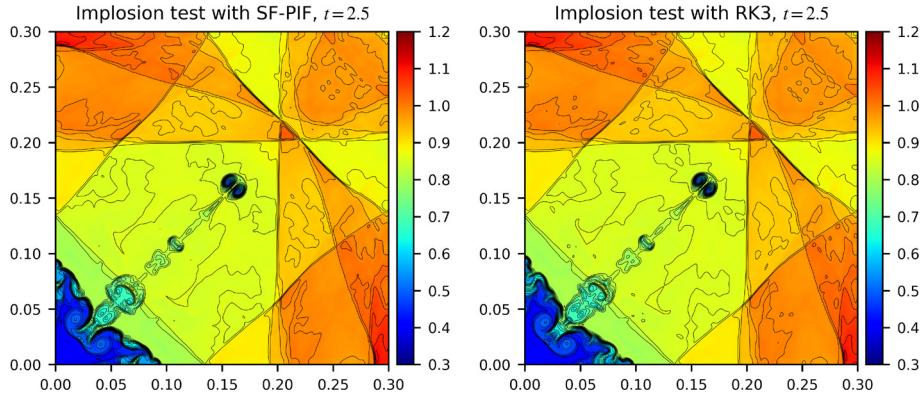
fifth-order spatial discretization is slightly compromised by the lower third-order time integration schemes. In general, an exact analysis of this phenomenon is highly problem dependent and it often becomes intractable in many nonlinear cases. In practice though, this issue can be improved by one of the two approaches, timestep reductions, or developments of fourth-order or higher temporal methods. As discussed, the first option is not favorable in most large-scale simulations, and hence the latter is to be considered as a better alternative from the mathematical standing point.

On the right panel, we illustrate the performance of SF-PIF measured in terms of the CPU time to reach a target error threshold, say  $L_1$  error of  $10^{-3}$ . As demonstrated, the results of the two single-stage methods, SF-PIF and PIF, coincide to each other and their performance is approximately twice faster than the multi-stage method of RK3. It is worth to be noted that the SF-PIF approach can be readily swappable with an RK integrator in an existing code without too much effort, leaving any existing spatial implementations intact mostly. Moreover, such a code transformation with SF-PIF is more advantageous in simplicity than with the original PIF method because SF-PIF replaces PIF's need for analytic derivations of the Jacobian and Hessian terms with the system-free approximations, which have shown to be highly commensurate with the analytical counterparts of the PIF scheme.

#### 4.2.2. 2D Riemann problems

Next, we consider 2D Riemann problems which are extensively studied in [58–60] and widely used for the code verifications [45,61–63,37]. Among a wide range of different configurations, we demonstrate two test cases, Configuration 3 and Configuration 5, following the descriptions in [63,45]. Both test cases are resolved on a square simulation box of  $[0, 1] \times [0, 1]$  with  $400 \times 400$  grid cells using outflow boundary conditions.

The results of Configuration 3 and Configuration 5 are shown in Fig. 6 and Fig. 7, respectively. The density maps are visualized with pseudo-color schemes ranging between 0.1 and 1.8 in Configuration 3, and between 0.69 and 4.2 in Configuration 5. Over-plotted are 40 levels of contour lines using the same density range in each configuration. We see that the SF-PIF result is well comparable to that of RK3, which assures that SF-PIF method is eminently reliable in temporally advancing solutions with discontinuities and shocks in 2D, while at the same time, preserving the expected symmetries and flow profiles in each problem. In both, however, there is yet no formation of secondary Kelvin-Helmholtz instabilities, typically exhibited as vortical rollups along the slip lines – the regions connecting the inner edges of the triangle “arm”



**Fig. 8.** The density profile of the implosion test with SF-PIF (left) and with RK3 (right). The color map ranges from 0.3 to 1.2, and 40 evenly-spaced contour lines are over-plotted with the same range.

s" surrounding the mushroom-jet and the stem of the mushroom-jet. Direct comparisons with other FDM test results on the same grid resolution with comparable accuracy (e.g., Fig. 12 in [37]; Fig. 6 in [63]) reveal that their test results have more pronounced rollup structures developed along the slip lines. This implies that both of the test results here have more numerical diffusivity than those approaches therein [63,37]. We first emphasize that developments of small-scale features relevant to grid scales depend much more sensitively on a combination of the choice of spatial solvers (e.g., WENO5 with the global Rusanov flux-splitting in our case) and grid resolutions, but less on a temporal solver. It is well-known that the global Rusanov flux-splitting (used in this study but not in [63,37]) has added numerical diffusivity, which can delay an onset of the secondary instability formation in our experiments. Secondly, with mathematical rigor, an increasing number of rollups in any given scheme should only be understood as a proof of a *less* amount of numerical dissipation, but not as blanket justification of a broader concept of being a *better* numerical method than others because less numerical dissipation in an algorithm usually reduces the stability of the algorithm.

Measuring a *proper* amount of such rollups in any given simulations on a set of specific setups (i.e., grid resolutions, spatial and temporal orders of numerical methods, parameters, etc.) is not a simple task. Such a study will need to involve an extensive systemic comparison analysis that requires careful validation and verification tests, the topics of which are beyond the scope of this paper.

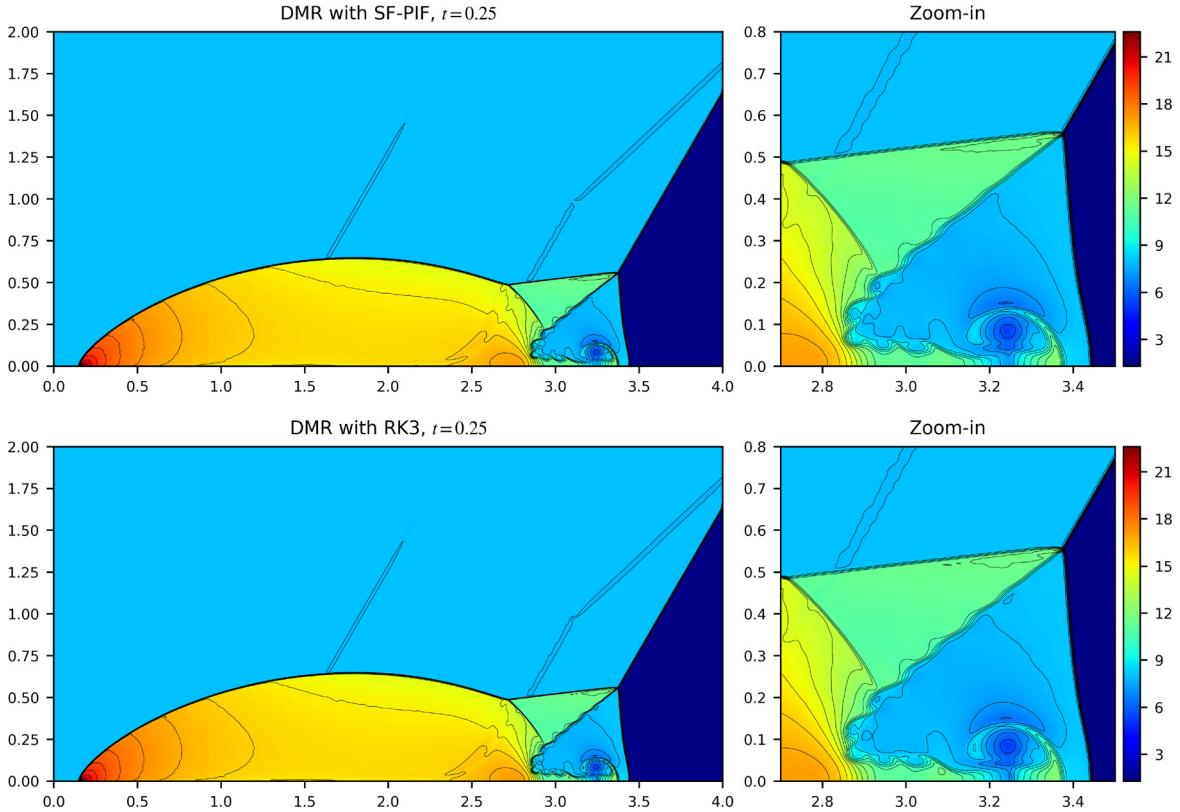
#### 4.2.3. Implosion test

The next problem to consider is the implosion test problem introduced by Hui et al. [64]. An unsteady flow configuration is given as an initial condition which launches a converging shock wave towards the domain center. We follow a simpler version by Liska and Wendroff [65], which takes only the right upper quadrant of the original setup in [64] as the simulation domain. In this setup, the simulation is initialized on a region of a square domain,  $[0, 0.3] \times [0, 0.3]$ , enclosed with reflecting walls, in which case a converging shock wave is launched toward the lower left corner at  $(x, y) = (0, 0)$ . The initial shock wave gets bounced by the reflecting walls and produces a double Mach reflection along two edges of  $x = 0$  and  $y = 0$ . As a consequence, two jets are formed along the edges moving toward the origin  $(x, y) = (0, 0)$  and collide each other. This two-jet collision then ejects a newly-formed jet into the diagonal direction  $x = y$ . Reflecting shocks continuously interact with the diagonal jet, turning it into a longer and narrower shape over time. The observed structures of filaments and fingers along the diagonal jet and at its base are progressively intensified by the Ritchmyer-Meshkov instability, a level of which depends sensitively on numerical dissipation. The shape of the jet is the key view point of the implosion test since it is a good indicator of a code's symmetric property and numerical dissipation. If the numerical scheme fails to maintain a high level of symmetry, the jet will eventually be derailed off-diagonally and deformed over time. Besides, an excess amount of numerical dissipation will turn the jet into a less narrow and less elongated shape along the diagonal.

We display our results on a  $400 \times 400$  grid resolution at  $t = 2.5$ . The result with SF-PIF is on the left panel in Fig. 8 and RK3 on the right. These results can also be directly compared with Fig. 4.7 in [65] and Fig. 17 in [66]. We clearly see that the contour lines of SF-PIF (as well as RK3) are shown to retain the diagonal symmetry at a highly sufficient level. At the same time, the shape of the diagonal jet using SF-PIF matches well with the shape using RK3, and hence is sufficient to demonstrate that the numerical dissipation in SF-PIF is well-managed compared with RK3.

#### 4.2.4. Double Mach reflection

Our last test using the Euler equations in 2D is the double Mach reflection problem introduced by Woodward and Colella [54]. We follow the original setup except that we doubled the domain size in the  $y$ -direction to avoid any disturbances by the numerical artifacts from the top boundary [67]. A Mach 10 planar shock is initialized at the left side of the domain with a  $30^\circ$  angle to the reflecting bottom surface. As the shock propagates to the right, the bottom wall continuously bounces off the shock wave and creates a round reflected shock. The solution further evolves into forming two contact



**Fig. 9.** Left column: The density profile of the double Mach reflection (DMR) problem solved with SF-PIF (top) and RK3 (bottom). Right column: Close-up views near the triple-point of the density profile. The contour lines display 30 linearly separated points. All plots use the same range of density, [1.3, 22.6]. All calculations are conducted on a grid resolution of  $1024 \times 512$  with  $C_{\text{cfl}} = 0.4$ .

discontinuities and two Mach stems, as well as a jet along the bottom surface. The formation of this jet is similar to the formation of the two jets in the previous implosion test, the collision of which led to the upward moving diagonal jet.

The main point of discussion is to observe the appearance of the contact discontinuity that spans from the triple-point at  $(x, y) \approx (3.38, 5.5)$  in Fig. 9 to the dense jet along the bottom wall, displayed in the panels on the right column in Fig. 9. A lower amount of numerical dissipation in a scheme makes this region more susceptible to Kelvin-Helmholtz instability, leading to an onset of vortical roll-ups along the slip line. Henceforth, the amount of such roll-ups serves as an indicator of the scheme's numerical diffusivity. As discussed in the 2D Riemann problems in Section 4.2.2, we once more point out that such an assessment should only be used to address each scheme's numerical diffusivity, but not to be used to rank different schemes without careful validation and verification studies.

The density results are represented in Fig. 9, and the detailed views are shown in the right panel. Shown in the top row in Fig. 9 is the density profile integrated using SF-PIF in the range of [1.3, 22.6] at  $t = 0.25$ , computed on a grid resolution of  $1024 \times 512$ . We also plot 30 evenly spaced contour lines of density in the same range. The result with RK3 is in the bottom row. We can see that the time prediction of density using SF-PIF matches sufficiently well with the result using RK3, confirming the validity of SF-PIF in comparison.

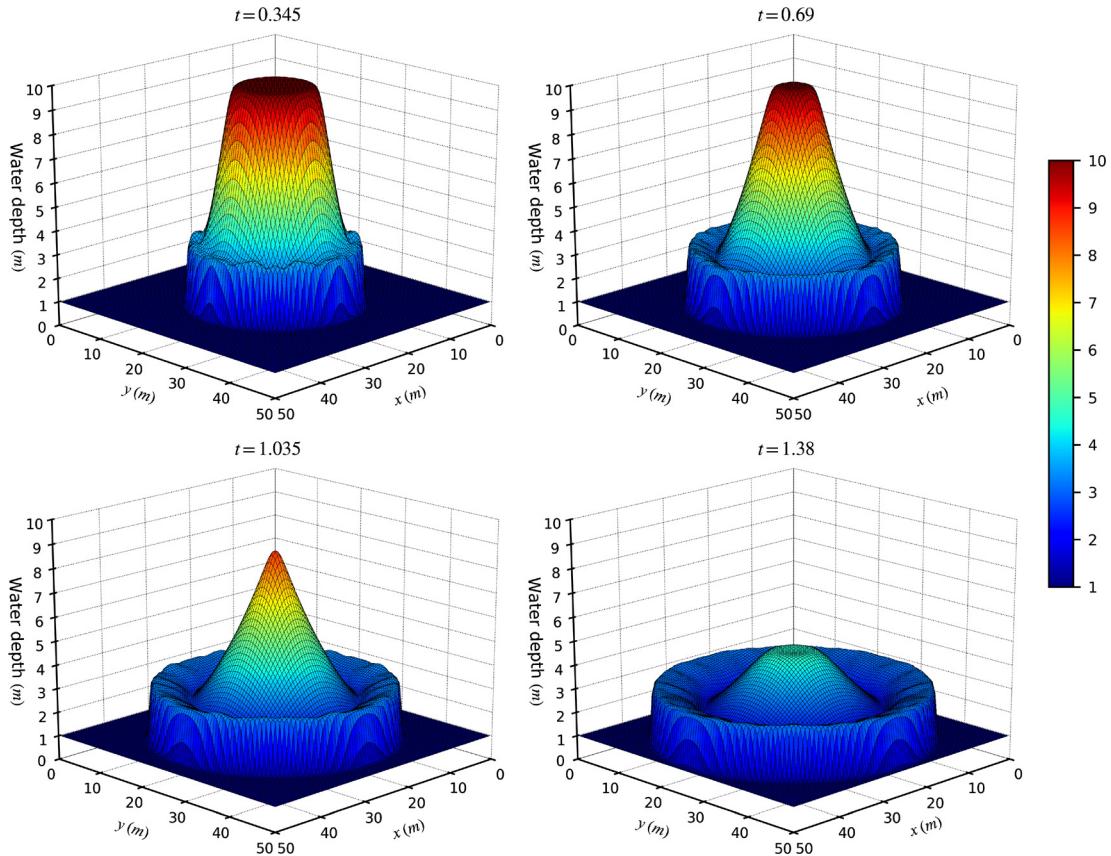
#### 4.3. Shallow water equations

In this section, we switch the governing equations to a system of 2D shallow water equations (SWE) without a source term, defined by a conservative hyperbolic system as,

$$\mathbf{U} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix}, \quad \mathbf{G}(\mathbf{U}) = \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix}. \quad (36)$$

Here,  $h$  is the vertical depth of the fluid,  $\mathbf{v} = (u, v)$  is a vector of vertically-averaged velocity components in  $x$  and  $y$  directions. Denoted as  $g$  is a gravitational acceleration in the negative vertical  $z$  direction, which is averaged out in the derivation of the shallow water equations.

The sole purpose of presenting this new system of equations is to demonstrate the flexibility of our SF-PIF scheme, in that the system-free approach allows an easy code implementation without the need for another analytical calculation of



**Fig. 10.** Snapshots of the circular dam breaking simulation at four different times,  $t = 0.345, 0.69, 1.035$ , and  $1.38$  seconds. A volume of water in rest is initially confined in a cylindrical dam of a radius  $r = 11$  m and a height  $h = 10$  m. The simulation starts with an instantaneous removal of the cylindrical wall located at the center of the domain  $[50 \text{ m} \times 50 \text{ m}]$  resolved on a grid resolution of  $100 \times 100$ . The gravitational force is exerted on the steady water, which triggers the onset of the gravitational collapse of the entire volume of water, making the circular splash in the outer rim as well as the ripple effects in the central region that move radially outward in time.

new Jacobian and Hessian terms of the new governing system. By the system-independent property of the SF-PIF method, the process of changing from the 2D Euler code to the 2D SWE code is no more than switching the governing equations. This process is much simpler than what is needed in the original PIF method, enabling the overall code transition is highly smooth and transparent without any extra effort to modify a substantial portion of code lines.

We conduct a simple circular dam breaking simulation, which has been widely adopted for code validation purposes in many SWE literatures [68–70]. Initially, a volume of still water is confined in the virtual (i.e., invisible) cylindrical wall with a radius of 11 meters (m), located at the center of simulation domain,  $[50 \text{ m} \times 50 \text{ m}]$  resolved on a  $100 \times 100$  grid resolution. The depth of the water inside of the wall is 10 m and 1 m outside. This configuration would be considered as an SWE version of 2D Riemann problem. Explicitly, the initial condition is given as,

$$(h, u, v) = \begin{cases} (10 \text{ m}, 0, 0) & \text{for } r \leq 11 \text{ m}, \\ (1 \text{ m}, 0, 0) & \text{for } r > 11 \text{ m}, \end{cases} \quad (37)$$

where  $r$  is the distance from the center of domain,  $r = \sqrt{(x - 25 \text{ m})^2 + (y - 25 \text{ m})^2}$ . The outflow boundary condition is applied for both directions, and we set the gravitational acceleration  $g = 9.81 \text{ m s}^{-2}$ . Again, the Courant number is set to  $C_{\text{cf}} = 0.4$ .

As shown in Fig. 10, the results with SF-PIF are well comparable with the results using the same configuration reported in [68–70]. We conclude that the overall spherical symmetry and the sharp profile at the wave front are well maintained in the snapshots at four different times,  $t = 0.345, 0.69, 1.035$ , and  $1.38$ .

#### 4.4. Performance comparison with SF-PIF and RK3

In this section, we present a summary of the performance comparison between SF-PIF and RK3. The comparison was conducted on five test setups selected from the previous sections, including one smooth flow test of the isentropic vortex

**Table 2**

Performance results for various test problems. All the performance results (measured in seconds) are averaged over 10 simulation runs which are conducted on a Coffee Lake quad-core i7 Intel CPU with a clock speed of 2.7 GHz, Turbo Boost up to 4.5 GHz, utilizing four parallel threads.

Test Problem	Grid Resolution	RK3		SF-PIF	
		CPU Time	Speedup	CPU Time	Speedup
Vortex advection	$400 \times 400$	1727.71 s	1.0	652.18 s	0.38
2DRP-C3	$400 \times 400$	1205.60 s	1.0	486.79 s	0.40
2DRP-C5	$400 \times 400$	377.26 s	1.0	142.12 s	0.38
DMR	$1024 \times 512$	6458.33 s	1.0	2610.05 s	0.40
Dam breaking	$100 \times 100$	3.27 s	1.0	1.59 s	0.49

advection in Section 4.2.1 and four discontinuous flow tests of the 2D Riemann problems in Section 4.2.2, the double Mach reflection problem in Section 4.2.4, and finally the circular dam breaking problem in Section 4.3. The same setup parameters (e.g., simulation times, Courant numbers) are retained in each run.

The results are summarized in Table 2, where the tested grid resolutions are displayed for exposition purpose. Throughout the problems tested herein, the performance of SF-PIF is consistently at least twice faster than the standard three-stage SSP-RK3 integrator on both smooth and discontinuous problems. We conclude that this factor of two in performance gain in SF-PIF over SSP-RK3 remains as strong evidence that characterizes the computational efficiency of our SF-PIF algorithm, regardless of differing flow conditions with or without shocks and discontinuities.

## 5. Conclusion

In this paper, we have improved the flexibility of the PIF method in [40] by introducing a new Jacobian-free and Hessian-free approach. With this improvement, referred to as the system-free (SF) approach, the resulting SF-PIF method can be readily implemented in any existing RK-based FDM codes by swapping any multi-stage RK integrators with the single step, third-order accurate SF-PIF integrator.

The major advantage of SF-PIF lies in ease of its code implementation for practical use. In particular, SF-PIF can be applied easily to a different hyperbolic system of equations without hassle by virtue of our system-independent formulation of Jacobian-vector and Hessian-vector-vector multiplications, which operationally replaces the analytical expressions of the Jacobians and Hessians terms in the original PIF method.

In the present study, we have tested our SF-PIF algorithm to solve a wide range of well-known benchmark test problems in one and two spatial dimensions. The test results show that the solution accuracy and stability of SF-PIF are proved to be equally comparable with the solutions computed using arguably the most popular choice of the three-stage SSP-RK3 solver. Moreover, compared to RK3, SF-PIF achieves a twice fast code performance gain while maintaining the same third-order accuracy. We also have demonstrated that the SF-PIF method delivers the same order of accuracy as the original PIF method, while simplifying the calculations of the Jacobian and Hessian terms. The two methods are found to be nigh indistinguishable in their solution accuracy, stability, and performance. Mathematically speaking, the fact they are identical assures that the system-free approximation in SF-PIF is highly reliable in the Jacobian and Hessian term calculations in comparison with the analytical calculation in PIF. Practically speaking, as the SF-PIF method does not sensitively rely on a system of governing equations, it provides an increased adaptability to effortlessly switch to a different system of equations. This adaptability has been fully demonstrated in terms of switching the existing Euler equations code to a code for the shallow water equations in Section 4.3.

An extension of the current work is to design a fourth-order (or higher) SF-PIF method. However, a direct extension to a fourth-order scheme requires complex vector calculations and evaluations of the next high derivative terms, higher than the Hessian term in the Taylor expansion. Such a naive extension will be inevitably facing an increasing complexity in code implementations, and hence resulting the code performance less attractive in comparison with the use of conventional RK-based solvers. Reducing such complexities in designing a fourth- or higher-order SF-PIF method is essential, and a relevant work will be further investigated in our future studies.

## CRediT authorship contribution statement

**Youngjun Lee:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Dongwook Lee:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Writing – original draft, Writing – review & editing.

## Declaration of competing interest

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

## Acknowledgements

We acknowledge use of the lux supercomputer at UC Santa Cruz, funded by NSF MRI grant AST 1828315. The authors thank Dr. Adam C. Reyes for very helpful discussions and feedback during the early stage of the development.

## Appendix A. The optimal choices of $\epsilon$

We follow the analysis in [50] to determine the optimal value of  $\epsilon_{\text{Jac}}^{\text{op}}$  for the Jacobian-free approximation. Let  $\mathbf{F}(\mathbf{U})$  be an exact function value of  $\mathbf{F}$  at  $\mathbf{U}$ , and let  $\mathbf{F}^*(\mathbf{U}) = \mathbf{F}(\mathbf{U}) + \delta\mathbf{F}(\mathbf{U})$  be an approximation to  $\mathbf{F}(\mathbf{U})$ , where  $\delta\mathbf{F}(\mathbf{U})$  is a perturbation of  $\mathbf{F}(\mathbf{U})$  that is potentially due from roundoff errors and truncation errors and is assumed to be bounded by the machine accuracy, i.e.,  $\|\delta\mathbf{F}(\mathbf{U})\| \leq \epsilon_{\text{mach}} \sim 2.2204 \times 10^{-16}$ . The main idea is to choose  $\epsilon_{\text{Jac}}^{\text{op}}$  in such a way that the error is minimized when approximating  $\frac{\partial\mathbf{F}}{\partial\mathbf{U}} \cdot \mathbf{V}$  using the central differencing approximation of  $\mathbf{F}^*(\mathbf{U})$  in Eq. (25), i.e.,

$$\frac{\partial\mathbf{F}}{\partial\mathbf{U}} \cdot \mathbf{V} \approx \frac{\mathbf{F}^*(\mathbf{U} + \sigma\mathbf{V}) - \mathbf{F}^*(\mathbf{U} - \sigma\mathbf{V})}{2\sigma} = \frac{\mathbf{F}(\mathbf{U} + \sigma\mathbf{V}) + \delta\mathbf{F}(\mathbf{U} + \sigma\mathbf{V}) - \mathbf{F}(\mathbf{U} - \sigma\mathbf{V}) - \delta\mathbf{F}(\mathbf{U} - \sigma\mathbf{V})}{2\sigma}. \quad (\text{A.1})$$

For the sake of this analysis, we assume that the function  $\mathbf{F}: \mathbb{R}^n \rightarrow \mathbb{R}^n$  is defined to be continuously differentiable sufficiently everywhere,  $\mathbf{F} \in C^k(\mathbb{R}^n)$ . We now define the error  $E$  by the difference between the central differencing approximation in Eq. (A.1) and  $\frac{\partial\mathbf{F}}{\partial\mathbf{U}} \cdot \mathbf{V}$ ,

$$\begin{aligned} E &= \frac{\mathbf{F}^*(\mathbf{U} + \sigma\mathbf{V}) - \mathbf{F}^*(\mathbf{U} - \sigma\mathbf{V})}{2\sigma} - \frac{\partial\mathbf{F}}{\partial\mathbf{U}} \cdot \mathbf{V} \\ &= \frac{\mathbf{F}(\mathbf{U} + \sigma\mathbf{V}) - \mathbf{F}(\mathbf{U} - \sigma\mathbf{V})}{2\sigma} + \frac{\delta\mathbf{F}(\mathbf{U} + \sigma\mathbf{V}) - \delta\mathbf{F}(\mathbf{U} - \sigma\mathbf{V})}{2\sigma} - \frac{\partial\mathbf{F}}{\partial\mathbf{U}} \cdot \mathbf{V} \\ &= \frac{1}{2\sigma} \left[ 2\sigma \frac{\partial\mathbf{F}}{\partial\mathbf{U}} \cdot \mathbf{V} + \sigma^3 \int_0^1 (1-t)^2 \mathbf{F}^{(3)}(\mathbf{U} + t\sigma\mathbf{V}) \cdot \mathbf{V}^3 dt \right] + \frac{\delta\mathbf{F}(\mathbf{U} + \sigma\mathbf{V}) - \delta\mathbf{F}(\mathbf{U} - \sigma\mathbf{V})}{2\sigma} - \frac{\partial\mathbf{F}}{\partial\mathbf{U}} \cdot \mathbf{V} \\ &= \mathcal{O}\left(\frac{\sigma^2}{2} + \frac{\epsilon_{\text{mach}}}{2\sigma}\right), \end{aligned} \quad (\text{A.2})$$

where we have used the Taylor series expansion around  $\mathbf{U}$  for each term in which the remainders after the third power are given as the integral form as below,

$$\begin{aligned} \mathbf{F}(\mathbf{U} + \sigma\mathbf{V}) - \mathbf{F}(\mathbf{U} - \sigma\mathbf{V}) &= \mathbf{F}(\mathbf{U}) + \sigma \frac{\partial\mathbf{F}}{\partial\mathbf{U}} \cdot \mathbf{V} + \frac{\sigma^2}{2} \frac{\partial^2\mathbf{F}}{\partial\mathbf{U}^2} \cdot \mathbf{V} \cdot \mathbf{V} + \frac{\sigma^3}{2} \int_0^1 (1-t)^2 \mathbf{F}^{(3)}(\mathbf{U} + t\sigma\mathbf{V}) \cdot \mathbf{V}^3 dt \\ &\quad - \mathbf{F}(\mathbf{U}) + \sigma \frac{\partial\mathbf{F}}{\partial\mathbf{U}} \cdot \mathbf{V} - \frac{\sigma^2}{2} \frac{\partial^2\mathbf{F}}{\partial\mathbf{U}^2} \cdot \mathbf{V} \cdot \mathbf{V} + \frac{\sigma^3}{2} \int_0^1 (1-t)^2 \mathbf{F}^{(3)}(\mathbf{U} + t\sigma\mathbf{V}) \cdot \mathbf{V}^3 dt \\ &= 2\sigma \frac{\partial\mathbf{F}}{\partial\mathbf{U}} \cdot \mathbf{V} + \sigma^3 \int_0^1 (1-t)^2 \mathbf{F}^{(3)}(\mathbf{U} + t\sigma\mathbf{V}) \cdot \mathbf{V}^3 dt. \end{aligned} \quad (\text{A.3})$$

The optimal choice of  $\epsilon_{\text{Jac}}^{\text{op}}$  is to be obtained by considering the minimization problem of the leading error term in the last line in Eq. (A.2), resulting in the first expression in Eq. (28).

A similar approach is to follow for the Hessian-free approximation to minimize the error  $E$ ,

$$\begin{aligned} E &= \frac{\mathbf{F}^*(\mathbf{U} + \sigma\mathbf{V}) - 2\mathbf{F}^*(\mathbf{U}) + \mathbf{F}^*(\mathbf{U} - \sigma\mathbf{V})}{\sigma^2} - \frac{\partial^2\mathbf{F}}{\partial\mathbf{U}^2} \cdot \mathbf{V} \cdot \mathbf{V} \\ &= \frac{\mathbf{F}(\mathbf{U} + \sigma\mathbf{V}) - 2\mathbf{F}(\mathbf{U}) + \mathbf{F}(\mathbf{U} - \sigma\mathbf{V})}{\sigma^2} + \frac{\delta\mathbf{F}(\mathbf{U} + \sigma\mathbf{V}) - 2\delta\mathbf{F}(\mathbf{U}) + \delta\mathbf{F}(\mathbf{U} - \sigma\mathbf{V})}{\sigma^2} - \frac{\partial^2\mathbf{F}}{\partial\mathbf{U}^2} \cdot \mathbf{V} \cdot \mathbf{V} \\ &= \frac{1}{\sigma^2} \left[ \sigma^2 \frac{\partial^2\mathbf{F}}{\partial\mathbf{U}^2} \cdot \mathbf{V} \cdot \mathbf{V} + \frac{\sigma^4}{3} \int_0^1 (1-t)^3 \mathbf{F}^{(4)}(\mathbf{U} + t\sigma\mathbf{V}) \cdot \mathbf{V}^4 dt \right] \\ &\quad + \frac{\delta\mathbf{F}(\mathbf{U} + \sigma\mathbf{V}) - 2\delta\mathbf{F}(\mathbf{U}) + \delta\mathbf{F}(\mathbf{U} - \sigma\mathbf{V})}{\sigma^2} - \frac{\partial^2\mathbf{F}}{\partial\mathbf{U}^2} \cdot \mathbf{V} \cdot \mathbf{V} \\ &= \mathcal{O}\left(\frac{\sigma^2}{3} + \frac{\epsilon_{\text{mach}}}{\sigma^2}\right). \end{aligned} \quad (\text{A.4})$$

As before, we obtain  $\varepsilon_{\text{hes}}^{\text{op}}$  in the second expression in Eq. (28) by minimizing the leading error term in Eq. (A.4).

## Appendix B. WENO-like numerical differentiation

In order to minimize unforeseen oscillations at shock fronts in shock-dominant problems (e.g., see Fig. 2) with SF-PIF and PIF method, we present a new numerical difference operator that adopts the nonlinear discontinuity handling of the WENO method [4] in discretizing the first derivatives of flux  $\frac{\partial \mathbf{F}_i}{\partial x}$  (similarly for  $\frac{\partial \mathbf{G}_j}{\partial y}$  in 2D) on the 5-point WENO stencil, instead of the naive fourth-order central differencing approximation as presented in Eq. (21). This WENO-like discretization provides an added feasibility to capture the shock front better as shown in Fig. 2d. Our numerical experiments also show that discretizing the first derivative flux terms using the WENO-like discretization is sufficient for suppressing numerical oscillations near shocks and discontinuities, retaining the standard central differencing for the second derivative terms in Eq. (22) since their coefficient (i.e.,  $\frac{\Delta t^2}{3!}$  in Eq. (16)) is much smaller than the one for the first derivative terms (i.e.,  $\frac{\Delta t}{2!}$  in Eq. (16)). Our numerical tests indicate that the WENO-like differencing increases the overall computational loads of SF-PIF by 12% in 1D and 18% in 2D.

Alike the fifth-order WENO reconstruction scheme, we consider the three central differencing schemes within three different small sub-stencils, i.e.,  $[i-3+s : i-1+s]$ ,  $s = 1, 2, 3$ , each of which represents the first-order differentiation of  $u$  at  $x = x_i$ ,

$$\begin{aligned} d_1 &= \frac{1}{2\Delta x} (u_{i-2} - 4u_{i-1} + 3u_i), \\ d_2 &= \frac{1}{2\Delta x} (-u_{i-1} + u_{i+1}), \\ d_3 &= \frac{1}{2\Delta x} (-3u_i + 4u_{i+1} - u_{i+2}). \end{aligned} \quad (\text{B.1})$$

We aim to approximate  $du/dx|_{x=x_i}$  as a convex combination of  $d_s$  with the nonlinear weights  $\omega_s$ , where  $s = 1, 2, 3$ ,

$$\frac{du}{dx} \Big|_{x=x_i} \approx \omega_1 d_1 + \omega_2 d_2 + \omega_3 d_3. \quad (\text{B.2})$$

In a smooth region, following the original context of WENO, we anticipate the nonlinear weights  $\omega_s$  are reduced to the linear weights  $\gamma_s$  so that the convex combination represents the approximation of  $du/dx$  in a bigger 5-point stencil. That is to say, the convex combination with the linear weights  $\gamma_s$  should be equal to Eq. (21),

$$\gamma_1 d_1 + \gamma_2 d_2 + \gamma_3 d_3 = \frac{1}{12\Delta x} (u_{i-2} - 8u_{i-1} + 8u_{i+1} - u_{i+2}). \quad (\text{B.3})$$

In this case, the linear weights are given as,

$$\gamma_1 = \frac{1}{6} \quad \gamma_2 = \frac{4}{6} \quad \gamma_3 = \frac{1}{6}. \quad (\text{B.4})$$

The nonlinear weights are calculated in the same way of the classical WENO reconstruction method [4] with the same choice of the smoothness indicators  $\beta_s$ ,  $s = 1, 2, 3$ ,

$$\omega_s = \frac{\tilde{\omega}_s}{\sum_{s=1}^3 \tilde{\omega}_s}, \quad \text{where} \quad \tilde{\omega}_s = \frac{\gamma_s}{(\epsilon + \beta_s)^m}, \quad (\text{B.5})$$

where the  $\epsilon$  is a small positive number used to avoid the division by zero, for which we choose  $\epsilon = 1.0 \times 10^{-36}$ . The parameter  $m$  determines the rate of changes of  $\beta_s$ , and we choose  $m = 1$  for capturing the discontinuities sharper, as reported in [45].

## References

- [1] Norbert Attig, Paul Gibbon, Th. Lippert, Trends in supercomputing: the European path to exascale, *Comput. Phys. Commun.* 182 (9) (2011) 2041–2046.
- [2] ASCAC Subcommittee, et al., Top ten exascale research challenges, US Department of Energy Report, 2014.
- [3] Phillip Colella, Paul R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, *J. Comput. Phys.* 54 (1) (1984) 174–201.
- [4] Guang-Shan Jiang, Chi-Wang Shu, Efficient implementation of weighted ENO schemes, *J. Comput. Phys.* 126 (1) (1996) 202–228.
- [5] Sigal Gottlieb, Chi-Wang Shu, Total variation diminishing Runge-Kutta schemes, *Math. Comput. Am. Math. Soc.* 67 (221) (1998) 73–85.
- [6] Sigal Gottlieb, Chi-Wang Shu, Eitan Tadmor, Strong stability-preserving high-order time discretization methods, *SIAM Rev.* 43 (1) (2001) 89–112.
- [7] Sigal Gottlieb, David I. Ketcheson, Chi-Wang Shu, Strong Stability Preserving Runge-Kutta and Multistep Time Discretizations, World Scientific, 2011.
- [8] Dinshaw S. Balsara, Chi-Wang Shu, Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy, *J. Comput. Phys.* 160 (2) (2000) 405–452.
- [9] Andrea Mignone, Petros Tzeferacos, Gianluigi Bodo, High-order conservative finite difference GLM-MHD schemes for cell-centered MHD, *J. Comput. Phys.* 229 (17) (2010) 5896–5920.

[10] Raymond J. Spiteri, Steven J. Ruuth, A new class of optimal high-order strong-stability-preserving time discretization methods, *SIAM J. Numer. Anal.* 40 (2) (2002) 469–491.

[11] Raymond J. Spiteri, Steven J. Ruuth, Non-linear evolution using optimal fourth-order strong-stability-preserving Runge-Kutta methods, *Math. Comput. Simul.* 62 (1–2) (2003) 125–135.

[12] Steven J. Ruuth, Raymond J. Spiteri, Two barriers on strong-stability-preserving time discretization methods, *J. Sci. Comput.* 17 (1–4) (2002) 211–220.

[13] Andrew Giuliani, Lilia Krivodonova, On the optimal CFL number of SSP methods for hyperbolic problems, *Appl. Numer. Math.* 135 (2019) 165–172.

[14] E.F. Toro, R.C. Millington, L.A.M. Nejad, Towards very high order Godunov schemes, in: *Godunov Methods*, Springer, 2001, pp. 907–940.

[15] Bram Van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, *J. Comput. Phys.* 32 (1) (1979) 101–136.

[16] Matania Ben-Artzi, Joseph Falcovitz, A second-order Godunov-type scheme for compressible fluid dynamics, *J. Comput. Phys.* 55 (1) (1984) 1–32.

[17] Matania Ben-Artzi, Jiequan Li, Hyperbolic balance laws: Riemann invariants and the generalized Riemann problem, *Numer. Math.* 106 (3) (2007) 369–425.

[18] Vladimir A. Titarev, Eleuterio F. Toro, ADER: arbitrary high order Godunov approach, *J. Sci. Comput.* 17 (1–4) (2002) 609–618.

[19] Vladimir A. Titarev, Eleuterio F. Toro, ADER schemes for three-dimensional non-linear hyperbolic systems, *J. Comput. Phys.* 204 (2) (2005) 715–736.

[20] Michael Dumbser, Dinshaw S. Balsara, Eleuterio F. Toro, Claus-Dieter Munz, A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes, *J. Comput. Phys.* 227 (18) (2008) 8209–8253.

[21] Dinshaw S. Balsara, Tobias Rumpf, Michael Dumbser, Claus-Dieter Munz, Efficient, high accuracy ADER-WENO schemes for hydrodynamics and divergence-free magnetohydrodynamics, *J. Comput. Phys.* 228 (7) (2009) 2480–2516.

[22] Dinshaw S. Balsara, Higher-order accurate space-time schemes for computational astrophysics—part I: finite volume methods, *Living Rev. Comput. Astrophys.* 3 (1) (2017) 2.

[23] Dinshaw S. Balsara, Chad Meyer, Michael Dumbser, Huijing Du, Zhiliang Xu, Efficient implementation of ADER schemes for Euler and magnetohydrodynamical flows on structured meshes—speed comparisons with Runge-Kutta methods, *J. Comput. Phys.* 235 (2013) 934–969.

[24] Michael Dumbser, Olindo Zanotti, Arturo Hidalgo, Dinshaw S. Balsara, ADER-WENO finite volume schemes with space-time adaptive mesh refinement, *J. Comput. Phys.* 248 (2013) 257–286.

[25] Francesco Fambri, Michael Dumbser, Olindo Zanotti, Space-time adaptive ADER-DG schemes for dissipative flows: compressible Navier-Stokes and resistive MHD equations, *Comput. Phys. Commun.* 220 (2017) 297–318.

[26] Stéphane Clain, Steven Diot, Raphaël Loubère, A high-order finite volume method for systems of conservation laws—multi-dimensional optimal order detection (MOOD), *J. Comput. Phys.* 230 (10) (2011) 4028–4050.

[27] Steven Diot, Stéphane Clain, Raphaël Loubère, Improved detection criteria for the multi-dimensional optimal order detection (MOOD) on unstructured meshes with very high-order polynomials, *Comput. Fluids* 64 (2012) 43–63.

[28] Steven Diot, Raphaël Loubère, Stéphane Clain, The multidimensional optimal order detection method in the three-dimensional case: very high-order finite volume method for hyperbolic systems, *Int. J. Numer. Methods Fluids* 73 (4) (2013) 362–392.

[29] Walter Boscheri, Raphaël Loubère, Michael Dumbser, Direct arbitrary-Lagrangian–Eulerian ADER-MOOD finite volume schemes for multidimensional hyperbolic conservation laws, *J. Comput. Phys.* 292 (2015) 56–87.

[30] Raphaël Loubère, Michael Dumbser, Steven Diot, A new family of high order unstructured MOOD and ADER finite volume schemes for multidimensional systems of hyperbolic conservation laws, *Commun. Comput. Phys.* 16 (3) (2014) 718–763.

[31] Olindo Zanotti, Michael Dumbser, Efficient conservative ADER schemes based on WENO reconstruction and space-time predictor in primitive variables, *Comput. Astrophys. Cosmol.* 3 (1) (2016) 1.

[32] Matthew R. Norman, Hal Finkel, Multi-moment ADER-Taylor methods for systems of conservation laws with source terms in one dimension, *J. Comput. Phys.* 231 (20) (2012) 6622–6642.

[33] Matthew R. Norman, Algorithmic improvements for schemes using the ADER time discretization, *J. Comput. Phys.* 243 (2013) 176–178.

[34] Matthew R. Norman, A WENO-limited, ADER-DT, finite-volume scheme for efficient, robust, and communication-avoiding multi-dimensional transport, *J. Comput. Phys.* 274 (2014) 1–18.

[35] L. Del Zanna, O. Zanotti, N. Bucciantini, P. Londrillo, ECHO: a Eulerian conservative high-order scheme for general relativistic magnetohydrodynamics and magnetodynamics, *Astron. Astrophys.* 473 (1) (2007) 11–30.

[36] Yuxi Chen, Gábor Tóth, Tamas I. Gombosi, A fifth-order finite difference scheme for hyperbolic equations on block-adaptive curvilinear grids, *J. Comput. Phys.* 305 (2016) 604–621.

[37] Adam Reyes, Dongwook Lee, Carlo Graziani, Petros Tzeferacos, A variable high-order shock-capturing finite difference method with GP-WENO, *J. Comput. Phys.* 381 (2019) 189–217.

[38] Ernst Hairer, Gerhard Wanner, Multistep-multistage-multiderivative methods for ordinary differential equations, *Computing* 11 (3) (1973) 287–303.

[39] David C. Seal, Yaman Güclü, Andrew J. Christlieb, High-order multiderivative time integrators for hyperbolic conservation laws, *J. Sci. Comput.* 60 (1) (2014) 101–140.

[40] Andrew J. Christlieb, Yaman Güclü, David C. Seal, The Picard integral formulation of weighted essentially nonoscillatory schemes, *SIAM J. Numer. Anal.* 53 (4) (2015) 1833–1856.

[41] David C. Seal, Qi Tang, Zhengfu Xu, Andrew J. Christlieb, An explicit high-order single-stage single-step positivity-preserving finite difference WENO method for the compressible Euler equations, *J. Sci. Comput.* 68 (1) (2016) 171–190.

[42] Jianxian Qiu, Chi-Wang Shu, Finite difference WENO schemes with Lax-Wendroff-type time discretizations, *SIAM J. Sci. Comput.* 24 (6) (2003) 2185–2198.

[43] Yan Jiang, Chi-Wang Shu, Mengping Zhang, An alternative formulation of finite difference weighted ENO schemes with Lax-Wendroff time discretization for conservation laws, *SIAM J. Sci. Comput.* 35 (2) (2013) A1137–A1160.

[44] Sergei Konstantinovich Godunov, A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics, *Mat. Sb.* 89 (3) (1959) 271–306.

[45] Dongwook Lee, Hugues Faller, Adam Reyes, The piecewise cubic method (PCM) for computational fluid dynamics, *J. Comput. Phys.* 341 (2017) 230–257.

[46] Peter N. Brown, Youcef Saad, Hybrid Krylov methods for nonlinear systems of equations, *SIAM J. Sci. Stat. Comput.* 11 (3) (1990) 450–481.

[47] Dana A. Knoll, David E. Keyes, Jacobian-free Newton-Krylov methods: a survey of approaches and applications, *J. Comput. Phys.* 193 (2) (2004) 357–397.

[48] Dana A. Knoll, H. Park, Kord Smith, Application of the Jacobian-free Newton-Krylov method to nonlinear acceleration of transport source iteration in slab geometry, *Nucl. Sci. Eng.* 167 (2) (2011) 122–132.

[49] Charles William Gear, Youcef Saad, Iterative solution of linear equations in ODE codes, *SIAM J. Sci. Stat. Comput.* 4 (4) (1983) 583–601.

[50] Heng-Bin An, Ju Wen, Tao Feng, On finite difference approximation of a matrix-vector product in the Jacobian-free Newton-Krylov method, *J. Comput. Appl. Math.* 236 (6) (2011) 1399–1409.

[51] James Kent, Jared P. Whitehead, Christiane Jablonowski, Richard B. Rood, Determining the effective resolution of advection schemes. Part I: dispersion analysis, *J. Comput. Phys.* 278 (2014) 485–496.

[52] James Kent, Christiane Jablonowski, Jared P. Whitehead, Richard B. Rood, Determining the effective resolution of advection schemes. Part II: numerical testing, *J. Comput. Phys.* 278 (2014) 497–508.

- [53] Gary A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, *J. Comput. Phys.* 27 (1) (1978) 1–31.
- [54] Paul Woodward, Phillip Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, *J. Comput. Phys.* 54 (1) (1984) 115–173.
- [55] Chi-Wang Shu, Stanley Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, in: *Upwind and High-Resolution Schemes*, Springer, 1989, pp. 328–374.
- [56] Chi-Wang Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, Springer, 1998, pp. 325–432.
- [57] Seth C. Spiegel, H.T. Huynh, James R. DeBonis, A survey of the isentropic Euler vortex problem using high-order methods, in: *22nd AIAA Computational Fluid Dynamics Conference*, 2015, p. 2444.
- [58] Tong Zhang, Yu Xi Zheng, Conjecture on the structure of solutions of the Riemann problem for two-dimensional gas dynamics systems, *SIAM J. Math. Anal.* 21 (3) (1990) 593–630.
- [59] Carsten W. Schulz-Rinne, Classification of the Riemann problem for two-dimensional gas dynamics, *SIAM J. Math. Anal.* 24 (1) (1993) 76–88.
- [60] Carsten W. Schulz-Rinne, James P. Collins, Harland M. Glaz, Numerical solution of the Riemann problem for two-dimensional gas dynamics, *SIAM J. Sci. Comput.* 14 (6) (1993) 1394–1414.
- [61] Dinshaw S. Balsara, Multidimensional HLLE Riemann solver: application to Euler and magnetohydrodynamic flows, *J. Comput. Phys.* 229 (6) (2010) 1970–1993.
- [62] Paweł Buchmüller, Christiane Helzel, Improved accuracy of high-order WENO finite volume methods on Cartesian grids, *J. Sci. Comput.* 61 (2) (2014) 343–368.
- [63] Wai-Sun Don, Zhen Gao, Peng Li, Xiao Wen, Hybrid compact-WENO finite difference scheme with conjugate Fourier shock detection algorithm for hyperbolic conservation laws, *SIAM J. Sci. Comput.* 38 (2) (2016) A691–A711.
- [64] W.H. Hui, P.Y. Li, Z.W. Li, A unified coordinate system for solving the two-dimensional Euler equations, *J. Comput. Phys.* 153 (2) (1999) 596–637.
- [65] Richard Liska, Burton Wendroff, Comparison of several difference schemes on 1D and 2D test problems for the Euler equations, *SIAM J. Sci. Comput.* 25 (3) (2003) 995–1017.
- [66] James M. Stone, Thomas A. Gardiner, Peter Teuben, John F. Hawley, Jacob B. Simon, Athena: a new code for astrophysical MHD, *Astrophys. J. Suppl. Ser.* 178 (1) (2008) 137.
- [67] Friedemann Kemm, On the proper setup of the double Mach reflection as a test case for the resolution of gas dynamics codes, *Comput. Fluids* 132 (2016) 72–75.
- [68] Francisco Alcrudo, Pilar Garcia-Navarro, A high-resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations, *Int. J. Numer. Methods Fluids* 16 (6) (1993) 489–505.
- [69] Eleuterio F. Toro, *Shock-Capturing Methods for Free-Surface Shallow Flows*, John Wiley, 2001.
- [70] Anargyros I. Delis, Th. Katsaounis, Numerical solution of the two-dimensional shallow water equations by the application of relaxation methods, *Appl. Math. Model.* 29 (8) (2005) 754–783.