



# Efficient Sparse-Grid Implementation of a Fifth-Order Multi-resolution WENO Scheme for Hyperbolic Equations

Ernie Tsybulnik<sup>1</sup> · Xiaozhi Zhu<sup>1</sup> · Yong-Tao Zhang<sup>1</sup>

Received: 31 October 2021 / Revised: 3 June 2022 / Accepted: 1 July 2022  
© Shanghai University 2022

## Abstract

High-order accurate weighted essentially non-oscillatory (WENO) schemes are a class of broadly applied numerical methods for solving hyperbolic partial differential equations (PDEs). Due to highly nonlinear property of the WENO algorithm, large amount of computational costs are required for solving multidimensional problems. In our previous work (Lu et al. in Pure Appl Math Q 14: 57–86, 2018; Zhu and Zhang in J Sci Comput 87: 44, 2021), sparse-grid techniques were applied to the classical finite difference WENO schemes in solving multidimensional hyperbolic equations, and it was shown that significant CPU times were saved, while both accuracy and stability of the classical WENO schemes were maintained for computations on sparse grids. In this technical note, we apply the approach to recently developed finite difference multi-resolution WENO scheme specifically the fifth-order scheme, which has very interesting properties such as its simplicity in linear weights' construction over a classical WENO scheme. Numerical experiments on solving high dimensional hyperbolic equations including Vlasov based kinetic problems are performed to demonstrate that the sparse-grid computations achieve large savings of CPU times, and at the same time preserve comparable accuracy and resolution with those on corresponding regular single grids.

**Keywords** Weighted essentially non-oscillatory (WENO) schemes · Multi-resolution WENO schemes · Sparse grids · High spatial dimensions · Hyperbolic partial differential equations (PDEs)

**Mathematics Subject Classification** 65M06 · 35L65

---

Research was partially supported by NSF Grant DMS-1620108.

---

✉ Yong-Tao Zhang  
yzhang10@nd.edu

Ernie Tsybulnik  
etsybuln@nd.edu

Xiaozhi Zhu  
xzhu4@nd.edu

<sup>1</sup> Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556, USA

## 1 Introduction

High-order accurate weighted essentially non-oscillatory (WENO) schemes are a class of broadly applied numerical methods for solving hyperbolic partial differential equations (PDEs) arising in science and engineering problems. This class of schemes is especially effective for problems whose solutions have both singularities (e.g., shock waves) and complicated smooth structures, e.g., see [29, 32]. High-order WENO schemes were first designed in [13, 19], then have been developed and studied extensively on both structured and unstructured meshes (e.g., [5, 7, 12, 17, 21, 27, 30, 31, 33, 35]). Due to highly non-linear property of the high-order accurate WENO algorithm, large amount of operations and computational costs are required in the simulations, especially for multidimensional problems with three or even higher spatial dimension. It is an important topic to develop efficient approach in implementing high-order WENO methods for solving high spatial dimensional problems.

Sparse-grid techniques are a class of efficient approximation tools used for solving high-dimensional problems in many scientific and engineering applications [2]. The main idea of sparse-grid methods is to reduce the number of degrees of freedom in regular single-grid computations, which was introduced in finite element simulations by Zenger [28]. For a  $d$  dimensional problem with  $N$  grid points in one coordinate direction, sparse-grid methods only involve  $O(N \cdot (\log N)^{d-1})$  degrees of freedom, rather than  $O(N^d)$  degrees of freedom in regular single-grid methods. In 1992, Griebel et al. [9] developed the sparse-grid combination technique, which provided an efficient way to implement sparse-grid methods. The key point of the sparse-grid combination technique is to design a linear combination of solutions on semi-coarsened sparse grids such that the resulted final solution can keep comparable accuracy order as that on a single full grid. This is achieved by choosing the combination coefficients to cancel certain leading-order error terms in sparse-grid solutions [9, 15, 16]. In our previous work [22, 23, 37], the sparse-grid combination technique was applied to the classical high-order WENO schemes [13, 25] in solving hyperbolic equations and convection-diffusion equations on high spatial dimensional domains, and it was shown that significant computational times were saved, while both the accuracy and stability of the classical high-order WENO schemes were maintained for simulations on sparse grids.

Recently in [34] a new type of WENO schemes, called multi-resolution WENO schemes, was developed to solve hyperbolic PDEs. Different from the classical WENO schemes [13, 25], this type of WENO schemes constructed WENO approximations on unequal-sized substencils. The multi-resolution WENO schemes exhibit many interesting properties such as their simplicity in constructing linear weights, which in general can be taken as arbitrary positive numbers with the only requirement that their sum equals 1. The multi-resolution WENO schemes simplify the classical WENO schemes on unstructured meshes [35], and improve the convergence of classical WENO schemes to steady state solutions [18, 36].

In this technical note, we extend our previous sparse-grid approach to the newly developed multi-resolution WENO schemes specifically the fifth-order finite difference multi-resolution WENO scheme in [34]. The goal is to obtain much more efficient computations of the fifth-order multi-resolution WENO (MRWENO5) scheme on sparse grids than that in their regular performance on a single full grid, for solving multidimensional hyperbolic PDEs. At the same time, comparable high-order accuracy of the multi-resolution WENO scheme in smooth regions and nonlinear stability in non-smooth regions of the solutions in sparse-grid simulations need to be preserved as that for simulations on regular single

grids. We would like to point out that this problem is not trivial, because the theoretical error analysis of nonlinear schemes such as these highly nonlinear WENO schemes on sparse grids is difficult and still an open problem, and the numerical experiment is the principal way to study these schemes. Here we use the sparse-grid combination technique to implement the MRWENO5 scheme on sparse grids. To improve the robustness of the algorithm, the MRWENO5 interpolation is applied for the prolongation step of sparse-grid combination. The rest of the paper is presented as following. In Sect. 2, we describe the numerical method and the procedure to adopt the sparse-grid combination technique to the MRWENO5 scheme, with a novel MRWENO5 prolongation. In Sect. 3, numerical experiments including solving three-dimensional (3D) and four-dimensional (4D) Vlasov based PDEs are performed to test the sparse-grid MRWENO5 method and verify significant CPU time savings by comparisons with single-grid simulations. Conclusions and discussions are provided in Sect. 4.

## 2 A Fifth-Order Sparse-Grid Multi-resolution WENO Finite Difference Method

We study efficient numerical methods for solving multidimensional hyperbolic PDEs,

$$u_t + \nabla \cdot f(u) = 0, \tag{1}$$

where  $u(x, t)$  is the unknown function, and  $f = (f_1, \dots, f_d)^T$  is the vector of flux functions defined on a spatial domain with the dimension  $d$ . We apply the method of lines (MOL) to solve Eq. (1). For spatial discretization, the recently developed fifth-order finite difference multi-resolution WENO scheme [34] is used. The goal of this technical note is to develop an effective way using sparse grids to achieve very efficient multi-resolution WENO simulations. We proceed by first describing the fifth-order finite difference multi-resolution WENO spatial discretization, then detailing the approach about its sparse-grid implementation by the sparse-grid combination technique with an MRWENO5 prolongation. Finally, the algorithm is summarized and given in its complete form.

### 2.1 The Fifth-Order Multi-resolution WENO Scheme (MRWENO5)

In discretizing Eq. (1), a conservative finite difference scheme is used. This scheme approximates the point values of unknown functions in the PDEs at a uniform (or smoothly varying) grid. Here the description for the discretization of derivatives will be given in one spatial direction, since the finite difference multi-resolution WENO schemes approximate multidimensional derivatives in a dimension by dimension way. Without loss of generality the  $x$ -direction derivative  $f(u)_x$  is considered, and discretizations in other directions follow a similar fashion. A conservative difference scheme is used to approximate the value of  $f(u)_x$  at a grid point with the  $x$ -coordinate  $x_i$ ,

$$f(u)_x|_{x=x_i} \approx \frac{1}{\Delta x} (\hat{f}_{i+1/2} - \hat{f}_{i-1/2}), \tag{2}$$

where  $\hat{f}_{i+1/2}$  is the numerical flux at the point  $x_{i+1/2} = (x_i + x_{i+1})/2$  and the uniform grid has the grid size  $\Delta x = (x_{i+1} - x_i)$  in the  $x$ -direction. To ensure correct upwind biasing and linear stability, the flux  $f(u)$  is split into a positive wind part  $f^+(u)$  (i.e.,  $\frac{df^+(u)}{du} \geq 0$  for the scalar case, or the corresponding eigenvalue is positive for the system case with a local

characteristic decomposition) and a negative wind part  $f^-(u)$  (i.e.,  $\frac{df^-(u)}{du} < 0$  or the corresponding eigenvalue is negative). As in [25], the popular ‘‘Lax-Friedrichs flux splitting’’ is employed as such

$$f^+(u) = \frac{1}{2}(f(u) + \alpha u), \quad f^-(u) = \frac{1}{2}(f(u) - \alpha u), \tag{3}$$

where  $\alpha = \max_u |f'(u)|$  over the range of  $u$ . Then using different stencils, the positive and negative numerical fluxes  $\hat{f}_{i+1/2}^+$  and  $\hat{f}_{i+1/2}^-$  are approximated by the MRWENO5 scheme separately. The final numerical flux in (2) is formed as  $\hat{f}_{i+1/2} = \hat{f}_{i+1/2}^+ + \hat{f}_{i+1/2}^-$ . The MRWENO5 approximations to numerical fluxes in [34] are summarized as the following.

The MRWENO5 approximation of the numerical flux  $\hat{f}_{i+1/2}^+$  is based on the stencils comprised of cells  $I_j = [x_{j-1/2}, x_{j+1/2}]$  with  $j = i - 2, \dots, i + 2$ , and numerical values  $f^+(u_j)$  at these grid points  $x_j$ . Note that for the simplicity of notations, here  $u_j$  denotes the numerical value of  $u$  at the grid point  $x = x_j$  along the grid lines of other spatial directions, with the understanding that the value at  $x_j$  may be different for different grid lines of other spatial directions. The procedure consists of the following five steps [34].

**Step 1** The MRWENO5 approximation has three substencils  $T_k = \{I_{i+1-k}, \dots, I_{i-1+k}\}$ ,  $k = 1, 2, 3$ . Based on them, we reconstruct  $2k - 2$  degree polynomials  $q_k(x)$  which satisfy

$$\frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} q_k(x) dx = f^+(u_i), \quad l = i - k + 1, \dots, i - 1 + k; \quad k = 1, 2, 3.$$

**Step 2** Obtain equivalent expressions for these reconstruction polynomials of different degrees, which are denoted by  $p_3(x)$ ,  $p_2(x)$ , and  $p_1(x)$ , and defined as follows:

$$p_3(x) = \frac{1}{\gamma_{3,3}} q_3(x) - \frac{\gamma_{1,3}}{\gamma_{3,3}} p_1(x) - \frac{\gamma_{2,3}}{\gamma_{3,3}} p_2(x), \tag{4}$$

$$p_2(x) = \frac{1}{\gamma_{2,2}} q_2(x) - \frac{\gamma_{1,2}}{\gamma_{2,2}} p_1(x), \tag{5}$$

$$p_1(x) = q_1(x). \tag{6}$$

Similar ideas to construct these expressions can also be found in the central WENO schemes [3, 17]. In principle, the linear weights  $\gamma$ 's can be any positive numbers on the only condition that  $\gamma_{1,2} + \gamma_{2,2} = 1$  with  $\gamma_{2,2} \neq 0$ , and  $\gamma_{1,3} + \gamma_{2,3} + \gamma_{3,3} = 1$  with  $\gamma_{3,3} \neq 0$ . In [34], a balance between the sharp and essentially non-oscillatory shock transitions in nonsmooth regions and accuracy in smooth regions is considered, and the values of these linear weights are taken as  $\gamma_{1,2} = 1/11$ ,  $\gamma_{2,2} = 10/11$ ,  $\gamma_{1,3} = 1/111$ ,  $\gamma_{2,3} = 10/111$ , and  $\gamma_{3,3} = 100/111$

**Step 3** Calculate the smoothness indicators  $\beta_{l_2}$ , which measure how smooth the functions  $p_{l_2}(x)$  for  $l_2 = 2, 3$  are in the interval  $[x_{i-1/2}, x_{i+1/2}]$ . The same approach as in [13] is used:

$$\beta_{l_2} = \sum_{\alpha=1}^{2(l_2-1)} \int_{x_{i-\Delta x/2}}^{x_{i+\Delta x/2}} \Delta x^{2\alpha-1} \left( \frac{d^\alpha p_{l_2}(x)}{dx^\alpha} \right)^2 dx, \quad l_2 = 2, 3. \tag{7}$$

The only exception is  $\beta_1$ , which is magnified from zero to a tiny value. See [34] for details.

**Step 4** Calculate the nonlinear weights based on the linear weights and the smoothness indicators. Here the WENO-Z type nonlinear weights [5] are applied, i.e., first we compute a quantity  $\tau$  which is based on the absolute differences of the smoothness indicators:

$$\tau = \left( \frac{\sum_{l=1}^2 |\beta_3 - \beta_l|}{2} \right)^2, \text{ then the nonlinear weights are}$$

$$\omega_{l_1} = \frac{\bar{\omega}_{l_1}}{\sum_{l=1}^3 \bar{\omega}_l}, \quad \bar{\omega}_{l_1} = \gamma_{l_1,3} \left( 1 + \frac{\tau}{\epsilon + \beta_{l_1}} \right), \quad l_1 = 1, 2, 3, \tag{8}$$

where  $\epsilon > 0$  which is a small number preventing the denominator from becoming 0.

**Step 5** The final MRWENO5 reconstruction of  $\hat{f}_{i+1/2}^+$  is obtained by

$$\hat{f}_{i+1/2}^+ = \sum_{l=1}^3 \omega_l p_l(x_{i+1/2}). \tag{9}$$

The procedure for the reconstruction of the numerical flux  $\hat{f}_{i+1/2}^-$  follows similarly in that it is mirror-symmetric with respect to  $x_{i+1/2}$ .

### 2.2 The Sparse-Grid MRWENO5 Scheme

In this section, we study how to implement the MRWENO5 scheme on sparse grids by incorporating a sparse-grid combination technique. The goal is to save the computational costs, and at the same time keep the accuracy and stability of the MRWENO5 scheme. The essential idea of the sparse-grid combination technique is to solve the PDE on several semi-coarsened grids (sparse grids) and combine the solutions from these semi-coarsened grids to get a solution on the most refined grid, which is corresponding to the usual single full grid in regular single-grid computations. The advantage of this technique is that the PDE is solved on a fewer number of grid points than the single full grid; therefore, computational costs are saved. The cost saving becomes more significant the higher the dimension of the problem. A good sparse-grid combination technique is expected to provide a final solution with comparable accuracy when compared to the solution obtained on a single full grid. Some previous work can be found in, e.g., [9, 15, 16, 22, 23, 37].

Without the loss of generality, the algorithm for two-dimensional (2D) cases is presented here, as procedures for higher dimensional cases are performed in a similar fashion. We consider a 2D domain  $[a, b]^2$  and define notations as follows. Begin by partitioning the domain into the coarsest mesh called a root grid  $\Omega^{0,0}$ . The root grid  $\Omega^{0,0}$  has the mesh size  $H = \frac{b-a}{N_r}$ , where  $N_r$  is the number of cells in each spatial direction. Next several refinements are performed in the  $x$  and  $y$  directions on the root grid to obtain a family of semi-coarsened sparse grids  $\{\Omega^{l_1, l_2}\}$ .  $l_1$  and  $l_2$  denote the levels of refinement in relation to the root grid in the  $x$  and  $y$  directions, respectively, where  $l_1 = 0, 1, \dots, N_L$  and  $l_2 = 0, 1, \dots, N_L$ ,  $N_L$  denoting the finest level of refinement. Thus, on each grid of semi-coarsened grids  $\{\Omega^{l_1, l_2}\}$ , in the  $x$  direction the mesh size is  $h_{l_1} = 2^{-l_1}H$  and similarly for the  $y$  direction  $h_{l_2} = 2^{-l_2}H$ . The finest grid is  $\Omega^{N_L, N_L}$  and has the grid size  $h = 2^{-N_L}H$  in both  $x$  and  $y$  directions. Note that a square domain is used here for the simplicity of the presentation; however, the algorithm here can be applied to any rectangular domain straightforwardly.

The MRWENO5 scheme described in the last section is used for spatial discretization of Eq. (1), and the popular third-order total variation diminishing Runge-Kutta (TVD-RK3) scheme [8, 25] is adopted for time discretization. Instead of solving the PDE (1) on a single full grid, we apply the sparse-grid combination technique to solve it on  $(2N_L + 1)$  sparse grids

$\{\Omega^{l_1, l_2}\}_I$ , where the index set  $I = \{(l_1, l_2) | l_1 + l_2 = N_L \text{ or } l_1 + l_2 = N_L - 1\}$ . More specifically, time marching of numerical solutions of the PDE (1) is carried out on these sparse grids:

$$\{\Omega^{0, N_L}, \Omega^{1, N_L-1}, \dots, \Omega^{N_L-1, 1}, \Omega^{N_L, 0}\} \quad \text{and} \quad \{\Omega^{0, N_L-1}, \Omega^{1, N_L-2}, \dots, \Omega^{N_L-2, 1}, \Omega^{N_L-1, 0}\},$$

and we obtain  $(2N_L + 1)$  sets of numerical solutions  $\{U^{l_1, l_2}\}_I$ . Then the next step is to combine solutions on these sparse grids to obtain the final solution on the finest grid  $\Omega^{N_L, N_L}$ , since the key point in sparse-grid computations to reduce computational cost is that the PDE is never solved directly on the full grid  $\Omega^{N_L, N_L}$ , but only on  $\{\Omega^{l_1, l_2}\}_I$  which in total contains fewer grid points. In this step, a prolongation operator  $P^{N_L, N_L}$  is first applied on each sparse-grid solution of  $\{U^{l_1, l_2}\}_I$  to obtain  $(2N_L + 1)$  solutions on the finest grid  $\Omega^{N_L, N_L}$ , then a combination of all of these solutions is performed to form the final solution  $\hat{U}^{N_L, N_L}$  on  $\Omega^{N_L, N_L}$ . The details of the prolongation operator  $P^{N_L, N_L}$  and the combination are described in the following.

### 2.2.1 MRWENO5 Prolongation $P^{N_L, N_L}$

We define a prolongation operator  $P^{N_L, N_L}$  such that  $P^{N_L, N_L}$  maps numerical solutions  $\{U^{l_1, l_2}\}_I$  on sparse grids onto the finest grid  $\Omega^{N_L, N_L}$ . For example, given a numerical solution  $U^{l_1, l_2}$  on  $\Omega^{l_1, l_2}$ , then  $P^{N_L, N_L} U^{l_1, l_2}$  provides numerical values on all of the grid points of  $\Omega^{N_L, N_L}$ . Implementation of prolongation operators is usually accomplished by interpolation procedure. Regular Lagrange interpolations can be used directly for a smooth solution, but in the case of solutions with discontinuities such as those we encounter with solving hyperbolic PDEs, more robust interpolations are required. Based on results in [9, 15, 16] for linear schemes and in [22, 23, 37] for nonlinear schemes, it is concluded that the final sparse-grid solution  $\hat{U}^{N_L, N_L}$  has similar accuracy orders as the corresponding single-grid one, as long as the interpolation accuracy order in prolongation is not less than the accuracy order of the numerical scheme itself used to solve PDEs. Therefore, here, we use the MRWENO5 interpolation in prolongation. The interpolations are implemented dimension by dimension, and the MRWENO5 interpolation procedure for the one-dimensional (1D) case is presented as the following.

Given values  $u_{i-2}, u_{i-1}, u_i, u_{i+1}, u_{i+2}$  at the grid points  $x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}$ , respectively, we find an MRWENO5 interpolation  $u_{\text{MRI}}(x)$  for any point  $x \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ . Denote the uniform grid size by  $h$ , and  $x_{i-\frac{1}{2}} = (x_{i-1} + x_i)/2$ ,  $x_{i+\frac{1}{2}} = (x_i + x_{i+1})/2$ . The MRWENO5 interpolation procedure is similar as the MRWENO5 approximation described in Sect. 2.1, except that here the Lagrange interpolation is used rather than the reconstruction to form the approximation polynomials. More specifically, first we form the approximation polynomials

$$p_1(x) = q_1(x), \tag{10}$$

$$p_2(x) = \frac{1}{\gamma_{2,2}} q_2(x) - \frac{\gamma_{1,2}}{\gamma_{2,2}} p_1(x), \tag{11}$$

$$p_3(x) = \frac{1}{\gamma_{3,3}} q_3(x) - \frac{\gamma_{1,3}}{\gamma_{3,3}} p_1(x) - \frac{\gamma_{2,3}}{\gamma_{3,3}} p_2(x), \tag{12}$$

where  $q_1(x) = u_i$ , the degree 0 polynomial which interpolates  $u$  on the point  $x_i$ ;  $q_2(x)$  is the degree two polynomial which interpolates  $u$  on the points  $x_{i-1}, x_i, x_{i+1}$ ;  $q_3(x)$  is the

degree four polynomial which interpolates  $u$  on the points  $x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}$ . These linear weights  $\gamma$ 's are same as those defined in Sect. 2.1, where  $\gamma_{1,2} = 1/11, \gamma_{2,2} = 10/11, \gamma_{1,3} = 1/111, \gamma_{2,3} = 10/111, \text{ and } \gamma_{3,3} = 100/111$ .

Then, Eq. (7) is used to calculate smoothness indicators  $\beta_2$  and  $\beta_3$  for these approximation polynomials  $p_2(x)$  and  $p_3(x)$ :

$$\beta_2 = \frac{121}{300} (4u_{i-1}^2 + (5u_{i+1} - 13u_i)u_{i-1} + 13u_i^2 + 4u_{i+1}^2 - 13u_i u_{i+1}), \tag{13}$$

$$\begin{aligned} \beta_3 = \frac{1}{67\,200\,000} & \left( 112\,756\,316u_{i-2}^2 - 37(22\,231\,031u_{i-1} - 29\,557\,877u_i + 17\,054\,405u_{i+1} \right. \\ & - 3\,632\,623u_{i+2})u_{i-2} + 1\,657\,473\,113u_{i-1}^2 + 3\,613\,771\,547u_i^2 + 1\,657\,473\,113u_{i+1}^2 \\ & + 112\,756\,316u_{i+2}^2 - 4\,707\,412\,996u_i u_{i+1} + u_{i-1}(-4\,707\,412\,996u_i + 2\,846\,027\,902u_{i+1} \\ & \left. - 631\,012\,985u_{i+2}) + 1\,093\,641\,449u_i u_{i+2} - 822\,548\,147u_{i+1}u_{i+2} \right). \end{aligned} \tag{14}$$

Similarly as in Sect. 2.1, the smoothness indicator  $\beta_1$  associated with the constant polynomial  $p_1(x)$  is magnified from zero to a tiny value as in [34]. The corresponding nonlinear weights  $\omega_1, \omega_2, \omega_3$  are computed using the formula (8). The final MRWENO5 interpolation is then given by

$$u_{\text{MRI}}(x) = \omega_1 p_1(x) + \omega_2 p_2(x) + \omega_3 p_3(x). \tag{15}$$

The dimension by dimension approach is implemented for multidimensional interpolations to obtain the MRWENO5 prolongation on the finest grid, for example the  $P^{N_L, N_L} U^{l_1, l_2}$  on  $\Omega^{N_L, N_L}$  in a 2D case.

### 2.2.2 Algorithm Summary

The algorithm of the sparse-grid MRWENO5 scheme is summarized here.

#### Algorithm: Sparse-Grid MRWENO5 Scheme

- Step 1 Restrict the initial condition of the PDE (1) to  $(2N_L + 1)$  sparse grids  $\{\Omega^{l_1, l_2}\}_I$  defined above. Namely, the initial condition functions are evaluated at grid points of sparse grids  $\{\Omega^{l_1, l_2}\}_I$ .
- Step 2 On each sparse grid  $\Omega^{l_1, l_2}$  in  $\{\Omega^{l_1, l_2}\}_I$ , solve the PDE (1) by the MRWENO5 scheme with the TVD-RK3 time stepping and march to the final time  $T$ . Then  $(2N_L + 1)$  sets of numerical solutions  $\{U^{l_1, l_2}\}_I$  are obtained.
- Step 3 At the final time  $T$ ,
  - on each sparse grid  $\Omega^{l_1, l_2}$  in  $\{\Omega^{l_1, l_2}\}_I$ , apply the MRWENO5 prolongation operator  $P^{N_L, N_L}$  on the numerical solution  $U^{l_1, l_2}$ , and obtain  $P^{N_L, N_L} U^{l_1, l_2}$  on the most refined grid  $\Omega^{N_L, N_L}$ ;
  - perform the combination to compute the final solution:

$$\hat{U}^{N_L, N_L} = \sum_{l_1+l_2=N_L} P^{N_L, N_L} U^{l_1, l_2} - \sum_{l_1+l_2=N_L-1} P^{N_L, N_L} U^{l_1, l_2}. \tag{16}$$

The algorithm is similar for higher dimensional problems with performing the prolongation operations to additional spatial directions. The sparse-grid combination formula for a general  $d$  dimensional problem is ([9])

$$\hat{U}^{N_L, \dots, N_L} = \sum_{m=N_L}^{N_L+d-1} (-1)^{d+N_L-(m+1)} \binom{d-1}{m-N_L} \sum_{|I_d|=m-(d-1)} P^{N_L, \dots, N_L} U^{I_1, \dots, I_d}. \quad (17)$$

Here  $N_L$  is the finest level of the sparse grids used in the simulation.  $I_d = (I_1, I_2, \dots, I_d)$  denotes the index of the levels of a sparse grid  $\Omega^{I_1, I_2, \dots, I_d}$ , and  $|I_d| = I_1 + I_2 + \dots + I_d$ .  $U^{I_1, \dots, I_d}$  is the numerical solution by solving the PDEs on the sparse grid  $\Omega^{I_1, \dots, I_d}$ , and  $P^{N_L, \dots, N_L}$  is the prolongation operator onto the finest grid  $\Omega^{N_L, \dots, N_L}$ .  $\hat{U}^{N_L, \dots, N_L}$  is the final solution of the sparse-grid combination on  $\Omega^{N_L, \dots, N_L}$ . In the numerical experiments of the next section, the 2D, 3D, and 4D formulas corresponding to  $d = 2, 3, 4$  in (17) are used.

**Remark 1** From the description of the sparse-grid MRWENO5 scheme of Sect. 2, we can see that the computational advantages of finite difference MRWENO methods on single grids [34] are preserved in this sparse-grid scheme. In the sparse-grid combination technique used here, PDEs are solved independently by the finite difference MRWENO5 scheme on selected semi-coarsened grids, and good properties of finite difference MRWENO methods (e.g., their simplicity in constructing linear weights) do not change on these grids. Furthermore, in the prolongation step, the MRWENO5 interpolation procedure is also simpler in the step of constructing linear weights, than the sparse-grid WENO5 method in [37] which uses the classical fifth-order WENO interpolation. For finite volume MRWENO methods on unstructured meshes [35], how their computational advantages are preserved is a very interesting topic. It depends on how to construct finite volume sparse-grid WENO schemes on unstructured meshes (e.g., unstructured triangular meshes), which is still an open problem as far as we know.

### 3 Numerical Experiments

In this section, numerical experiments on solving multidimensional problems including 3D and 4D Vlasov based PDEs are performed to test the sparse-grid MRWENO5 method and show a large amount of CPU time savings by comparisons with corresponding single-grid simulations. Error analysis on linear schemes for linear PDEs [9, 16] shows that the sparse-grid combination leads to a canceling in leading-order errors of numerical solutions on semi-coarsened grids, hence the accuracy order of the final solution of a sparse-grid simulation is maintained to be almost the same as that on a corresponding single full grid simulation. However, such sparse-grid error analysis is very difficult to carry out for the MRWENO5 method due to its high nonlinearity. Following our previous studies [22, 23, 37], instead of theoretical analysis, numerical simulations are used to show the fifth-order accuracy for the sparse-grid MRWENO5 scheme. Specifically, mesh refinement studies are performed to compute numerical convergence rates on successively refined grids. In [37], two different approaches, “refine root grid” and “refine levels”, are tested and compared for mesh refinement of sparse-grid simulations. For example, for 3D sparse grids with a  $10 \times 10 \times 10$  root grid and  $N_L = 3$ , the finest grid is  $80 \times 80 \times 80$ . The “refine root grid” approach is to refine the root grid, while the total number of semi-coarsened sparse-grid levels  $N_L + 1$  is kept unchanged. Therefore, if the

root grid is refined once to be  $20 \times 20 \times 20$ , we obtain the finest grid  $160 \times 160 \times 160$ . The “refine levels” approach refines the sparse-grid levels while keeping the root grid fixed. Therefore, if  $N_L = 3$  is refined once to be  $N_L = 4$  with the fixed  $10 \times 10 \times 10$  root grid, the finest grid  $160 \times 160 \times 160$  is also obtained. It is found in [37] that although the “refine levels” approach is more efficient and saves more CPU time costs than the “refine root grid” approach, it has obvious accuracy-order reductions for the fifth-order sparse-grid WENO scheme. The “refine root grid” approach can always achieve the desired fifth-order accuracy of the sparse-grid WENO scheme. Hence in this technical note, we use the “refine root grid” approach in numerical experiments. In addition, as discovered in [22, 37], to achieve the desired numerical accuracy, time step sizes used to march the PDEs on all semi-coarsened sparse grids have to be determined by the spatial grid size of the most refined grid, i.e.,  $\Omega^{N_L \cdot N_L}$  in the 2D cases, or  $\Omega^{N_L \cdot N_L \cdot N_L}$  in the 3D cases, etc. This way to choose time step sizes is also followed here. All of the numerical simulations in this technical note are performed on a 2.3 GHz, 16 GB RAM Linux workstation.

We first test the sparse-grid MRWENO5 method on nonlinear Burgers’ equations to study its numerical accuracy orders for smooth solution cases, and properties for the cases after shock waves form. Then the method is applied to 3D and 4D Vlasov based PDEs in kinetic simulations, to show its high efficiency and potential in solving high dimensional application problems. In this section, we use  $N_h$  to denote the number of computational cells in one spatial direction of the most refined grid in sparse grids or the corresponding single grid.

**Example 1 (A 2D Burgers’ equation).**

Consider the 2D Burgers’ equation

$$\begin{cases} u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0, \\ u(x, y, 0) = 0.3 + 0.7 \sin\left(\frac{\pi}{2}(x + y)\right) \end{cases} \quad (x, y) \in [-2, 2] \times [-2, 2], \quad (18)$$

with periodic boundary conditions. At first, the problem is solved till the final time  $T = 0.5/\pi^2$  when the solution is still smooth, to test numerical accuracy orders of schemes. In this example, we also compare the accuracy difference of the sparse-grid WENO5 method in [37] which uses the classical fifth-order WENO scheme [13], and the sparse-grid MRWENO5 method in this paper. So both the classical fifth-order WENO scheme and the multi-resolution fifth-order WENO scheme on sparse grids and corresponding single grids are used to solve the problem and compare the results. For the sparse-grid WENO5 scheme, the classical fifth-order WENO prolongation is used in sparse-grid combination, while the MRWENO5 prolongation is used for the sparse-grid MRWENO5 scheme. The TVD-RK3 scheme is used for the time evolution, so to test numerical accuracy orders of the fifth-order WENO schemes, we follow the common practice in the literature and take the time step size  $\Delta t_n$  as  $\Delta t_n = \min(\Delta t_1, \Delta t_2)$ , where  $\Delta t_1 = \frac{\text{CFL}}{\alpha_x/\Delta x + \alpha_y/\Delta y}$  and  $\Delta t_2 = (\min(\Delta x, \Delta y))^{5/3}$ . Note that CFL is the CFL number and taken as 0.5;  $\Delta x$  and  $\Delta y$  are grid sizes in the  $x, y$  directions, respectively, of the finest grid  $\Omega^{N_L \cdot N_L}$  and here  $\Delta x = \Delta y = h$ ;  $\alpha_x = \max_u |f'(u)|$  and  $\alpha_y = \max_u |g'(u)|$  with  $f(u) = g(u) = \frac{u^2}{2}$  in this case. For smooth solutions, we follow the suggestion in [6] and choose the values of  $\epsilon$  in both the MRWENO5 and the classical WENO5 schemes as a function of spatial grid sizes, instead of a fixed constant, for better accuracy errors and convergence orders. Specifically,  $\epsilon = h^2$  for single-grid computations; for sparse-grid computations, since spatial grid sizes are different on different directions of the spare grids, we take  $\epsilon = h_1^2$  in the WENO approximations of the  $x$  direction and similarly  $\epsilon = h_2^2$  in the WENO approximations of the  $y$  direction for WENO

**Table 1** Example 1, 2D Burgers' equation

Single-grid							
	$N_h \times N_h$	$L^\infty$ error	Order	$L^1$ error	Order	CPU/s	
	80 × 80	$2.20 \times 10^{-7}$		$5.41 \times 10^{-8}$		0.02	
	160 × 160	$7.22 \times 10^{-9}$	4.93	$1.77 \times 10^{-9}$	4.93	0.20	
	320 × 320	$2.30 \times 10^{-10}$	4.97	$5.66 \times 10^{-11}$	4.97	2.37	
	640 × 640	$7.25 \times 10^{-12}$	4.99	$1.78 \times 10^{-12}$	4.99	34.38	
Sparse-grid							
$N_r$	$N_L$	$N_h \times N_h$	$L^\infty$ error	Order	$L^1$ error	Order	CPU/s
10	3	80 × 80	$9.97 \times 10^{-6}$		$2.09 \times 10^{-6}$		0.05
20	3	160 × 160	$2.92 \times 10^{-8}$	8.41	$6.71 \times 10^{-9}$	8.28	0.24
40	3	320 × 320	$2.58 \times 10^{-10}$	6.83	$6.95 \times 10^{-11}$	6.59	1.81
80	3	640 × 640	$7.26 \times 10^{-12}$	5.15	$1.86 \times 10^{-12}$	5.23	20.48

Classical WENO5 scheme, comparison of numerical errors and CPU times for computations on single-grid and sparse-grid. Classical WENO5 interpolation for prolongation is employed in sparse-grid computations. Final time  $T = 0.5/\pi^2$ .  $N_r$ : number of cells in each spatial direction of a root grid.  $N_L$ : the finest level in a sparse-grid computation. CPU: CPU time for a complete simulation. CPU time unit: seconds

**Table 2** Example 1, 2D Burgers' equation

Single-grid							
	$N_h \times N_h$	$L^\infty$ error	Order	$L^1$ error	Order	CPU/s	
	80 × 80	$2.19 \times 10^{-7}$		$4.96 \times 10^{-8}$		0.04	
	160 × 160	$7.21 \times 10^{-9}$	4.93	$1.61 \times 10^{-9}$	4.95	0.62	
	320 × 320	$2.31 \times 10^{-10}$	4.97	$5.10 \times 10^{-11}$	4.98	6.85	
	640 × 640	$7.25 \times 10^{-12}$	4.99	$1.59 \times 10^{-12}$	5.00	73.46	
Sparse-grid							
$N_r$	$N_L$	$N_h \times N_h$	$L^\infty$ error	Order	$L^1$ error	Order	CPU/s
10	3	80 × 80	$1.82 \times 10^{-6}$		$2.85 \times 10^{-7}$		0.07
20	3	160 × 160	$8.30 \times 10^{-9}$	7.78	$2.18 \times 10^{-9}$	7.03	0.44
40	3	320 × 320	$2.34 \times 10^{-10}$	5.15	$5.32 \times 10^{-11}$	5.36	4.04
80	3	640 × 640	$7.26 \times 10^{-12}$	5.01	$1.60 \times 10^{-12}$	5.05	52.96

MRWENO5 scheme, comparison of numerical errors and CPU times for computations on single-grid and sparse-grid. MRWENO5 interpolation for prolongation is employed in sparse-grid computations. Final time  $T = 0.5/\pi^2$ .  $N_r$ : number of cells in each spatial direction of a root grid.  $N_L$ : the finest level in a sparse-grid computation. CPU: CPU time for a complete simulation. CPU time unit: seconds

schemes on each grid of semi-coarsened sparse grids  $\{\Omega^{l_1, l_2}\}_l$ . Similarly for both the MRWENO5 and the classical WENO5 prolongations,  $\epsilon$  is chosen in the same manner.

The  $L^\infty$  errors,  $L^1$  errors, and their numerical accuracy orders and CPU times are reported in Table 1 for the classical WENO5 scheme and Table 2 for the MRWENO5 scheme. We observe that the fifth-order accuracy is obtained for all schemes along with the mesh refinement. On the relatively coarse grids, the numerical errors of sparse-grid computations are greater than those on the corresponding single grids. With more

refined grids, the numerical errors become comparable to each other as the sparse-grid computation errors catch up to those on single-grid computations. A behavior of super-convergence is observed in sparse-grid computations. From the CPU time costs, on refined meshes we see around 30% – 40% computational time saved for simulations on sparse grids vs. single grids, for this 2D problem. In addition, due to the aforementioned time step size choice, for a relatively refined spatial grid, the total number of time steps to reach the final time  $T$  increases by a factor of about  $2^{5/3}$  if the spatial mesh  $N_h \times N_h$  is refined once. Therefore, with one spatial mesh refinement (the number of spatial grid points increases by a factor of about 4), the CPU time cost of the time evolution part in the simulation, which is the major computational cost part of the whole simulation, is expected to increase by a factor of about 12.7. In the mesh refinement study cases for sparse-grid computations here, the CPU time cost of the time evolution part also follows this way, i.e., it is expected to increase by a factor of about 12.7 with one spatial mesh  $N_h \times N_h$  refinement, when the spatial mesh is relatively refined. In Tables 1 and 2, we see a factor of about 11.3 and 13.1 increase of CPU times for the refined meshes of the sparse-grid computations, which is approximately the expected rate, with the differences coming from CPU time costs of the other procedures besides the time evolution part in the simulation and possible perturbations to CPU time measurements by other background programs of the computer system, etc. Comparing the results of the classical WENO5 and the MRWENO5 schemes, it is interesting to find that their numerical errors in single-grid computations are similar. However, on the relatively coarse grids of sparse-grid computations, the sparse-grid MRWENO5 scheme has much smaller numerical errors than the classical sparse-grid WENO5 scheme, although it needs more CPU time costs.

Next, we solve the problem till later time when shock waves appear in the solution. The numerical solutions of the MRWENO5 scheme by sparse-grid and the corresponding single-grid simulations, at the time  $T = 5/\pi^2$  when large gradients start to form and at  $T = 2$  when the discontinuities and shock profile become sharper are presented in Fig. 1. The sparse-grid root grid  $N_r = 80$ , and the finest level  $N_L = 3$ . Hence the most refined mesh in the sparse grids or the corresponding single grid has  $640 \times 640$  computational cells. The pictures of results by the classical WENO5 scheme are similar, hence are omitted here to save space. For discontinuous solutions, our numerical experiments suggest that smaller  $\epsilon$  values in sparse-grid WENO schemes give more stable results and better resolution, as that in regular high-order WENO schemes on single grids. Hence we take  $\epsilon = 10^{-10}$  as in [34]. From Fig. 1, we observe that the numerical solutions by the sparse-grid MRWENO5 scheme and its corresponding single-grid simulation are comparable. The nonlinear stability and high resolution properties of the MRWENO5 scheme for resolving shock waves are preserved well in the sparse-grid simulation. In terms of computational efficiency, it takes 547.95 s and 2 003.13 s of CPU time to complete the simulation at  $T = 5/\pi^2$  and  $T = 2$ , respectively, in the sparse-grid computation, while 706.46 s and 2 660.58 s of the CPU time are needed for finishing the simulation at  $T = 5/\pi^2$  and  $T = 2$ , respectively, in the corresponding single-grid computation. About 25% CPU time is saved by performing the MRWENO5 simulation on the sparse grids here. In the next 3D and 4D examples, we will observe much more significant CPU time saving by sparse-grid simulations.

#### **Example 2 (A 3D Burgers' equation).**

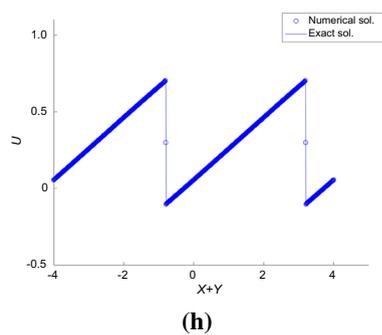
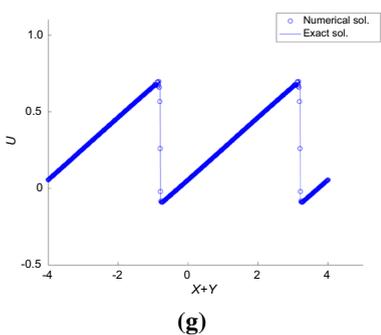
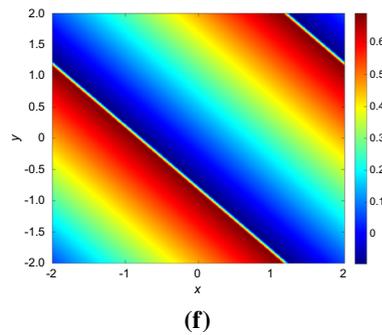
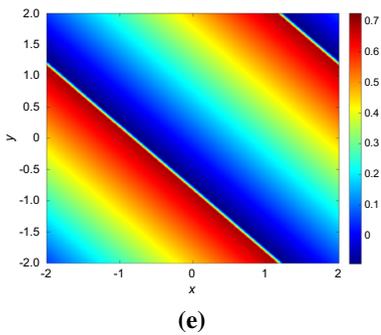
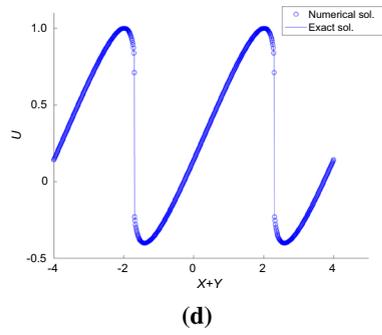
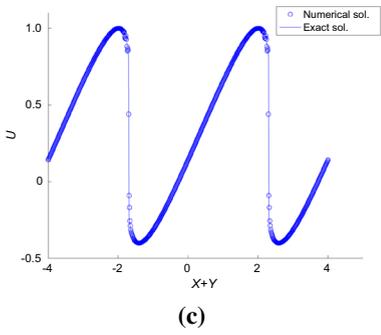
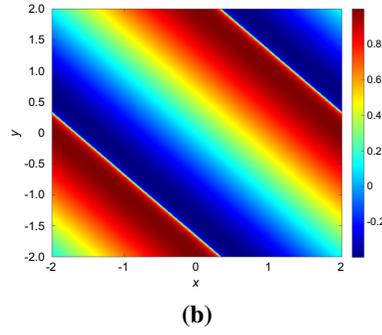
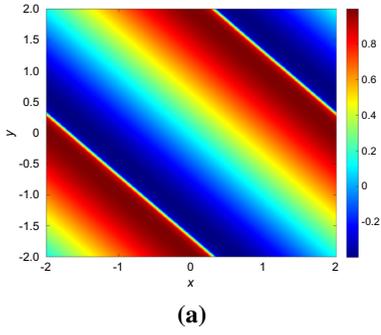
Consider the 3D Burgers' equation

**Fig. 1** Example 1, solution of 2D Burgers' equation by the MRWENO5 scheme on sparse grids ( $N_r = 80$  for root grid, finest level  $N_L = 3$  in the sparse-grid computation) and the corresponding  $640 \times 640$  single grid, using the MRWENO5 interpolation for prolongation in the sparse-grid combination. (a)–(d): solution at  $T = 5/\pi^2$ ; (e)–(h): solution at  $T = 2$ ; (a), (c), (e), (g): sparse-grid results; (b), (d), (f), (h): single-grid results; (a), (b), (e), (f): contour plots; (c), (d), (g), (h): 1D cutting-plots along  $x = y$ . Circles: numerical solutions; lines: exact solution

$$\begin{cases} u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y + \left(\frac{u^2}{2}\right)_z = 0, \\ u(x, y, z, 0) = 0.3 + 0.7 \sin\left(\frac{\pi}{3}(x + y + z)\right) \end{cases} \quad (x, y, z) \in [-3, 3] \times [-3, 3] \times [-3, 3], \tag{19}$$

with periodic boundary conditions. The MRWENO5 scheme on sparse grids and the corresponding single grids is used to solve the problem. The values of  $\epsilon$  in the MRWENO5 scheme and the MRWENO5 prolongation, and the time step size  $\Delta t_n$  are chosen in the similar way as Example 1. The CFL number is taken as 0.5. First we take the final time  $T = 0.5/\pi^2$  when the solution is still smooth. The  $L^\infty$  errors,  $L^1$  errors, and their numerical accuracy orders and CPU times are reported in Table 3. Again, we observe that the fifth-order accuracy is obtained along with the mesh refinement, for both the sparse-grid MRWENO5 scheme and the corresponding single-grid computations. Comparing the numerical errors of sparse-grid computations and the corresponding single-grid ones, it is found that on a relatively coarse grid, sparse-grid computation errors are greater than those on the corresponding single-grid, but with more refined meshes, the numerical errors are comparable to each other as the sparse-grid computation errors catch up to those of single-grid computations. For this 3D problem, the sparse-grid computations are much more efficient than the corresponding single-grid ones. From the CPU times reported in Table 3, it is observed that around 70% – 85% computational time is saved for simulations on sparse grids to reach a similar level of numerical errors to that on the corresponding single grids.

Next, we solve the problem till later time when shock waves appear in the solution. As the last example, the numerical solutions of the MRWENO5 scheme by sparse-grid and the corresponding single-grid simulations, at the time  $T = 5/\pi^2$  when large gradients start to form and at  $T = 2$  when the discontinuities and shock profile become sharper are presented in Fig. 2. Both 2D and 1D cutting plots of the numerical solutions on the plane  $z = 0$  are shown. The sparse-grid root grid  $N_r = 80$ , and the finest level  $N_L = 3$ . Hence the most refined mesh in the sparse grids or the corresponding single grid has  $640 \times 640 \times 640$  computational cells. From Fig. 2, we observe that the numerical solutions by the sparse-grid MRWENO5 scheme and its corresponding single-grid simulation are comparable, except that at  $T = 2$  when the shock profiles are sharp, the numerical solution by the sparse-grid MRWENO5 scheme has some overshoots at the shock locations. This phenomenon has also been observed in the classical sparse-grid WENO5 scheme in [37]. As discussed in [37], the overshoots are due to the final linear combination in the last step of the sparse-grid combination. Note that the numerical solutions by the MRWENO5 scheme on all sparse grids and the MRWENO5 prolongations are free of oscillations, and the oscillations appeared here just come from the final linear combination step which happens after the time evolutions are finished and is independent of the time evolution procedure. This linear combination is performed only once for a specific final time  $T$ . Hence the stability of the whole sparse-grid computations is not affected. How to resolve this issue in high-order sparse-grid WENO schemes is still an open problem and under investigation. In terms of computational efficiency, it takes 111 058.53 s and 434 827.47 s of CPU time to complete



**Table 3** Example 2, 3D Burgers' equation

Single-grid							
	$N_h \times N_h \times N_h$	$L^\infty$ error	Order	$L^1$ error	Order	CPU/s	
	$80 \times 80 \times 80$	$1.44 \times 10^{-6}$		$3.22 \times 10^{-7}$		3.62	
	$160 \times 160 \times 160$	$5.56 \times 10^{-8}$	4.70	$1.24 \times 10^{-8}$	4.70	83.36	
	$320 \times 320 \times 320$	$1.74 \times 10^{-9}$	5.00	$3.87 \times 10^{-10}$	5.00	2 630.67	
	$640 \times 640 \times 640$	$5.45 \times 10^{-11}$	4.99	$1.21 \times 10^{-11}$	4.99	55 508.71	
Sparse-grid							
$N_r$	$N_L$	$N_h \times N_h \times N_h$	$L^\infty$ error	Order	$L^1$ error	Order	CPU/s
10	3	$80 \times 80 \times 80$	$1.78 \times 10^{-5}$		$2.24 \times 10^{-6}$		1.71
20	3	$160 \times 160 \times 160$	$5.92 \times 10^{-8}$	8.23	$1.32 \times 10^{-8}$	7.40	22.48
40	3	$320 \times 320 \times 320$	$1.74 \times 10^{-9}$	5.09	$3.85 \times 10^{-10}$	5.10	429.81
80	3	$640 \times 640 \times 640$	$5.46 \times 10^{-11}$	5.00	$1.21 \times 10^{-11}$	4.99	10 606.14

MRWENO5 scheme, comparison of numerical errors and CPU times for computations on single-grid and sparse-grid. MRWENO5 interpolation for prolongation is employed in sparse-grid computations. Final time  $T = 0.5/\pi^2$ .  $N_r$ : number of cells in each spatial direction of a root grid.  $N_L$ : the finest level in a sparse-grid computation. CPU: CPU time for a complete simulation. CPU time unit: seconds

the simulation at  $T = 5/\pi^2$  and  $T = 2$ , respectively, in the sparse-grid computation, while 602 550.54 s and 2 545 996.15 s of CPU time are needed for finishing the simulation at  $T = 5/\pi^2$  and  $T = 2$ , respectively, in the corresponding single-grid computation. More than 80% CPU time is saved by performing the MRWENO5 simulation on the sparse grids here.

**Example 3 (A 3D viscous Burgers' equation).**

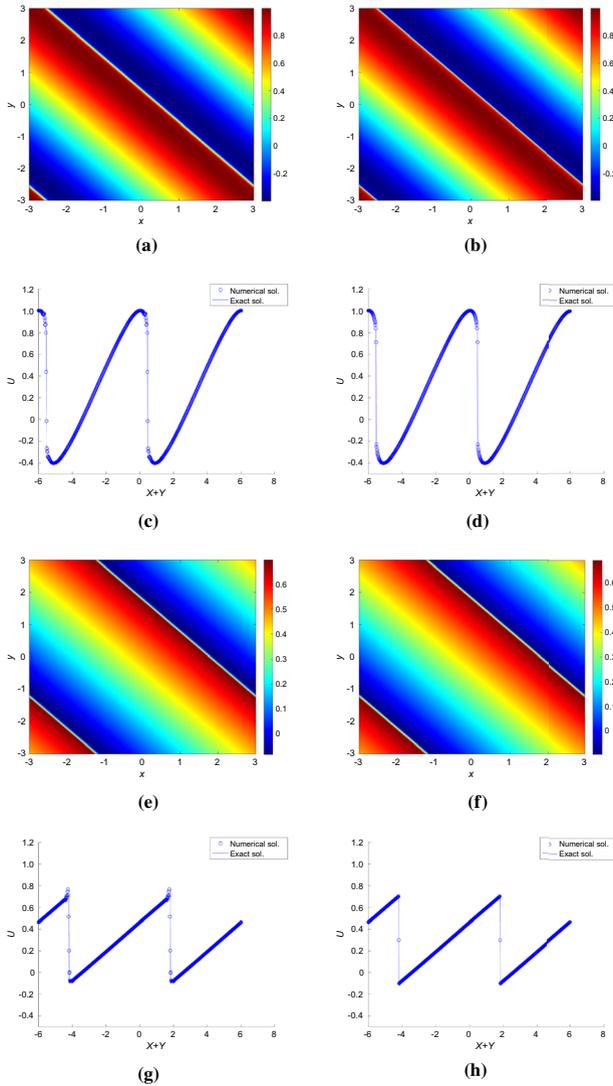
In Example 2, it is shown that there are some overshoots in the numerical solution of the sparse-grid MRWENO5 scheme for a sharp shock profile. In this example we add some small viscosity into the 3D Burgers' equation and solve the convection-dominated diffusion problem. It is interesting to observe the performance of the sparse-grid MRWENO5 scheme for such convection-dominated diffusion PDE. Consider the 3D viscous Burgers' equation:

$$\begin{cases} u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y + \left(\frac{u^2}{2}\right)_z = \nu(u_{xx} + u_{yy} + u_{zz}), \\ u(x, y, z, 0) = 0.3 + 0.7 \sin\left(\frac{\pi}{3}(x + y + z)\right) \end{cases} \quad (x, y, z) \in [-3, 3] \times [-3, 3] \times [-3, 3], \tag{20}$$

with periodic boundary conditions. The viscosity constant  $\nu = 0.01$  which gives a convection-dominated problem. The MRWENO5 discretization is applied to the hyperbolic terms. For the diffusion terms  $u_{xx} + u_{yy} + u_{zz}$ , we use a fourth-order central difference scheme, which has the following formulation, for example along a grid line of the  $x$ -direction:

$$(u_{xx})_i \approx \frac{1}{12(\Delta x)^2} (-u_{i+2} + 16u_{i+1} - 30u_i + 16u_{i-1} - u_{i-2}).$$

Simulations are performed on both sparse grids and the corresponding single grids to compare their results. MRWENO5 interpolation is used for sparse-grid prolongation. We solve the problem till the final time  $T = 2$  to compare with results of Example 2, and take  $\epsilon = 10^{-10}$  in MRWENO5 discretization and MRWENO5 prolongation as in Example 2.



**Fig. 2** Example 2, solution of 3D Burgers' equation by the MRWENO5 scheme on sparse grids ( $N_r = 80$  for root grid, finest level  $N_L = 3$  in the sparse-grid computation) and the corresponding  $640 \times 640 \times 640$  single grid, using the MRWENO5 interpolation for prolongation in the sparse-grid combination. (a)–(d): solution at  $T = 5/\pi^2$ ; (e)–(h): solution at  $T = 2$ ; (a), (c), (e), (g): sparse-grid results; (b), (d), (f), (h): single-grid results; (a), (b), (e), (f): 2D contour plots of the  $x$ - $y$  plane cutting at  $z = 0$ ; (c), (d), (g), (h): 1D cutting plot along  $x = y$  on the plane  $z = 0$ ; Circles: numerical solutions; lines: exact solution

The CFL number is taken as 0.4. In Fig. 3, both 2D and 1D cutting plots of the numerical solutions on the planes  $z = -3$  and  $z = 0$  are shown, for sparse-grid computation ( $N_r = 80$ ,  $N_L = 3$ ) and the corresponding  $640 \times 640 \times 640$  single-grid computation. We observe that the sparse-grid simulation results are similar to those on the single grid. Furthermore, the overshoots in the sparse-grid MRWENO5 results of Example 2 do not appear here any

more. Hence for this convection-dominated diffusion problem, the final linear combination in the last step of the sparse-grid combination does not cause any oscillations at the shock wave locations. In terms of computational efficiency, we observe 76%–89% CPU time savings when using sparse-grid computations vs. single-grid ones, as shown in Table 4 for different grids.

**Example 4 (A 4D Vlasov-Boltzmann transport equation).**

The related PDE systems in deterministic kinetic simulations can have up to six dimensions in spatial directions including both space and velocity variables. Hence they provide ideal benchmark problems to test numerical methods for solving high dimensional problems. WENO methods have been successfully applied to this class of PDEs, in, e.g., [4, 24]. Here we test the efficiency of the sparse-grid MRWENO5 scheme for simulating such problems. First, we solve a 4D Vlasov-Boltzmann transport equation in [10]:

$$f_t + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \mathbf{E}(\mathbf{x}) \cdot \nabla_{\mathbf{v}} f = L(f). \tag{21}$$

Here the unknown function  $f = f(t, \mathbf{x}, \mathbf{v})$  denotes the distribution of electrons which depends on space variables  $\mathbf{x} = (x_1, x_2)$ , velocity variables  $\mathbf{v} = (v_1, v_2)$ , and the time variable  $t$ .  $\mathbf{E}(\mathbf{x})$  is the external electric field which is given by a known electrostatic potential:

$$\mathbf{E}(\mathbf{x}) = -\nabla_{\mathbf{x}} \Phi(\mathbf{x}), \quad \Phi(\mathbf{x}) = \frac{|\mathbf{x}|^2}{2}.$$

$L(f)$  is the linear relaxation operator defined as

$$L(f) \triangleq \frac{\mu_{\infty}(\mathbf{v})\rho(t, \mathbf{x}) - f(t, \mathbf{x}, \mathbf{v})}{\tau},$$

where  $\mu_{\infty}(\mathbf{v})$  is the absolute Maxwellian distribution defined as

$$\mu_{\infty}(\mathbf{v}) \triangleq \frac{e^{-\frac{|\mathbf{v}|^2}{2\theta}}}{(2\pi\theta)^{d/2}},$$

and the macroscopic density  $\rho(t, \mathbf{x})$  is defined as

$$\rho(t, \mathbf{x}) = \int_{\mathbf{v}} f(t, \mathbf{x}, \mathbf{v}) d\mathbf{v}.$$

For the 4D case here the space dimension  $d = 2$ . The parameter  $\theta$  is the kinetic temperature and  $\tau = 1/k$  with  $k$  being the constant transition probability of scatters passing from one state into another one. They are taken as  $\theta = \tau = 1$  in this example. The domain is  $\Omega = [-5, 5] \times [-5, 5] \times [-5, 5] \times [-5, 5]$ . The problem has the initial condition

$$f(0, x_1, x_2, v_1, v_2) = \frac{1}{s} \sin\left(\frac{x_1^2}{2}\right)^2 \cos\left(\frac{x_2^2}{2}\right)^2 e^{-\frac{x_1^2 + x_2^2 + v_1^2 + v_2^2}{2}},$$

where  $s$  is the normalization constant such that  $\int_{\Omega} f(0, x_1, x_2, v_1, v_2) dx_1 dx_2 dv_1 dv_2 = 1$ . Zero boundary conditions are prescribed on the domain boundaries.

The sparse-grid MRWENO5 method is applied to this problem, and results are compared with the corresponding single-grid computation. As in Example 1, we choose the values of  $\epsilon$  in both the MRWENO5 scheme and the MRWENO5 prolongation as a function of spatial grid sizes. The CFL number is taken to be 0.4 in the simulations. This 4D PDE is

solved till the final time  $T = 3$ . The plots of numerical solutions of  $f$  in different 2D planes with fixed third and fourth direction coordinates at different time  $t$  are shown in Figs. 4 and 5, on  $160 \times 160 \times 160 \times 160$  grid ( $N_r = 20$  for the root grid, the finest level  $N_L = 3$  in the sparse-grid computation). It is clearly observed that the sparse-grid MRWENO5 scheme and its corresponding single-grid computation generate similar results. On the other hand, the sparse-grid simulations are significantly more efficient than the corresponding single-grid simulations. Table 5 reports the simulation CPU times on two different meshes. 95% CPU time is saved by carrying out sparse-grid computations rather than single-grid simulations in this example.

**Example 5 (A 3D Vlasov-Maxwell system).**

As the second example to test the efficiency of the sparse-grid MRWENO5 scheme for simulating kinetic problems, we solve a 3D example of the Vlasov-Maxwell system from [26], which is a simplified single species Vlasov-Maxwell system. It has one spatial variable and two velocity variables, by assuming that the system is uniform in other variable directions of the whole six-dimensional (6D) domain. The system is

$$\begin{cases} f_t + \xi_2 f_{x_2} + (E_1 + \xi_2 B_3) f_{\xi_1} + (E_2 - \xi_1 B_3) f_{\xi_2} = 0, \\ \frac{\partial B_3}{\partial t} = \frac{\partial E_1}{\partial x_2}, \quad \frac{\partial E_1}{\partial t} = \frac{\partial B_3}{\partial x_2} - j_1, \quad \frac{\partial E_2}{\partial t} = -j_2. \end{cases} \tag{22}$$

Here the system has one physical space variable  $x_2$  and two velocity variables  $\xi_1, \xi_2$ , and the domain is  $\Omega_x \times \Omega_\xi$  with  $\Omega_x$  the physical space and  $\Omega_\xi$  the velocity space.  $x_2 \in \Omega_x$  and  $(\xi_1, \xi_2) \in \Omega_\xi$ .  $f = f(x_2, \xi_1, \xi_2, t)$  is the distribution of electrons.  $E_1 = E_1(x_2, t)$  and  $E_2 = E_2(x_2, t)$  are the electric field components.  $B_3 = B_3(x_2, t)$  is the magnetic field component. Hence, the system has a 2D electric field  $\mathbf{E} = (E_1(x_2, t), E_2(x_2, t), 0)$  and a 1D magnetic field  $\mathbf{B} = (0, 0, B_3(x_2, t))$  in the physical space. The current densities  $j_1(x_2, t)$  and  $j_2(x_2, t)$  are given by

$$j_1 = \iint_{\Omega_\xi} f(x_2, \xi_1, \xi_2, t) \xi_1 d\xi_1 d\xi_2, \quad j_2 = \iint_{\Omega_\xi} f(x_2, \xi_1, \xi_2, t) \xi_2 d\xi_1 d\xi_2.$$

The system is initialized by

$$f(x_2, \xi_1, \xi_2, 0) = \frac{1}{\pi\beta} e^{-\xi_2^2/\beta} [\delta e^{-(\xi_1 - v_{0,1})^2/\beta} + (1 - \delta) e^{-(\xi_1 + v_{0,2})^2/\beta}],$$

$$E_1(x_2, 0) = E_2(x_2, 0) = 0, \quad B_3(x_2, 0) = b \sin(k_0 x_2).$$

Here the computational domain is taken as  $\Omega_x = [0, 2\pi/k_0]$  and  $\Omega_\xi = [-1.2, 1.2]^2$ , subject to periodic boundary conditions. The parameters in the system are the same as in [26]:  $\delta = 0.5$ ,  $b = 0.001$ ,  $\beta = 0.01$ ,  $v_{0,1} = v_{0,2} = 0.3$ , and  $k_0 = 0.2$ . In this technical note, we focus on testing the efficiency of the designed sparse-grid MRWENO5 scheme in solving this 3D kinetic problem, and refer to [26] about the introduction of the physical background of the system and these parameters.

The sparse-grid MRWENO5 scheme is applied to this 3D problem, and we compare the sparse-grid simulation results with the corresponding single-grid computation results. The system is solved till the final time  $T = 50$ . The values of  $\epsilon$  are chosen to be  $\epsilon = 10^{-10}$  as in [34], for both the MRWENO5 scheme and the MRWENO5 prolongation. The CFL number is taken to be 0.4 in the simulations. To preserve the positivity

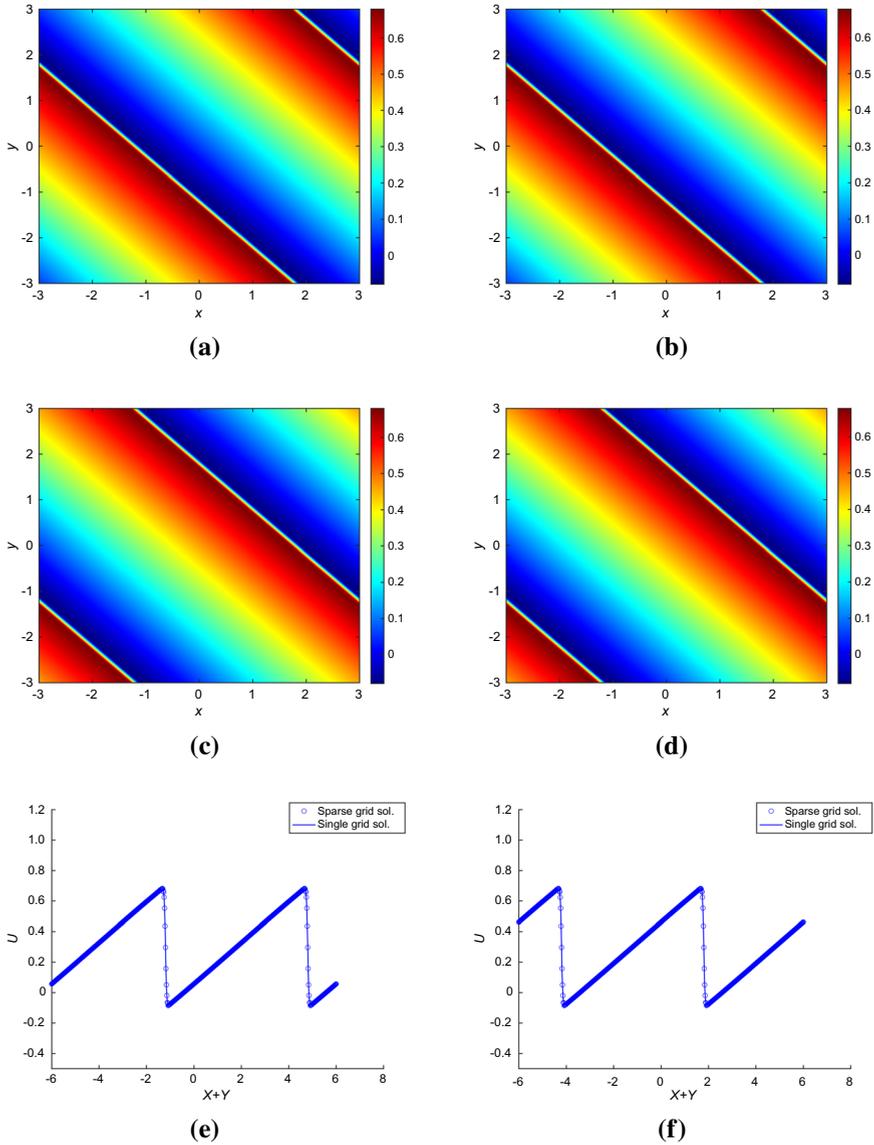
**Table 4** Example 3, 3D viscous Burgers' equation

$N_h \times N_h \times N_h$	CPU time on sparse-grid	CPU time on single-grid
$80 \times 80 \times 80$	33.33	140.63
$160 \times 160 \times 160$	523.39	2 975.09
$320 \times 320 \times 320$	7 918.83	69 204.82
$640 \times 640 \times 640$	179 061.77	796 560.45

Comparison of CPU times for MRWENO5 computations on sparse grids and the corresponding single grids. Final time  $T = 2$ . CFL number is 0.4. The finest level  $N_L = 3$  in sparse-grid computations. CPU time unit: seconds

property of the numerical solutions  $f$ , we apply a conservative bound-preserving sweeping technique in [20]. The bound-preserving technique in [20] is a simple sweeping algorithm which acts as a postprocessing step and is independent of the underlying numerical schemes. The algorithm first pre-sets an order for all grid points, then a “right” sweep is followed by a “left” sweep to correct any numerical values which are beyond the required bounds. In each sweeping, if the numerical value at a grid point is beyond the bound, then both that value and the next grid point’s value are corrected using the given positive weights and the bound values. It is shown in [20] that this simple algorithm can preserve the bounds of the numerical solution and keep the accuracy of the underlying numerical schemes. We adopt this algorithm here as a final step to postprocess the obtained solutions and preserve the lower bound of the numerical solutions of  $f$  to be 0. The plots of numerical solutions of  $f$  in the 2D plane with a fixed  $x_2$  coordinate at different time  $t$  are shown in Fig. 6, on a  $160 \times 160 \times 160$  grid ( $N_r = 20$  for the root grid, the finest level  $N_L = 3$  in the sparse-grid computation). Again, we observe that the sparse-grid MRWENO5 scheme and its corresponding single-grid computation generate similar results, and the sharp gradient of the solution is resolved very well and stably in the sparse-grid MRWENO5 scheme for this Vlasov-Maxwell system. However, in terms of computational efficiency, the sparse-grid computations are much more efficient than the corresponding single-grid ones. Table 6 reports the CPU time costs on two different meshes. About 80% CPU time is saved by carrying out sparse-grid simulations rather than single-grid simulations in this 3D example.

Here we have shown the efficiency and CPU time saving of performing the MRWENO5 simulations on sparse grids by solving a simplified 3D Vlasov-Maxwell system. It indicates that the sparse-grid MRWENO5 scheme could be a promising method in simulating complex kinetic systems. Next we carry out some preliminary studies about conservation property of the sparse-grid MRWENO5 scheme on some macroscopic quantities of this system. As in [26], we define the scaled total mass as  $m = \frac{1}{L_y} \int_{\Omega} f d\xi_1 d\xi_2 dx_2$ , where  $\Omega = \Omega_x \times \Omega_x$  and  $L_y = 2\pi/k_0$ .  $L_y$  is the size of the physical domain  $\Omega_x$ . The scaled kinetic energy  $K = K_1 + K_2$ , where  $K_1, K_2$  are the scaled kinetic energies in each direction and they are defined as  $K_1 = \frac{1}{2L_y} \int_{\Omega} f \xi_1^2 d\xi_1 d\xi_2 dx_2, K_2 = \frac{1}{2L_y} \int_{\Omega} f \xi_2^2 d\xi_1 d\xi_2 dx_2$ . The scaled electric energy  $\mathcal{E} = \mathcal{E}_1 + \mathcal{E}_2$ , where  $\mathcal{E}_1, \mathcal{E}_2$  are the scaled electric energies in each direction and they are defined as  $\mathcal{E}_1 = \frac{1}{2L_y} \int_{\Omega_x} E_1^2 dx_2, \mathcal{E}_2 = \frac{1}{2L_y} \int_{\Omega_x} E_2^2 dx_2$ . The scaled magnetic energy is  $\mathcal{B}_3 = \frac{1}{2L_y} \int_{\Omega_x} B_3^2 dx_2$ . Hence the scaled total energy is calculated as  $K + \mathcal{E} + \mathcal{B}_3$ . The scaled total momentum



**Fig. 3** Example 3, solution of 3D viscous Burgers' equation by the MRWENO5 scheme on sparse grids ( $N_r = 80$  for root grid, finest level  $N_L = 3$  in the sparse-grid computation) and the corresponding  $640 \times 640 \times 640$  single grid, using the MRWENO5 interpolation for prolongation in the sparse-grid combination. Final time  $T = 2$ . (a), (c): sparse-grid results; (b), (d): single-grid results. (a), (b): 2D contour plots of the  $x$ - $y$  plane cutting at  $z = -3$ ; (c), (d): 2D contour plots of the  $x$ - $y$  plane cutting at  $z = 0$ ; (e): 1D cutting plot along  $x = y$  on the plane  $z = -3$ ; (f): 1D cutting plot along  $x = y$  on the plane  $z = 0$ . Solid lines: single-grid solution; circles: sparse-grid solution

$P = P_1 + P_2$ , where the components  $P_1 = \frac{1}{L_y} \left( \int_{\Omega} \xi_1 f d\xi_1 d\xi_2 dx_2 + \int_{\Omega_x} E_2 B_3 dx_2 \right)$  and  $P_2 = \frac{1}{L_y} \left( \int_{\Omega} \xi_2 f d\xi_1 d\xi_2 dx_2 - \int_{\Omega_x} E_1 B_3 dx_2 \right)$ . In Fig. 7, time evolutions of these scaled quan-

**Table 5** Example 4, 4D Vlasov–Boltzman equation

$N_h \times N_h \times N_h \times N_h$	CPU time on sparse-grid	CPU time on single-grid
$80 \times 80 \times 80 \times 80$	5 179.67	84 825.69
$160 \times 160 \times 160 \times 160$	112 655.92	2 277 988.00

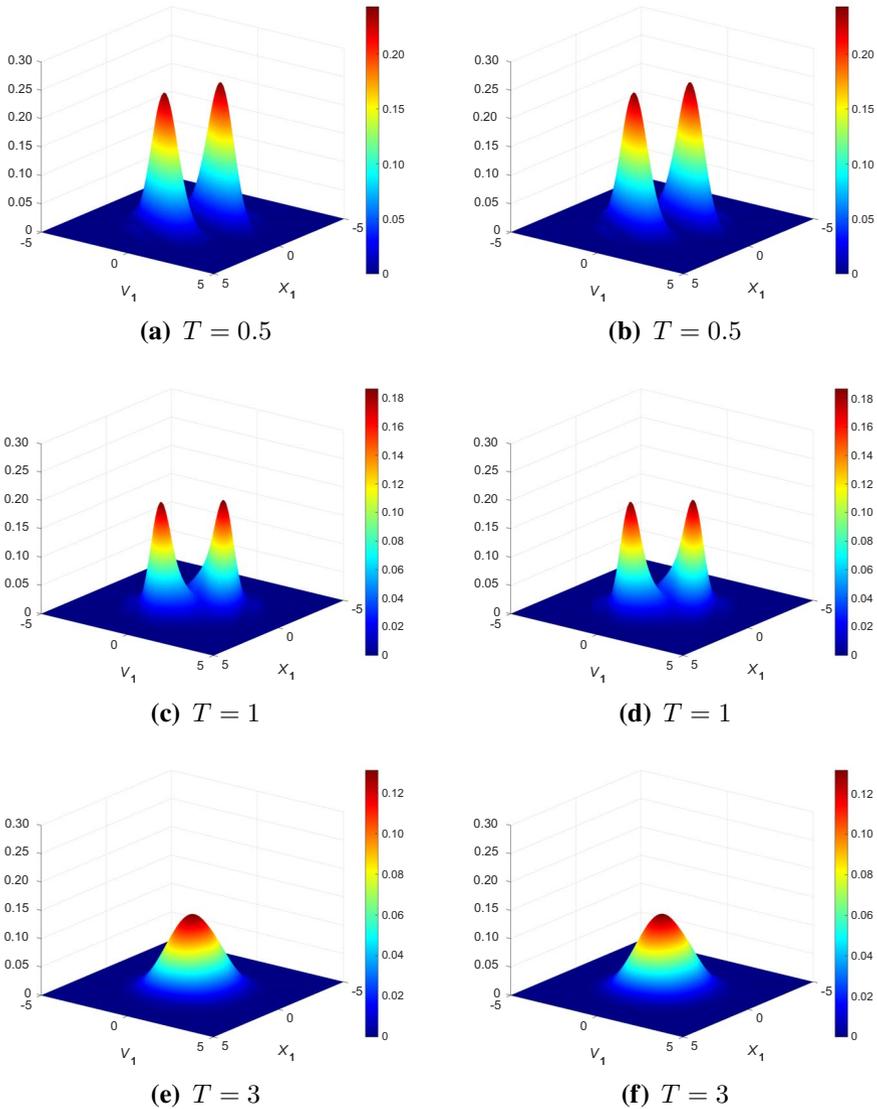
Comparison of CPU times for MRWENO5 computations on sparse grids and the corresponding single grids. Final time  $T = 3$ . CFL number is 0.4. The finest level  $N_L = 3$  in sparse-grid computations. CPU time unit: seconds

ties including the total mass, total energy, total momentum, and kinetic, magnetic, electric energies of the numerical solution by the sparse-grid MRWENO5 scheme with  $N_r = 20$  for the root grid and the finest level  $N_L = 3$  are reported. For the time period considered here, we observe that the largest error for the total mass is less than  $5 \times 10^{-4}$ . The largest errors for the total energy and the total momentum are on the order of  $10^{-4}$ . Here we would like to emphasize that the Vlasov–Maxwell system is a very complex problem and there are many challenging issues in the physics it models, its mathematical theory, and numerical method developments. Although beyond the scope of this technical note, more detailed and in-depth studies on applications of the proposed sparse-grid MRWENO5 scheme to the Vlasov–Maxwell system in kinetic simulations, for example, studying the performance of the method in solving the system for a long time period, improving the method’s conservation errors for the physical invariants, comparing the method with other methods for solving the Vlasov–Maxwell system, etc., are important and will be carried out in our future research.

## 4 Conclusions

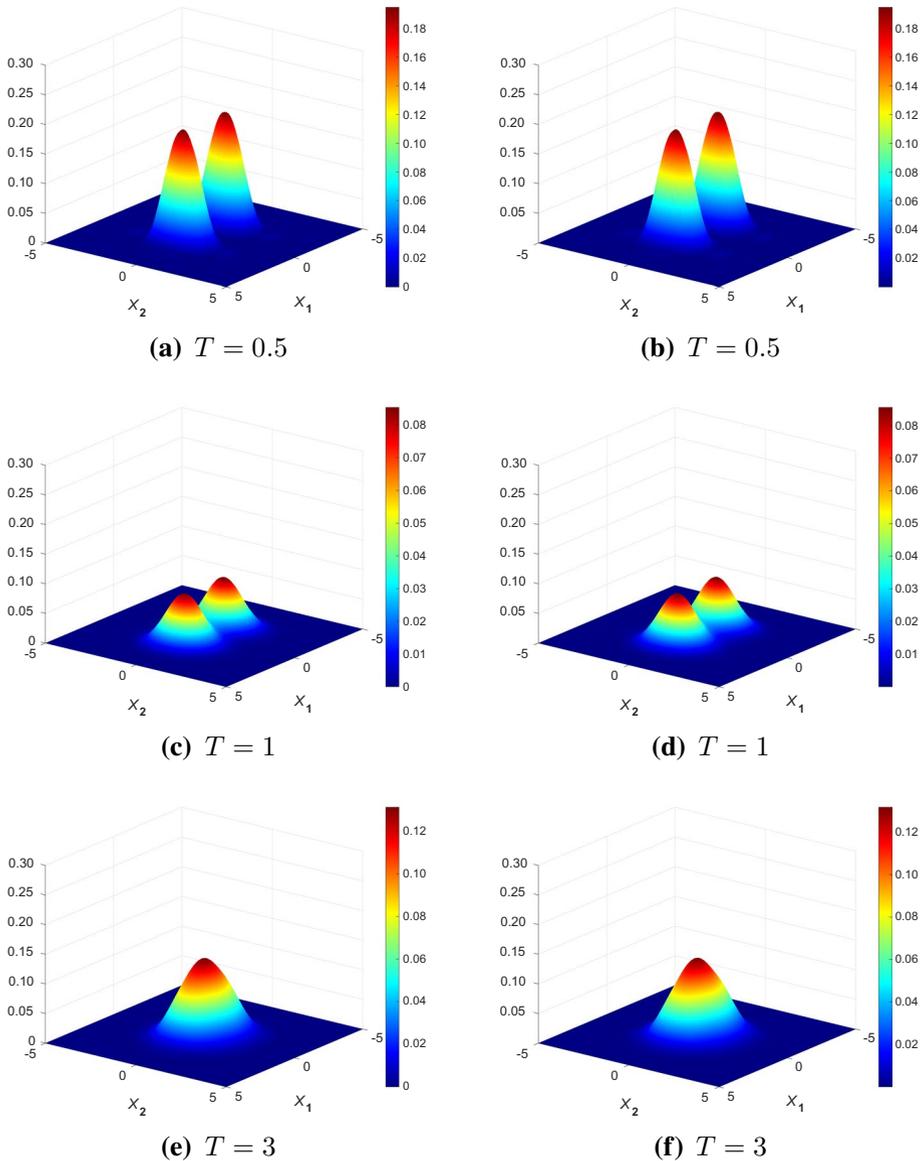
In this technical note, we extend our previous sparse-grid WENO approach in [22, 37] to the recently developed finite difference multi-resolution WENO scheme in [34] for efficiently solving multidimensional problems. To obtain a robust algorithm, the MRWENO5 interpolation is designed in the prolongation step of sparse-grid combination technique. Numerical experiments on 2D, 3D, and 4D problems are performed for the sparse-grid multi-resolution WENO method to show that a much more efficient algorithm than a regular method on single grids to solve the multidimensional equations is achieved. A large amount of CPU time costs are saved for higher dimensional problems. For example, the sparse-grid MRWENO5 scheme saves 95% CPU time for solving a 4D Vlasov equation, by comparing with the corresponding single-grid simulations.

As discussed in [37], there are still quite a few open problems to be investigated further for the method. For example, due to the high nonlinearity of the MRWENO5 scheme, it is still unknown how to perform theoretical error analysis for the nonlinear sparse-grid scheme, which has been done for the linear schemes in solving linear PDEs in the literature. Another open question is that for some problems (e.g., the 3D inviscid Burgers’ equation in this technical note), some oscillations or noises may appear around the shock locations when the shock wave is strong or its profile is very sharp, due to the final linear combination formulas in the last step of the sparse-grid combination technique. Although this combination is done only once for a specific final time  $T$  and it will



**Fig. 4** Example 4, solutions  $f$  of the 4D Vlasov-Boltzmann equation at different time  $T$  by the MRWENO5 scheme on sparse grids ( $N_r = 20$  for root grid, finest level  $N_L = 3$  in the sparse-grid computation) and the corresponding  $160 \times 160 \times 160$  single grid, using the MRWENO5 interpolation for prolongation in the sparse-grid combination. 2D cuts of solutions in the  $x_1$ - $v_1$  plane at  $x_2 = v_2 = 0$ . CFL = 0.4. (a), (c), (e): sparse-grid results; (b), (d), (f): single-grid results. (a), (b): final time  $T = 0.5$ ; (c), (d): final time  $T = 1$ ; (e), (f): final time  $T = 3$

not cause any stability issue in the whole simulations, it is crucial to resolve this issue in our next research. One possible way to tackle this problem is to develop sparse-grid WENO methods directly based on a set of nested grids and multiresolution analysis, and not to use the sparse-grid combination technique. This will be a very interesting topic.



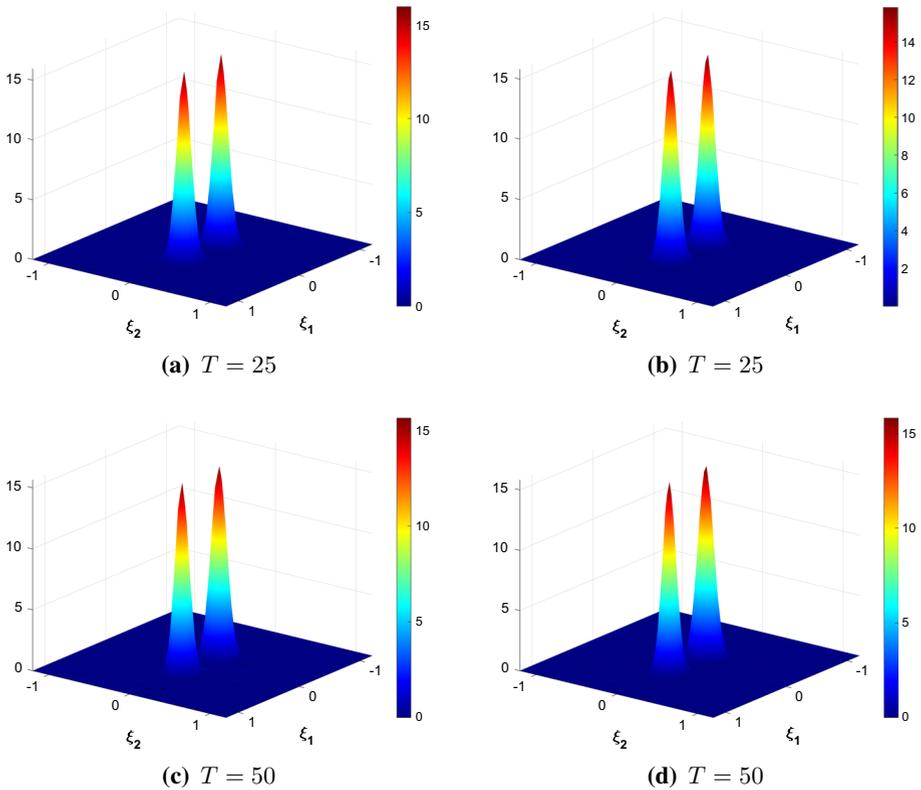
**Fig. 5** Example 4, solutions  $f$  of the 4D Vlasov-Boltzmann equation at different time  $T$  by the MRWENO5 scheme on sparse grids ( $N_r = 20$  for root grid, finest level  $N_L = 3$  in the sparse-grid computation) and the corresponding  $160 \times 160 \times 160 \times 160$  single grid, using the MRWENO5 interpolation for prolongation in sparse-grid combination. 2D cuts of solutions in the  $x_1$ - $x_2$  plane at  $v_1 = v_2 = 0$ . CFL = 0.4. (a), (c), (e): sparse-grid results; (b), (d), (f): single-grid results. (a), (b): final time  $T = 0.5$ ; (c), (d): final time  $T = 1$ ; (e), (f): final time  $T = 3$

Multiresolution analysis has been successfully used to develop sparse-grid discontinuous Galerkin methods in [10, 11, 26]. To develop sparse-grid MRWENO methods along this way, we will use the WENO reconstruction procedure based on the hierarchical

**Table 6** Example 5, 3D Vlasov-Maxwell system

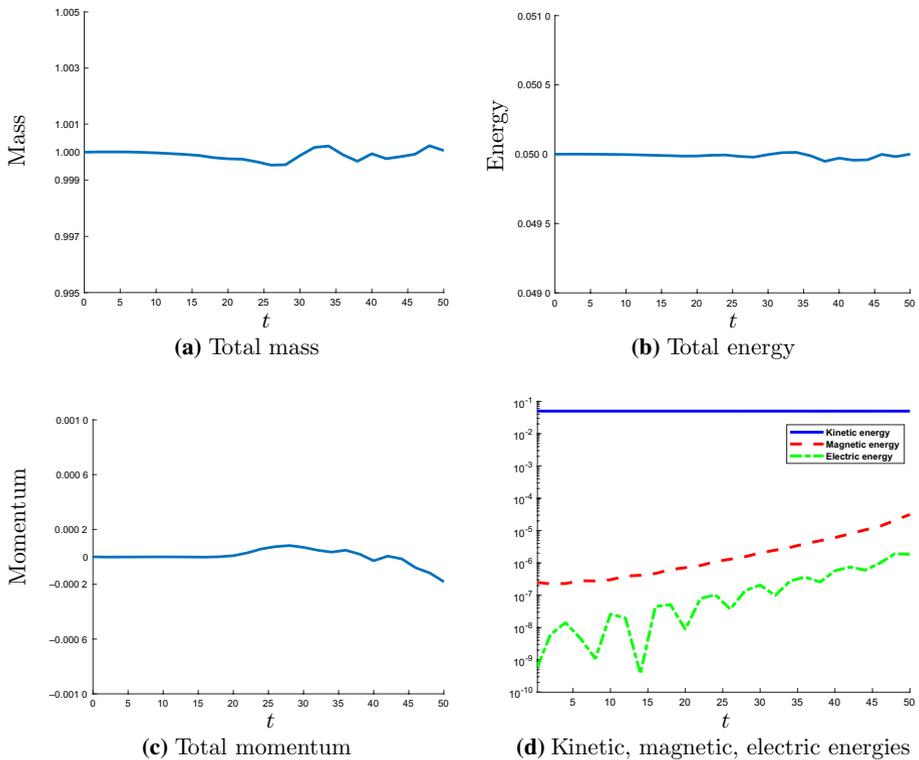
$N_h \times N_h \times N_h$	CPU time on sparse-grid	CPU time on single-grid
$80 \times 80 \times 80$	196.43	890.55
$160 \times 160 \times 160$	2 764.62	13 347.87

Comparison of CPU times for MRWENO5 computations on sparse grids and the corresponding single grids. Final time  $T = 50$ . CFL number is 0.4. The finest level  $N_L = 3$  in sparse-grid computations. CPU time unit: seconds



**Fig. 6** Example 5, solution  $f$  of the 3D Vlasov-Maxwell system at different time  $T$  by the MRWENO5 scheme on sparse grids ( $N_r = 20$  for root grid, finest level  $N_L = 3$  in the sparse-grid computation) and the corresponding  $160 \times 160 \times 160$  single grid, using the MRWENO5 interpolation for prolongation in the sparse-grid combination. CFL = 0.4. 2D cuts of solutions in the  $\xi_1 - \xi_2$  plane at  $x_2 = \pi/k_0$ . (a), (c): sparse-grid results; (b), (d): single-grid results. (a), (b): final time  $T = 25$ ; (c), (d): final time  $T = 50$

basis in [14] and incorporate high-order MRWENO approximations in it. Due to the hierarchical structure of the reconstruction procedure in [14], it can be naturally applied to sparse grids, and a tensor product approach can be used for multidimensional problems. Furthermore, this kind of hierarchical basis WENO methods on sparse grids will



**Fig. 7** Example 5, the Vlasov-Maxwell system. Time evolution of total mass, total energy, total momentum, and kinetic, magnetic, electric energies of the numerical solution by the MRWENO5 scheme on sparse grids ( $N_r = 20$  for root grid, finest level  $N_L = 3$  in the sparse-grid computation), using the MRWENO5 interpolation for prolongation in the sparse-grid combination. (a): total mass; (b): total energy; (c): total momentum; (d): kinetic, magnetic, and electric energies

be promising to be combined with adaptive approaches, since the hierarchical basis or the hierarchical structure in the schemes provide a convenient way to design error indicators in adaptive methods, as in [1, 11]. Detailed development of this new method is our next work.

## Compliance with Ethical Standards

**Conflict of Interest** The authors declare that there is no conflict of interest.

## References

1. Alves, M.A., Cruz, P., Mendes, A., Magalhães, F.D., Pinho, F.T., Oliveira, P.J.: Adaptive multiresolution approach for solution of hyperbolic PDEs. *Comput. Methods Appl. Mech. Eng.* **191**, 3909–3928 (2002)
2. Bungartz, H.-J., Griebel, M.: Sparse grids. *Acta Numerica* **13**, 147–269 (2004)

3. Capdeville, G.: A central WENO scheme for solving hyperbolic conservation laws on non-uniform meshes. *J. Comput. Phys.* **227**, 2977–3014 (2008)
4. Carrillo, J.A., Gamba, I.M., Majorana, A., Shu, C.-W.: A WENO-solver for the transients of Boltzmann-Poisson system for semiconductor devices: performance and comparisons with Monte Carlo methods. *J. Comput. Phys.* **184**, 498–525 (2003)
5. Castro, M., Costa, B., Don, W.S.: High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws. *J. Comput. Phys.* **230**, 1766–1792 (2011)
6. Don, W.-S., Borges, R.: Accuracy of the weighted essentially non-oscillatory conservative finite difference schemes. *J. Comput. Phys.* **250**, 347–372 (2013)
7. Dumbser, M., Käser, M., Titarev, V.A., Toro, E.F.: Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *J. Comput. Phys.* **226**, 204–243 (2007)
8. Gottlieb, S., Shu, C.-W., Tadmor, E.: Strong stability-preserving high-order time discretization methods. *SIAM Rev.* **43**, 89–112 (2001)
9. Griebel, M., Schneider, M., Zenger, C.: A combination technique for the solution of sparse grid problems. In: Beauwens, R., de Groen, P. (eds) *Iterative Methods in Linear Algebra*, pp. 263–281. North-Holland, Amsterdam (1992)
10. Guo, W., Cheng, Y.: A sparse grid discontinuous Galerkin method for high-dimensional transport equations and its application to kinetic simulations. *SIAM J. Sci. Comput.* **38**, A3381–A3409 (2016)
11. Guo, W., Cheng, Y.: An adaptive multiresolution discontinuous Galerkin method for time-dependent transport equations in multidimensions. *SIAM J. Sci. Comput.* **39**, A2962–A2992 (2017)
12. Henrick, A., Aslam, T., Powers, J.: Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points. *J. Comput. Phys.* **207**, 542–567 (2005)
13. Jiang, G.-S., Shu, C.-W.: Efficient implementation of weighted ENO schemes. *J. Comput. Phys.* **126**, 202–228 (1996)
14. Kolb, O.: A third order hierarchical basis WENO interpolation for sparse grids with application to conservation laws with uncertain data. *J. Sci. Comput.* **74**, 1480–1503 (2018)
15. Lastdrager, B., Koren, B., Verwer, J.: Solution of time-dependent advection-diffusion problems with the sparse-grid combination technique and a rosenbrock solver. *Comput. Methods Appl. Math.* **1**, 86–99 (2001)
16. Lastdrager, B., Koren, B., Verwer, J.: The sparse-grid combination technique applied to time-dependent advection problems. *Appl. Numer. Math.* **38**, 377–401 (2001)
17. Levy, D., Puppo, G., Russo, G.: Central WENO schemes for hyperbolic systems of conservation laws. *Math. Model. Numer. Anal.* **33**, 547–571 (1999)
18. Li, L., Zhu, J., Zhang, Y.-T.: Absolutely convergent fixed-point fast sweeping WENO methods for steady state of hyperbolic conservation laws. *J. Comput. Phys.* **443**, 110516 (2021)
19. Liu, X.-D., Osher, S., Chan, T.: Weighted essentially non-oscillatory schemes. *J. Comput. Phys.* **115**, 200–212 (1994)
20. Liu, Y., Cheng, Y., Shu, C.-W.: A simple bound-preserving sweeping technique for conservative numerical approximations. *J. Sci. Comput.* **73**, 1028–1071 (2017)
21. Liu, Y., Zhang, Y.-T.: A robust reconstruction for unstructured WENO schemes. *J. Sci. Comput.* **54**, 603–621 (2013)
22. Lu, D., Chen, S., Zhang, Y.-T.: Third order WENO scheme on sparse grids for hyperbolic equations. *Pure Appl. Math. Q.* **14**, 57–86 (2018)
23. Lu, D., Zhang, Y.-T.: Krylov integration factor method on sparse grids for high spatial dimension convection-diffusion equations. *J. Sci. Comput.* **69**, 736–763 (2016)
24. Qiu, J.-M., Christlieb, A.: A conservative high order semi-Lagrangian WENO method for the Vlasov equation. *J. Comput. Phys.* **229**, 1130–1149 (2010)
25. Shu, C.-W.: Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In: Cockburn, B., Johnson, C., Shu, C.-W., Tadmor, E. (eds) *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations. Lecture Notes in Mathematics, Volume 1697*. Springer (1998)
26. Tao, Z., Guo, W., Cheng, Y.: Sparse grid discontinuous Galerkin methods for the Vlasov-Maxwell system. *J. Comput. Phys. X* **3**, 100022 (2019)
27. Yamaleev, N., Carpenter, M.: A systematic methodology for constructing high-order energy stable WENO schemes. *J. Comput. Phys.* **228**, 4248–4272 (2009)
28. Zenger, C.: Sparse grids. In: Hackbusch, W. (ed) *Notes on Numerical Fluid Mechanics*, vol. 31, pp. 241–251. Vieweg, Braunschweig (1991)
29. Zhang, S., Jiang, S., Zhang, Y.-T., Shu, C.-W.: The mechanism of sound generation in the interaction between a shock wave and two counter rotating vortices. *Phys. Fluids* **21**, 076101 (2009)

30. Zhang, Y.-T., Shu, C.-W.: High order WENO schemes for Hamilton-Jacobi equations on triangular meshes. *SIAM J. Sci. Comput.* **24**, 1005–1030 (2003)
31. Zhang, Y.-T., Shu, C.-W.: Third order WENO scheme on three dimensional tetrahedral meshes. *Commun. Comput. Phys.* **5**, 836–848 (2009)
32. Zhang, Y.-T., Shu, C.-W., Zhou, Y.: Effects of shock waves on Rayleigh-Taylor instability. *Phys. Plasmas* **13**, 062705 (2006)
33. Zhu, J., Qiu, J.: A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws. *J. Comput. Phys.* **318**, 110–121 (2016)
34. Zhu, J., Shu, C.-W.: A new type of multi-resolution WENO schemes with increasingly higher order of accuracy. *J. Comput. Phys.* **375**, 659–683 (2018)
35. Zhu, J., Shu, C.-W.: A new type of third-order finite volume multi-resolution WENO schemes on tetrahedral meshes. *J. Comput. Phys.* **406**, 109212 (2020)
36. Zhu, J., Shu, C.-W.: Convergence to steady-state solutions of the new type of high-order multi-resolution WENO schemes: a numerical study. *Commun. Appl. Math. Comput.* **2**, 429–460 (2020)
37. Zhu, X., Zhang, Y.-T.: Fast sparse grid simulations of fifth order WENO scheme for high dimensional hyperbolic PDEs. *J. Sci. Comput.* **87**, 44 (2021)