ICCFD13-2026-118

Ensemble Kalman Inversion Optimized Low-storage Compact Finite Volume Method on Unstructured Grids

Yanqing Shi*, Qian Wang*
Corresponding author: qian.wang@csrc.ac.cn
* Beijing Computational Science Research Center, Beijing 100193, China

Abstract: This paper presents an optimized low-storage variational reconstruction for high-order finite volume methods on unstructured grids for compressible flow simulations. The reconstruction polynomial in each cell is obtained by minimizing a cost function defined through interfacial jumps. Unlike conventional formulations based on surface integrals of jump terms, the proposed method evaluates the weighted sum of the jumps of the reconstruction polynomial and its spatial derivatives only at interface centers. This enables efficient matrix-free evaluation of the right-hand side of the local linear reconstruction equation system, avoiding the storage of large matrices and substantially reducing memory consumption. The derivative weights in the reconstruction are optimized using an ensemble Kalman inversion framework. The optimization maximizes an efficiency metric defined as the reciprocal of the product of solution error and computational cost, evaluated using a linear advection problem with a broadband wave-packet initial condition. This design promotes both high resolution and computational efficiency. Numerical results for inviscid flow problems show that the optimized low-storage variational finite volume scheme is remarkably more efficient than both reference and non-optimized schemes, demonstrating the effectiveness of the proposed optimization strategy.

Keywords: Finite Volume, Ensemble Kalman Inversion, Low-storage Reconstruction, Parameter Optimization.

1. Introduction

With advances in computer hardware and numerical methods, computational fluid dynamics (CFD) has emerged as an effective tool for investigating flow problems in both scientific and industrial fields. Although the second-order finite volume method on unstructured grids [1] remains the mainstream approach in engineering, its highly dissipative and highly dispersive nature limits its application in high-resolution flow simulations. In the past few decades, various high-order numerical methods on unstructured grids, such as high-order FV [2–5], discontinuous Galerkin (DG) [6, 7], hybrid FV/DG [8–10], residual distribution (RD) [11] and flux reconstruction (FR) [12–14], have been extensively developed. Compared with their second-order counterparts, high-order methods generally exhibit lower dissipation and dispersion errors, allowing them to resolve multiscale flow field features more accurately even on relatively coarse grids. Furthermore, these methods can also handle complex geometries. Consequently, developing high-order numerical methods for unstructured grids has become an important approach to improve the accuracy and efficiency of CFD simulations, and deepen our understanding of complex flow mechanisms.

Variational reconstruction is one of the compact reconstructions [4, 15–17] that can achieve high-order accuracy on a compact stencil, thereby overcoming the bottleneck associated with large reconstruction stencils in the development of efficient high-order finite volume methods. The principle behind this reconstruction method is to define a cost function that measures the smoothness of the piecewise polynomial distribution across the computational domain. The piecewise polynomial distribution is obtained by minimizing a cost function that is the summation of intefacial jump integrals (IJIs). On each cell interface, an IJI is defined to measure the jumps of the reconstruction polynomial and its spatial derivatives, and

the face integral terms in IJIs can be computed exactly using Gauss quadrature formulas. Therefore, efficiently implementing variational reconstruction typically requires the pre-computing and storage of large matrices, leading to substantial memory costs, particularly in the case of multidimensional polyhedral cells. The weights of the spatial derivative jump integral terms, termed as derivative weights, are the free parameters of the variational reconstruction, and different derivative weights will result in different reconstruction schemes. In [18] the derivative weights are generalized using spectral properties obtained via Fourier analysis on structured meshes. Inspired by Fourier analysis, Zhou et. al [19] trained an artificial neural network (ANN) with sampled sine waves as initial conditions to minimize linear advection solution errors on compact stencils, enabling prediction of optimal interface derivative weights from local mesh geometry. This avoids the difficulty of deriving such optimizations analytically on unstructured grids.

In this paper, we porpose a low-storage variational reconstruction. Furthermore, we develop a posterior optimization framework based on ensemble Kalman inversion (EKI) for numerical schemes on unstructured grids. For the low-storage variational reconstruction, we define the cost function as the sum of interfacial jump functions, which measure the discontinuities of reconstruction polynomial and its spatial derivatives between adjacent cells only at the interface centers. These quantities can be computed efficiently on the fly, eliminating the need for matrix storage and resulting in low memory usage. As described in [20], the derivative weights should be carefully selected to maintain the computational rotational invariance of reconstruction. Unlike in previous schemes [18, 19, 21], where the derivative weights are all independent, partial derivatives of the same order will now share the same weights. This reduces the number of parameters to be optimized while preserving the rotational invariance, making it more suitable for EKI optimization.

Ensemble Kalman inversion is an ensemble-based iterative parameter inversion method that has been widely used for identifying unknown model parameters [22]. In this paper, the derivative weights corresponding to different geometric partitioning are optimized by minimising an objective function defined as the reciprocal of efficiency calculated by solving the linear advection equation with a wave pocket composed by a series of weighted sine waves as the initial condition. Therefore, the optimization of derivative weights can be formulated as a least-squares problem. From this perspective, this paper treats EKI is not as a strict posterior sampling method, but as a gradient-free and adjoint-free ensemble optimization algorithm [23], which is particularly well-suited for strongly nonlinear, non-differentiable, or black-box forward models. Specifically, for each geometric discretization, two separate meshes, representing high- and low-quality respectively, are generated for the computational domain to ensure the optimized schemes are more robust. In each iteration, the same initial condition is applied to both low- and high-quality meshes, and the governing equation is solved using schemes defined by weights from an ensemble of parameter particles, over a sufficient duration to allow the wave to propagate through the entire computational domain. Subsequently, the solution errors in each cells and the total CPU time can be obtained, which are the corresponding predicted observations. We then apply the EKI update formula to adjust the ensemble of parameter particles based on the discrepancy between the predicted and target observations. As the iterations progress, the ensemble gradually converges towards the parameter region that best matches the target, ultimately yielding the desired weights. Numerical results for two-dimensional Euler test cases demonstrate that the optimized low-storage variational finite volume (VFV) schemes are remarkably more efficient than the reference one.

The remainder of this paper is organized as follows. Section 2 presents the low-storage variational finite volume method on unstructured grids. Section 3 presents the optimization framework based on EKI for the low-storage variational reconstruction. Numerical results are given in Section 4, conclusion and future work are given in Section 5.

2. Compact high-order finite volume method on unstructured grids

2.1 General framework of high-order finite volume methods on unstuctured grids

This subsection describes the general framework of the high-order cell-centered finite volume method for compressible inviscid flows on unstructured grids. The governing equations are the Euler equations in conservative form, which can be expressed as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = 0, \tag{1}$$

where $\mathbf{U}$ is the conservative variabel vector and $\mathbf{F}$ is the inviscid flux tensor defined by

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho \mathbf{u} \\ \rho E \end{pmatrix}, \qquad \mathbf{F} = \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + p\mathbf{I} \\ (\rho E + p)\mathbf{u} \end{pmatrix}. \tag{2}$$

Here ρ is the density, $\mathbf{u}$ is the velocity , p is the pressure, and $\mathbf{I}$ is the identity tensor. E is the total energy defined by

$$E = \frac{1}{\gamma - 1}\frac{p}{\rho} + \frac{1}{2}\mathbf{u} \cdot \mathbf{u}, \tag{3}$$

where γ is the ratio of specific heat.

The computational domain Ω is decomposed into N non-overlapping control volumes, also called elements or cells, i.e. $\Omega = \bigcup_{i=1}^{N} \Omega_i$. Integrating the governing equation (1) over the cell Ω_i yields the following semi-discrete finite volume scheme:

$$\frac{d\overline{\mathbf{U}}_i}{dt} = -\frac{1}{\overline{\Omega}_i} \oint_{\partial \Omega_i} \mathbf{F} \cdot \mathbf{n} \, dA, \tag{4}$$

where $\overline{\mathbf{U}}_i = \frac{1}{\overline{\Omega}_i} \int_{\Omega_i} \mathbf{U}(\mathbf{x}, t) \, dV$ denotes the cell-average of the conservative variables, $\overline{\Omega}_i$ denotes the volume of Ω_i, $\partial\Omega_i$ denotes the boundary of Ω_i and $\mathbf{n}$ denotes the outward unit normal to $\partial\Omega_i$. The integral of flux in (4) is computed by using a Gauss quadrature, i.e.,

$$\oint_{\partial \Omega_i} \mathbf{F} \cdot \mathbf{n} dA \approx \sum_{f \in \partial \Omega_i} \sum_{g=1}^{N_g} \omega_g \mathbf{F}|_{\mathbf{x}=\mathbf{x}_{f,g}} \cdot \mathbf{n}_f A_f, \tag{5}$$

where $\mathbf{n}_f$ and A_f are the outward unit normal vector and area of cell interface f, respectively. Here, N_g denotes the number of quadrature points, while $\mathbf{x}_{f,g}$ and ω_g denote the position and weight of the g-th quadrature point on f, respectively. The flux function $\mathbf{F}|_{\mathbf{x}=\mathbf{x}_{f,g}} \cdot \mathbf{n}_f$ is computed by using the approximate Riemann solver of Roe [24]

$$\mathbf{F}|_{\mathbf{x}=\mathbf{x}_{f,g}} \cdot \mathbf{n}_f \approx \widetilde{\mathbf{F}}(\mathbf{U}_L, \mathbf{U}_R, \mathbf{n}_f) = \frac{1}{2}\left[\mathbf{F}(\mathbf{U_L}) + \mathbf{F}(\mathbf{U_R})\right] \cdot \mathbf{n}_f - \frac{1}{2}|\widetilde{A}_n|(\mathbf{U}_R - \mathbf{U}_L), \tag{6}$$

with an entropy-fix of Sanders et al. [25]. The absolute Jacobian $|\widetilde{A}_n|$ is evaluated at the Roe averages. The left and right states $\mathbf{U}_L$ and $\mathbf{U}_R$ at each Gaussian quadrature point $\mathbf{x}_{f,g}$ in (6) are obtained from the reconstruction polynomials of cell i and its face-neighboring cell j sharing f, respectively. Once the right-hand-side is computed, the semi-discrete FV scheme reduces to an ordinary differential equation (ODE) that can be integrated in time to update the cell-averages, which will be presented in Section 2.3.

2.2 Low-storge variational reconstruction

The issue of memory cost arises in variational reconstruction when large matrices need to be precomputed and stored. In this work, we proposed the low-storage variational reconstruction to resolve the issue of

memory cost. For clarity, the low-storage variational reconstruction is illustrated in two dimensions for a single scalar component $u(x, y)$ of $\mathbf{U}(x, y)$, as reconstruction is performed component by component. The compact reconstruction stencil $S_i = \{i, j_1, j_2, j_3\}$ of cell i is shown in Figure 1.

On each cell $\Omega_i \in \Omega$, the conservative variable $u(x, y)$ is approximated by a degree k polynomial

$$u_i(x, y) = \overline{u}_i + \sum_{l=1}^{N_c(k)} u_i^l \varphi_{l,i}(x, y), \tag{7}$$

where $\{u_i^l\}$ are the unknown basis coefficient and $N_c(k) = (k+1)(k+2)/2-1$ is the number of coefficients for two-dimensional case. $\{\varphi_{l,i}\}$ are the zero-mean basis functions, defined by

$$\begin{gathered} \varphi_{l,i} = \delta x_i^m \delta y_i^n - \overline{\delta x_i^m \delta y_i^n}, \\ \delta x_i = (x - x_i)/\Delta x_i,\ \delta y_i = (y - y_i)/\Delta y_i, \quad \overline{\delta x_i^m \delta y_i^n} = \frac{1}{\overline{\Omega}_i} \int_{\Omega_i} \delta x_i^m \delta y_i^n \, dxdy, \end{gathered} \tag{8}$$

where (x_i, y_i) denotes the centroid of Ω_i, m and n are powers of the corresponding basis functions. The characteristic length scales $(\Delta x_i, \Delta y_i)$ of Ω_i, which are introduced to prevent the condition number of the reconstruction matrix from increasing during grid refinement [26, 27], are defined following Luo et al. [28]

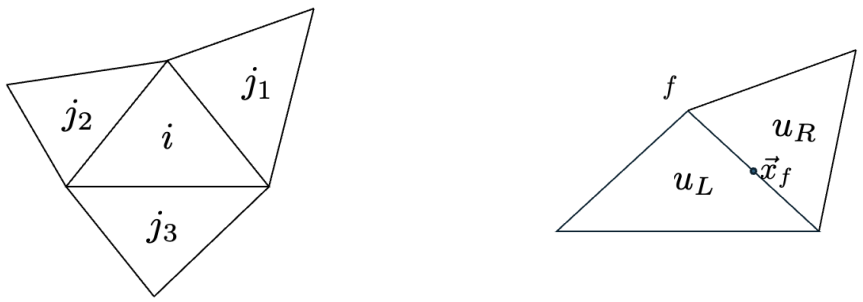
$$\Delta x_i = (x_{\max} - x_{\min})/2, \quad \Delta y_i = (y_{\max} - y_{\min})/2, \tag{9}$$

where $x_{\max}, x_{\min}, y_{\max}$ and $y_{\min}$ are the maximum and minimum coordinates of Ω_i in x- and y-direction.

In low-storage variational reconstruction, the coefficients of the reconstruction polynomials are obtained by solving a system of linear equations derived from the minimization of a global cost function. Different cost functions result in different reconstruction schemes. In this paper, an interfactial jump function on cell interface f is defined as

$$I_f = \sum_{n_x+n_y=0}^{k} \omega_{f,n_x n_y} \, (d_{LR})^{2(n_x+n_y)} \left(\frac{\partial^{n_x+n_y} u_L}{\partial x^{n_x} \, \partial y^{n_y}}\bigg|_{\vec{x}=\vec{x}_f} - \frac{\partial^{n_x+n_y} u_R}{\partial x^{n_x} \, \partial y^{n_y}}\bigg|_{\vec{x}=\vec{x}_f} \right)^2, \tag{10}$$

where $\vec{x}_f$ is the midpoint of interface f, L and R represent the left and right cells adjacent to interface f, and d_{LR} is the distance between the centroids of the two adjacent cells. The cell interface is shown in Figure 1. $\left\{\omega_{f,n_x n_y} | \omega_{f,n_x n_y} > 0\right\}_{n_x+n_y=0}^{k}$, with $\omega_{f,00} = 1$, are the weights of spatial derivatives jump



(a) Compact reconstruction stencil. (b) The triangular mesh element interface.

Figure 1: The triangular elements.

terms on interface f, which can be adjustable, and will be optimized using an ensemble Kalman inversion approach in Section 3. The interfactial jump function I_f measures the discontinuities of spatial derivatives of the reconstruction polynomials between adjacent cells only at the interface centers. The cost function

is the sum of interfacial jump functions over the whole computational domain, i.e.,

$$I = \sum_{f=1}^{N_f} I_f, \tag{11}$$

where N_f denotes the total number of cell interfaces in the computational domain. The linear equation system can be obtained by using the variational method, i.e

$$\frac{\partial I}{\partial u_i^l} = 0,\ l = 1, ..., N_c(k),\ i = 1, ..., N, \tag{12}$$

then, the reconstruction polynomial coefficients $u_i^l, l = 1, ..., N_c(k)$ can be derived by solving the system. Since the unknown coefficients u_i^l appear only in the interfacial jump functions associated with the faces of Ω_i, as observed in (10), they can be determined using only the information of the face-neighboring cells within the compact stencil S_i shown in Figure 1. By substituting (7) into (10)-(12), we obtain the following linear equation system

$$\mathbf{A}_i \mathbf{u}_i = \sum_{j \in S_i, j \neq i} \mathbf{B}_i^j \mathbf{u}_j + \mathbf{b}_i,\ i = 1, ..., N, \tag{13}$$

where the elements of matrices $\mathbf{A}_i, \mathbf{B}_i^j \in \mathbb{R}^{N_c(k) \times N_c(k)}$ and vectors $\mathbf{u}_i, \mathbf{u}_j, \mathbf{b}_i \in \mathbb{R}^{N_b(k)}$ are

$$\begin{aligned}
\mathbf{A}_i[l,q] &= \sum_{j \in S_i,\ j \neq i} \sum_{n_x+n_y=0}^{k} \omega_{f,n_x n_y} d_{ij}^{2n_x+2n_y} \frac{\partial^{n_x+n_y} \varphi_{l,i}(\vec{x})}{\partial x^{n_x} \partial y^{n_y}} \frac{\partial^{n_x+n_y} \varphi_{q,i}(\vec{x})}{\partial x^{n_x} \partial y^{n_y}}, \\
\mathbf{B}_i^j[l,q] &= \sum_{n_x+n_y=0}^{k} \omega_{f,n_x n_y} d_{ij}^{2n_x+2n_y} \frac{\partial^{n_x+n_y} \varphi_{l,i}(\vec{x})}{\partial x^{n_x} \partial y^{n_y}} \frac{\partial^{n_x+n_y} \varphi_{q,j}(\vec{x})}{\partial x^{n_x} \partial y^{n_y}}, \\
\mathbf{b}_i[l] &= \sum_{j \in S_i,\ j \neq i} \omega_{f,00}\, \varphi_{l,i}(\vec{x})\, (\bar{u}_j - \bar{u}_i), \\
\mathbf{u}_i[l] &= u_i^l, \quad \mathbf{u}_j[l] = u_j^l.
\end{aligned} \tag{14}$$

It is shown in (13) that the low-storage variational reconstruction is implicit. By assembling the linear equations (13) of all cells, we can obtain a global linear equation system

$$\mathbf{A}\mathbf{u} = \mathbf{b}, \tag{15}$$

where

$$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}, \mathbf{D} = \{\mathbf{A}_i\}, \mathbf{L} = \left\{\mathbf{B}_i^j, j < i\right\}, \mathbf{U} = \left\{\mathbf{B}_i^j, j > i\right\}, \mathbf{u} = \{\mathbf{u}_i\}, \mathbf{b} = \{\mathbf{b}_i\}. \tag{16}$$

It has been proven that the matrix $\boldsymbol{A}$ is symmetric and positive definite in [4], which guatantees the existence and uniqueness of the solution of the linear equation system (15). Then, the linear equation system (15) can be solved iteratively using block Gauss-Seidel method or block SOR method as discussed in [4]. In this paper, the block SOR method

$$\mathbf{u}_i^{(s+1)} = (1-w)\, \mathbf{u}_i^{(s)} + w \left[\sum_{j \in S_i,\ j<i} \mathbf{A}_i^{-1} \mathbf{B}_i^j \mathbf{u}_j^{(s+1)} + \sum_{j \in S_i,\ j>i} \mathbf{A}_i^{-1} \mathbf{B}_i^j \mathbf{u}_j^{(s)} + \mathbf{A}_i^{-1} \mathbf{b}_i \right], \quad i = 1, \ldots, N, \tag{17}$$

is used to solve (15) iteratively, where s denote the iteration step and w is the relaxation factor.

In the low-storage reconstruction procedure, determining the reconstruction polynomial within a given cell require only the evaluation of the spatial derivatives at the interface centers, as shown in (10), the

linear equation system can further simplify to

$$\mathbf{A_i u}_i = \sum_{j \in S_i, j \neq i} \mathbf{B}_i^j \mathbf{u}_j + \mathbf{b}_i = \mathbf{R}_i, \quad i = 1, ..., N \tag{18}$$

where the elements of the vectors $\mathbf{R}_i \in \mathbb{R}^{N_c(k)}$ are

$$\mathbf{R}_i[l] = \sum_{j \in S_i,\, j \neq i} \sum_{n_x+n_y=0}^{k} \omega_{f,n_x n_y} d_{ij}^{2n_x+2n_y} \frac{\partial^{n_x+n_y} u_j(\vec{x})}{\partial x^{n_x} \partial y^{n_y}} \frac{\partial^{n_x+n_y} \varphi_{l,i}(\vec{x})}{\partial x^{n_x} \partial y^{n_y}} + \sum_{j \in S_i,\, j \neq i} \varphi_{l,i}(\vec{x})(\overline{u}_j - \overline{u}_i). \tag{19}$$

By leveraging the sparsity of the basis function derivatives, i.e., only consider the non-zero terms, the vectors $\mathbf{R}_i$ can be computed efficiently during each iteration step. The cost function of the traditional variational reconstruction is based on the IJI [4], therefore, it is often necessary to store matrices $\mathbf{A}_i$ and $\mathbf{B}_i^j$ in advance to ensure high efficiency. In the low-storge variational reconstruction, only the matric $\mathbf{A}_i$ need to be store in advance, resulting in low memory usage. Taking a three-dimensional hexahedron as an example, it is necessary to store seven matrices for the traditional variational reconstruction. Theoretically, for a hexahedron, storage savings up to 85% are possible for the low-storage variational reconstruction.

The weights $\left\{\omega_{f,n_x n_y} | \omega_{f,n_x n_y} > 0\right\}_{n_x+n_y=0}^{k}$ also called derivative weights in [18, 19], are the free parameters that affect the efficiency of the high-order low-storage VFV scheme. Recently, Huang et al. [18] found a set of optimal derivative weights

$$\begin{aligned} &\omega_{f,10} = \omega_{f,01} = 0.5295^2, \\ &\omega_{f,20} = \omega_{f,11} = \omega_{f,02} = 0.2117^2, \\ &\omega_{f,30} = \omega_{f,21} = \omega_{f,12} = \omega_{f,03} = 0.2117^2, \end{aligned} \tag{20}$$

for cubic variational reconstruction through a spectral property analysis in one-dimension and extensted to multi-dimension. Zhou et al. [19] used a machine learning optimization technique to predict the optimal values of the derivative weights at cell interfaces on unstructured grids, with local geometry given as input. The two methods for optimizing the derivative weights did not take into account the rotational invariance of reconstruction. In this paper, to maintain the computational rotational invariance of reconstruction, the derivative weights for the cubic low-storage variational reconstruction should satisfy

$$\begin{aligned} &\omega_{f,10} = \omega_{f,01} = \omega_{f,1}, \\ &\omega_{f,20} = \omega_{f,02} = \frac{1}{4}\omega_{f,2}, \quad \omega_{f,11} = \frac{1}{2}\omega_{f,2}, \\ &\omega_{f,30} = \omega_{f,03} = \frac{1}{36}\omega_{f,3}, \quad \omega_{f,21} = \omega_{f,12} = \frac{1}{12}\omega_{f,3}, \end{aligned} \tag{21}$$

where the weights $\{\omega_{p=nx+ny}\}_{p=1}^{k}$ are free parameters.

In [19], it is observed that, on iotropic triangular meshes, whether the derivative weights are set to the same value for every edge or vary from edge to edge, the calculation results do not differ significantly. Therefore, we will ignore the influence of the local geometric shape and apply uniform weights $\omega_p (p = 1, 2, 3)$ across the computional domain. The weights can easily be set as $\omega_p = 1 (p = 1, 2, 3)$, in which such a choice has good performance in [20].

To suppress spurious oscillations when reconstructing a solution with discontinuities, a WBAP limiter [29, 30] is employed in this paper. Moreover, a problem-independent troubled-cell indicator [31] is introduced to improve computational efficiency by restricting the application of the limiter to troubled cells.

2.3 Reconstruction and time integration coupled iteration procedure

The reconstruction polynomial defined by the linear equation system (15) is solved iteratively, and will be very expensive if the iteration needs to reach convergence at each single time step. This computational efficiency issue has been resolved through the implementation of a coupled iteration procedure involving a reconstruction and implicit dual-time stepping method in [16]. In the coupled iteration procedure, the reconstruction iteration is performed only once at each pseudo time step. This makes the reconstruction and time integration procedures converge synchronously. The use of the coupled iteration prevents additional cost from the implicit low-storage variational reconstruction, thus ensuring the high computational efficiency of the low-storage variational finite volume method.

In this work, the LU-SGS [32, 33] scheme is used for time integration in the steady-state simulation. The fourth-order accurate singly-diagonally implicit Runge–Kutta (SDIRK) scheme [34] is used for time integration in the unsteady flow simulation. More details can be found in [16].

3. The optimization based on ensemble Kalman inversion for low-storge variational reconstruction

As pointed out in Section 2.2, the weights ω_p are tunable parameters that govern the performance of the high-order low-storage VFV scheme. Different weights actually result in different low-storage variational reconstruction schemes. The choices of the weights do not influence the k-exactness of the reconstruction, and consequently, the optimization will not affect the formal order of accuracy of the low-storage VFV scheme.

In this section, we develop a posterior optimization framework employing EKI for the low-storage VFV scheme, in which the scheme will run up on both high-quality and low-quality grids to obtain the weights corresponding to different geometric partitionings. The structure of the optimization for cubic ($k = 3$) low-storage variational reconstruction will be presented in the remainder of this section. The relaxation factor in block SOR method is set $w = 1$ in this Section.

3.1 Optimization problem

The optimization of numerical schemes based on Fourier analysis on structured grids, which can be regarded as a priori optimization method, often considers a single Fourier mode as the initial condition for the linear advection equation. The modified wavenumber can be calculated and used to analyse the dispersion and dissipation errors [35, 36]. The optimal parameters of the numerical scheme can be obtained by minimizing a cost function that is a weighted sum of the dispersion and dissipation errors for a given set of wavenumbers [18, 35]. Zhou et al. [19] borrowed the idea and trained an ANN by minimizing solution errors of a linear advection equation with sine waves as the initial conditions to predict the derivative weights of variational reconstruction on triangular grids.

In this paper, we use a wave packet composed by a series of weighted sine waves as the initial condition for the linear advection equation

$$\frac{\partial u}{\partial t} + \mathbf{a} \cdot \frac{\partial u}{\partial \mathbf{x}} = 0, \tag{22}$$

where $\mathbf{a} = (a\cos\psi, a\sin\psi)$ is the wave speed. Since the solution errors for waves of different wavenumbers can differ by orders of magnitude, we adopt the approach of Zhou et al. [19], whereby the different waves are normalised using the reference errors obtained from the reference low-storage variational finite volume scheme designed with the reference weights [20], i.e.,

$$\omega_1 = \omega_2 = \omega_3 = 1. \tag{23}$$

Then, we can design the initial condition as

$$u(x,y) = \sum_{k_m} \frac{e^{-\mu k_m h}}{\sum_{k_j} e^{-\mu k_j h}} \frac{\sin\left(k_m \left(x\cos\theta + y\sin\theta\right)\right)}{\mathrm{err}_{L2,\mathrm{ref}}^{km}}, \tag{24}$$

where $\mathrm{err}_{L2,\mathrm{ref}}^{km}$ denotes the L_2 error of the numerical solution at time T, computed by the reference low-storage VFV scheme (23) for the equation (22) with the sine wave $\sin\left(k_m \left(x\cos\theta + y\sin\theta\right)\right)$ as the initial condition. Here, θ denotes the orientation of the wave. The waves are also modified by the $e^{-\mu k_m h}$ to control the relative importance of the solution errors at different wavenumbers, where h is the characteristic length of the computional domain, and μ is a ajustabal parameter. In this paper, we set $a = 1$, $\mu = 1$ and $\psi = \theta$.

3.2 Optimization target

Since the weights affect the accuracy, efficiency, and robustness of the low-storage variational finite volume scheme, during the optimization process, two separate meshes representing high- and low-quality, respectively, are used for the subdivision of different geometric shapes within the computational domain. The low-storage VFV scheme, which is designed with weights that need to be optimised, as presented in Section 2.2-2.3, is adapted to solve the equation (22) with the given initial condition (24). During the optimization process, we intend to use the solution errors calculated based on the reference weights (23) to evaluate the results of the optimization. Specifically, after solving the equation (22) using this numerical scheme up to time T, the output values for each cell within the meshes are obtained as

$$G_i(\boldsymbol{\omega}) = \frac{\left(u_i^h - u_i^e\right) T_{\mathrm{cpu}}}{\mathrm{err}_{L2,\mathrm{ref}} \sqrt{\overline{\Omega}}\, T_{\mathrm{cpu}}^{\mathrm{ref}}}, \quad i = 1, ..., N, \tag{25}$$

where u_i^h and u_i^e are the numerical solution and exact solution for the cell i, respectively. T_{cpu} and $T_{\mathrm{cpu}}^{\mathrm{ref}}$ are the CPU times required for the low-storage VFV schemes designed with the weights obtained during the iteration process and the reference one (23), respectively, to solve the equation (22) up to time T. $\mathrm{err}_{L2,\mathrm{ref}}$ is the L_2 solution error calculated using the reference low-storage VFV scheme for the equation (22) with initial condition (24). $\overline{\Omega}$ is the total volume of the computional domain.

The optimization objective is

$$\begin{aligned} &J_h(\boldsymbol{\omega}) = ||G(\boldsymbol{\omega})||^2_{\boldsymbol{\Gamma}_h^{-1}} = \frac{(\sum_i^N (u_i^h - u_i^e)^2\, \overline{\Omega}_i) T_{\mathrm{cpu}}^2}{\mathrm{err}_{L2,\mathrm{ref}}^2\, \overline{\Omega} (T_{\mathrm{cpu}}^{\mathrm{ref}})^2} = \frac{\mathrm{err}_{L2}^2 T_{\mathrm{cpu}}^2}{\mathrm{err}_{L2,\mathrm{ref}}^2 (T_{\mathrm{cpu}}^{\mathrm{ref}})^2}, \quad \boldsymbol{\omega}* = \mathrm{argmin} J_h(\boldsymbol{\omega}), \\ &G(\boldsymbol{\omega}) = [G_1(\boldsymbol{\omega}), ..., G_N(\boldsymbol{\omega})]^T, \end{aligned} \tag{26}$$

where $\boldsymbol{\Gamma}_h = diag\{\frac{1}{\overline{\Omega}_1}, ..., \frac{1}{\overline{\Omega}_N}\}$. It can be observed that the objective function can also be written as

$$J_h(\boldsymbol{\omega}) = \frac{\eta_{\mathrm{ref}}^2}{\eta_{\boldsymbol{\omega}}^2}, \quad \eta = \frac{1}{T_{\mathrm{cpu}} \cdot \mathrm{err}_{L_2}}, \tag{27}$$

where a lower value indicates better performance. η is the computional efficency. Larger η means the same accuracy at lower cost or higher accuracy at the same cost [16].

3.3 EKI Optimization procedures

From (26), we can see that the optimization objective is essentially a least-squares problem, which makes it well suited for the ensemble Kalman inversion method. One classical application of EKI is solving inverse problems of the form

$$y = G(\boldsymbol{\omega}) + \eta, \tag{28}$$

where y denotes the observed data, $G(\boldsymbol{\omega})$ is the predicted observational data, $\boldsymbol{\omega}$ denotes the unknown parameter, and η is Gaussian observational noise with mean zero and covariance $\boldsymbol{\Gamma}$.

To make our scheme more robust, as mentioned in Section 3.2, we use the numerical scheme up to time T on low- and high- quality meshes, respectively, and use the output values (25) to optimize the weights. This would result in a high computational cost for training a neural network. Furthermore, we also take the CPU time into account here, which is computed by using the operating system's timing functions and is not differentiable with respect to the weights. Ensemble Kalman inversion is an ensemble-based derivative-free optimization method that naturally supports parallel computing. Instead of iteratively updating a single parameter vector estimate, EKI evolves an ensemble of parameter particles. At each iteration, the ensemble is propagated through the forward model, and the empirical cross-covariance between the parameter ensemble and the associated outputs is used to construct a Kalman-type update.

In this study, the initial parameter ensemble is generated based on a given reference weight vector $\boldsymbol{\omega}_{\text{ref}}$ (23). Since $\boldsymbol{\omega} > 0$, we reparameterize $\boldsymbol{\omega}$ as $\boldsymbol{\omega} = \exp(\boldsymbol{\zeta})$ and optimize with respect to $\boldsymbol{\zeta}$ instead. Specifically, a Gaussian distribution is constructed with $\boldsymbol{\zeta}_{\text{ref}} = \ln(\boldsymbol{\omega}_{\text{ref}})$ as the mean and $\boldsymbol{\sigma}_{\text{psd}}$ as the standard deviation. The initial ensemble of parameter particles is then sampled as

$$\boldsymbol{\zeta}_0^{(j)} \sim \mathcal{N}\left(\boldsymbol{\zeta}_{\text{ref}}, \text{diag}(\boldsymbol{\sigma}_{\text{psd}}^2)\right), \qquad j = 1, \ldots, J, \tag{29}$$

where J denotes the ensemble size. The ensemble of parameter at iteration n is denoted by $\left\{\boldsymbol{\zeta}_n^{(j)}\right\}_{j=1}^{J}$.

For each ensemble particle $\boldsymbol{\zeta}_n^{(j)}$ at iteration n, the low-storage VFV scheme is specified by $\boldsymbol{\zeta}_n^{(j)}$. The predicted observations are then obtained by solving the governing equation (22) with this scheme, i.e.,

$$\mathbf{g}_n^{(j)} = G(\exp(\boldsymbol{\zeta}_n^{(j)})), \qquad j = 1, \ldots, J. \tag{30}$$

The predicted observations $\{\mathbf{g}_n^{(j)}\}_{j=1}^{J}$ are compared with the target observational data $\mathbf{y}$, here, we set $\boldsymbol{y} = 0$. The parameter ensemble is subsequently updated through a Kalman-type correction informed by the ensemble statistics. The update formula can be written as

$$\boldsymbol{\zeta}_{n+1}^{(j)} = \boldsymbol{\zeta}_n^{(j)} + \mathbf{C}_n^{\zeta g}\left(\mathbf{C}_n^{gg} + \boldsymbol{\Gamma}/\alpha_n\right)^{-1}\left(\mathbf{y} - \mathbf{g}_n^{(j)}\right), \qquad j = 1, \ldots, J, \tag{31}$$

where $\boldsymbol{\Gamma}$ denotes the observational error covariance matrix, $\mathbf{C}_n^{\zeta g}$ is the empirical cross-covariance between the parameter ensemble and the predicted observation ensemble, and $\mathbf{C}_n^{gg}$ is the empirical covariance of the predicted observations. The covariance matrices are defined as

$$\begin{aligned}
\mathbf{C}_n^{\zeta g} &= \frac{1}{J-1}\sum_{j=1}^{J}\left(\boldsymbol{\zeta}_n^{(j)} - \bar{\boldsymbol{\zeta}}_n\right)\left(\mathbf{g}_n^{(j)} - \bar{\mathbf{g}}_n\right)^T, \qquad \bar{\boldsymbol{\zeta}}_n = \frac{1}{J}\sum_{j=1}^{J}\boldsymbol{\zeta}_n^{(j)}, \\
\mathbf{C}_n^{gg} &= \frac{1}{J-1}\sum_{j=1}^{J}\left(\mathbf{g}_n^{(j)} - \bar{\mathbf{g}}_n\right)\left(\mathbf{g}_n^{(j)} - \quad \bar{\mathbf{g}}_n\right)^T, \qquad \bar{\mathbf{g}}_n = \frac{1}{J}\sum_{j=1}^{J}\mathbf{g}_n^{(j)}.
\end{aligned} \tag{32}$$

During the iterative optimization procedure, an adaptive damping factor α_n is introduced to control the update step size [37]. The damping factor is initialized as $\alpha_0 = 1$. At each iteration, a candidate update is computed using the current value of α_n, and the objective function is evaluated after the update. If the candidate update leads to a decrease in the objective function (26), the update is accepted and the damping factor is slightly increased according to

$$\alpha_{n+1} = \min\{1,\ 1.1\alpha_n\}. \tag{33}$$

The upper bound of 1 prevents the damping factor from becoming excessively large. If the candidate

update does not decrease the objective function, the update is rejected and the damping factor is reduced as

$$\alpha_{n+1} = 0.5\alpha_n. \tag{34}$$

The update is then recomputed using the reduced damping factor. This adaptive strategy allows the algorithm to take larger steps when the optimization is stable, while automatically reducing the step size when the objective function fails to improve. This procedure is repeated until a prescribed stopping criterion is satisfied, i.e., the maximum number of iterations $N_{\text{max\_iteration}}$ is reaching, the reduction in the objective function (26) no longer decreases within a given tolerance N_{tol} or $\alpha < \alpha_{min}$. Finally, the ensemble mean

$$\bar{\boldsymbol{\zeta}}_{N_{\text{ite}}} = \frac{1}{J}\sum_{j=1}^{J} \boldsymbol{\zeta}_{N_{\text{ite}}^{(j)}}, \tag{35}$$

can be used to obtain the optimized weights $\boldsymbol{\omega}_{\text{opt}} = \exp(\bar{\boldsymbol{\zeta}}_{N_{\text{ite}}})$, where N_{ite} is the total number of iterations.

3.4 Data set generation

This subsection ptesents the generation of data sets for the optimization based on EKI. As pointed out in Section 3.2, in order to make our scheme more robust, both high-quality and low-quality meshes that can be discretized using different element types, such as an all-triangular mesh or an all-quadrilateral mesh, will be used for the fixed computational domain. The computational domains are generated from a background mesh over domain $[0, 10] \times [0, 10]$, with a characteristic element size $h = 0.3125$ and $h = 0.5$ for quadrilateral and triangular meshes, respectively, as shown in Figure 2 and Figure 3.

(a) high-quality (b) low-quality

Figure 2: Quadrilateral meshes.

(a) high-quality (b) low-quality

Figure 3: Triangular meshes.

For a given discretization type, the low-quality and high-quality meshes use the same initial condition (24), which varies across different discretization types. $\text{err}_{L2,\text{ref}}^{k_m}$ in (24) are all calculated on the high-quality mesh for both triangular and quadrilateral discretizations.

The initial condition (24) is controlled by the wavenumber k_m and orientation θ. The limitations of the wavenumber and orientation are

$$0 < k_m h \leq \pi, \quad \theta \in [0, 2\pi). \tag{36}$$

In this paper, the parameters domain for initial condition sampling are set as $k_m \in [0.1\pi/h, 0.7\pi/h]$ and $\theta = \frac{\pi}{4}$. Twenty k_m points are obtained using the uniform distribution. Given the k_m and θ, the $\mathrm{err}^{k_m}_{L2,\mathrm{ref}}$, $\mathrm{err}_{L2,\mathrm{ref}}$ and u_i^e can be obtained in advance at time T. In this paper, the simulation duration is set to $T = 10$ to ensure that all waves can propagate across the entire computational domain within this time interval. The corresponding physical time steps are $\Delta t = 1/32$ and $\Delta t = 1/20$ for the quadrilateral and triangular meshes, respectively. The local pseudo time step is computed by

$$\Delta t_i = \mathrm{CFL}\frac{\overline{\Omega}_i}{\sum_{f\in\partial\Omega_i}\lambda_f A_f}, \quad \lambda_f = (\cos\theta, \sin\theta)\cdot \boldsymbol{n}_f. \tag{37}$$

The CFL number for local pseudo time step is 40 and the convergence criterion for inner iteration is that the residual decreases by 9 orders of magnitude.

In the EKI optimized process, we set the ensemble size $J = 32$, the maximum number of iterations $N_{\mathrm{max\_iteration}} = 200$, the toleracne $N_{\mathrm{tol}} = 10$, and the minimum damping factor $\alpha_{\min} = 1.0 \times 10^{-4}$. As both high-quality and low-quality meshes are used for the computational domain, we set $\kappa = 1.5$ here to emphasise the relative importance of the two grids for both the triangular and quadrilateral discretizations. In this work, EKI is mainly employed as a derivative-free iterative optimization method, rather than as a strict Bayesian posterior sampling method. Therefore, the observational error covariance matrix $\boldsymbol{\Gamma}$ plays a dual role that it weights the data-misfit term and also regularizes the update step. For simplicity, an isotropic form is adopted

$$\begin{aligned}
&\boldsymbol{\Gamma} = \{\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2\}, \quad \sigma_1 = \sigma = 0.1, \quad \sigma_2 = \kappa\sigma_1, \\
&\boldsymbol{\Gamma}_1 = \sigma_1^2 \mathrm{diag}\{1/\overline{\Omega}_1^{\mathrm{high}}, ..., 1/\overline{\Omega}_{N_{\mathrm{high}}}^{\mathrm{high}}\},\ \boldsymbol{\Gamma}_2 = \sigma_2^2 \mathrm{diag}\{1/\overline{\Omega}_1^{\mathrm{low}}, ..., 1/\overline{\Omega}_{N_{\mathrm{low}}}^{\mathrm{low}}\},
\end{aligned} \tag{38}$$

A larger value of σ leads to a more conservative parameter update, whereas a smaller value results in a stronger correction toward the target observation.

3.5 Optimization Results

Since the optimization objective is (27) , we use the metric E

$$E = \frac{\kappa}{1+\kappa}\frac{\eta_{\mathrm{ref}}^{\mathrm{high}}}{\eta_{\omega}^{\mathrm{high}}} + \frac{1}{1+\kappa}\frac{\eta_{\mathrm{ref}}^{\mathrm{low}}}{\eta_{\omega}^{\mathrm{low}}} \tag{39}$$

to measure the optimization performance relative to the reference scheme, where $\eta_{\mathrm{ref}}^{\mathrm{high}}$, $\eta_{\mathrm{ref}}^{\mathrm{low}}$ ($\eta_{\omega}^{\mathrm{high}}$, $\eta_{\omega}^{\mathrm{low}}$) are the effiency calculated by the reference scheme (the optimized scheme) on high- and low-quality meshes, respectivelly. When the weight $\boldsymbol{\omega}$ is set to the reference one, it is easy to see that the value of E is 1. For the quadrilateral discretization, the optimization results are

$$\begin{aligned}
&\omega_1 = 0.222604, \quad \omega_2 = 0.253230, \quad \omega_3 = 0.595188, \\
&E = 0.4087.
\end{aligned} \tag{40}$$

For the triangular discretization, the optimization results are

$$\begin{aligned}
&\omega_1 = 0.348449, \quad \omega_2 = 0.497156, \quad \omega_3 = 0.675714, \\
&E = 0.5984.
\end{aligned} \tag{41}$$

Numerical results in Section 4 will show that the optimization of the weights by EKI results in efficiency improvement by achieving the same accuracy at a lower cost.

4. Numerical results

This section presents the numerical results of the EKI optimized fourth-order low-storage VFV scheme for several two-dimensional benchmark test cases. The isentropic vortex, double mach, subsonic flow past a NACA0012 airfoil and Rayleigh-Taylor instability problems are used to test the accuracy, efficiency and resolution of the optimized scheme. The block SOR method uses $w = 1.3$ in this Section. The results of the following three fourth-order low-storage VFV schemes are also presented for comparison:
(1) **Non-optimized**: the derivative weights are set as $\omega_1 = 1$, $\omega_2 = 4$, $\omega_3 = 36$, which matain the rotational invariance of reconstruction, but the derivative weights are non-optimized.
(2) **Reference**: the reference weights $\omega_1 = \omega_2 = \omega_3 = 1$, which performs well in [20].
(3) **Huang et al. [18]**: derivative weights (20) for the traditional variational reconstruction, which are obtained via Fourier analysis in one dimension.

4.1 Isentropic vortex

The isentropic vortex transport problem [38] is used to test the accuracy of the low-storage VFV schemes and storage advantages over the traditional VFV scheme. The mean flow is $(\rho, u, v, p) = (1, 1, 1, 1)$. An isentropic vertex is added to the mean flow with the following perturbations

$$\begin{aligned}(\delta u, \delta v) &= \frac{\chi}{2\pi} e^{0.5(1-r^2)}(-\bar{y}, \bar{x}),\\ \delta T &= -\frac{(\gamma-1)}{\chi^2} 8\gamma\pi^2 e^{1-r^2},\\ \delta(S = p/\rho^\gamma) &= 0,\end{aligned} \tag{42}$$

where $(x, y) = (x - 5, y - 5)$, $r^2 = x^2 + y^2$ and the vertex strength $\chi = 5$. Periodic boundary conditions are imposed on all boundaries. Simulations are performed up to $T = 4.0$, using the fourth-order SDIRK method [34]. Successively refined high-quality and low-quality grids are used in the simulations, of which the coarsest meshes for quadrilateral and triangular discretizations are shown in Figure 4 and 5 .

(a) high-quality (b) low-quality

Figure 4: Quadrilateral meshes for the isentropic vertex problem with size $h = 0.625$.

(a) high-quality (b) low-quality

Figure 5: Triangular meshes for the isentropic vertex problem with size $h = 1$.

The grid sizes are $h = 5/8$ to $h = 5/128$ for quadrilateral discretization, and the corresponding physical time steps are $\Delta t = 1/16$ to $\Delta t = 1/256$. The grid sizes are $h = 1$ to $h = 1/16$ for triangular discretization, and the corresponding physical time steps are $\Delta t = 1/10$ to $\Delta t = 1/160$. The CFL number for local pseudo time step is 40. The convergence criterion for inner iteration is that the density residual decreases by 8 orders of magnitude for triangular discretization and the high-quality quadrilateral discretization, and the density residual decreases by 7 orders of magnitude is used for the low-quality quadrilateral discretization.

The L_1, L_2 and L_∞ density errors and rates of convergence on high-quality and low-quality quadrilateral meshes are listed in Table 1 and 2. The results in Table 1 and 2 show that all the four low-storage VFV schemes achieve the theoretical fourth-order accuracy. The error-cost plots for comparisons with

Table 1: Accuracy test results for the isentropic vortex problem on high-quality quadrilateral meshes.

Schemes	Grid size	L_1 error	Order	L_2 error	Order	L_∞ error	Order
present	5/8	2.260E−3	–	4.735E−3	–	3.804E−2	–
	5/16	3.907E−4	2.53	1.011E−3	2.23	1.410E−2	1.43
	5/32	1.037E−5	5.24	2.809E−5	5.17	3.820E−4	5.21
	5/64	2.311E−7	5.49	5.013E−7	5.81	4.373E−6	6.45
	5/128	1.045E−8	4.47	2.020E−8	4.63	1.549E−7	4.82
referecne	5/8	3.715E−3	–	9.194E−3	–	7.037E−2	–
	5/16	7.829E−4	2.25	2.097E−3	2.13	3.021E−2	1.22
	5/32	1.936E−5	5.34	5.258E−5	5.32	6.644E−4	5.51
	5/64	2.295E−7	6.40	5.789E−7	6.51	5.978E−6	6.80
	5/128	8.597E−9	4.74	1.582E−8	5.19	1.237E−7	5.59
Non-optimized	5/8	1.200E−2	–	3.092E−2	–	2.605E−1	–
	5/16	1.961E−3	2.61	4.368E−3	2.82	3.556E−2	2.87
	5/32	2.187E−4	3.17	5.339E−4	3.03	6.793E−3	2.39
	5/64	5.147E−6	5.41	1.442E−5	5.21	1.712E−4	5.31
	5/128	5.052E−8	6.67	1.444E−7	6.64	1.510E−6	6.82
Huang et al. [18]	5/8	2.538E−3	–	5.488E−3	–	4.365E−2	–
	5/16	4.460E−4	2.51	1.166E−3	2.24	1.615E−2	1.43
	5/32	1.291E−5	5.11	3.379E−5	5.11	4.214E−4	5.26
	5/64	2.907E−7	5.47	6.605E−7	5.68	6.050E−6	6.12
	5/128	1.235E−8	4.56	2.541E−8	4.70	2.041E−7	4.89

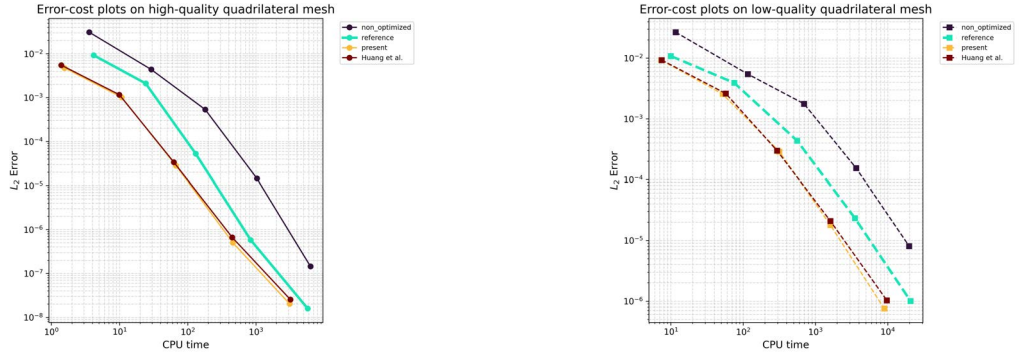
the aforementioned three low-storage VFV schemes on both high-quality and low-quality quadrilateral meshes are shown in 6. In Figure 6, we can notice that, the optimized low-storage VFV scheme is much more efficient than the reference and non-optimized low-storage VFV schemes. On low-quality grids, it also outperforms the reference scheme.

The L_1, L_2 and L_∞ density errors and rates of convergence on high-quality and low-quality quadrilateral meshes are listed in Table 3 and 4. The results in Table 3 and 4 show that the low-storage VFV schemes designed by the optimized, reference and non-optimized weights achieve the theoretical fourth-order accuracy. The computation by the low-storage VFV scheme with the weights of Huang et al. [18] on the low-quality mesh with $h = 1/4$ produced NaN values during the simulation. Hence, the corresponding errors and convergence orders are not reported, and only the available results up to $h = 1/2$ are included, which further indicates the necessity of optimizing the weights for the proposed scheme.

The error-cost plots for comparisons with the aforementioned low-storage VFV schemes on both high-quality and low-quality quadrilateral meshes are shown in Figure 7. Due to the occurrence of the NaN issue mentioned above in the weights of Huang et al. [18], this case is excluded from Figure 7. In Figure 7, we can notice that, the optimized low-storage VFV scheme is much more efficient than the reference and non-optimized low-storage VFV schemes.

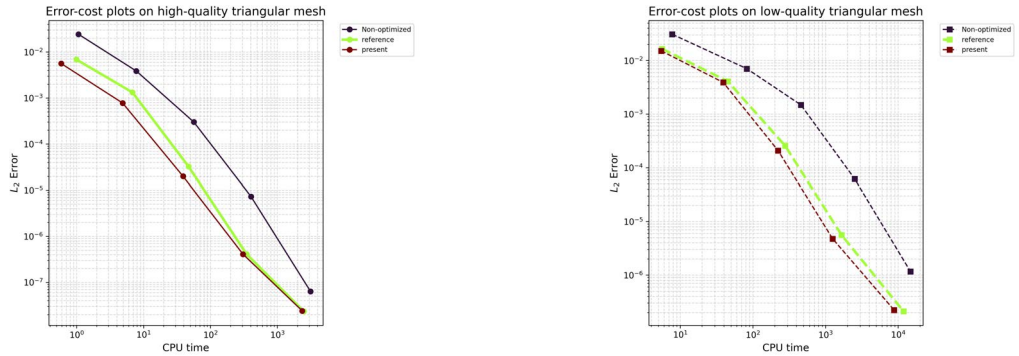
Table 2: Accuracy test results for the isentropic vortex problem on low-quality quadrilateral meshes.

Schemes	Grid size	L_1 error	Order	L_2 error	Order	L_∞ error	Order
present	5/8	6.888E−3	–	9.233E−3	–	4.466E−2	–
	5/16	1.392E−3	3.14	2.559E−3	2.52	3.229E−2	0.64
	5/32	1.647E−4	3.44	2.884E−4	3.52	3.019E−3	3.82
	5/64	9.857E−6	4.26	1.796E−5	4.20	1.633E−4	4.42
	5/128	3.845E−7	4.79	7.537E−7	4.68	8.239E−6	4.41
referecne	5/8	6.281E−3	–	1.081E−2	–	7.373E−2	–
	5/16	1.978E−3	2.27	3.910E−3	2.00	4.376E−2	1.03
	5/32	2.269E−4	3.49	4.335E−4	3.54	5.304E−3	3.40
	5/64	1.320E−5	4.31	2.321E−5	4.43	2.134E−4	4.86
	5/128	5.391E−7	4.72	1.007E−6	4.63	9.145E−6	4.65
Non-optimized	5/8	1.094E−2	–	2.673E−2	–	1.859E−1	–
	5/16	2.906E−3	2.61	5.430E−3	3.14	4.494E−2	2.79
	5/32	8.327E−4	2.01	1.768E−3	1.81	2.609E−2	0.88
	5/64	8.773E−5	3.41	1.539E−4	3.70	1.450E−3	4.38
	5/128	4.559E−6	4.36	8.069E−6	4.35	8.574E−5	4.17
Huang et al. [18]	5/8	6.953E−3	–	9.276E−3	–	4.370E−2	–
	5/16	1.430E−3	3.11	2.607E−3	2.50	3.308E−2	0.55
	5/32	1.705E−4	3.42	2.997E−4	3.48	3.511E−3	3.61
	5/64	1.176E−5	4.05	2.086E−5	4.03	1.998E−4	4.34
	5/128	5.290E−7	4.58	1.031E−6	4.44	1.296E−5	4.04



(a) high-quality (b) low-quality

Figure 6: Error-cost plots for the isentropic vortex problem on quadrilateral meshes.



(a) high-quality (b) low-quality

Figure 7: Error-cost plots for the isentropic vortex problem on triangular meshes.

Table 3: Accuracy test results for the isentropic vortex problem on high-quality triangular meshes

Schemes	Grid size	L_1 error	Order	L_2 error	Order	L_∞ error	Order
present	1	2.749E−3	–	5.603E−3	–	3.385E−2	–
	1/2	3.103E−4	3.04	7.769E−4	2.75	1.054E−2	1.63
	1/4	7.914E−6	5.75	1.988E−5	5.75	3.224E−4	5.47
	1/8	2.313E−7	4.90	4.077E−7	5.39	6.312E−6	5.45
	1/16	1.295E−8	4.22	2.399E−8	4.15	3.893E−7	4.08
reference	1	2.988E−3	–	6.858E−3	–	4.554E−2	–
	1/2	4.968E−4	2.50	1.329E−3	2.29	1.945E−2	1.19
	1/4	1.221E−5	5.81	3.258E−5	5.82	4.698E−4	5.84
	1/8	2.281E−7	5.52	4.120E−7	6.06	7.749E−6	5.69
	1/16	1.268E−8	4.23	2.332E−8	4.21	3.920E−7	4.37
Non-optimized	1	9.854E−3	–	2.427E−2	–	2.021E−1	–
	1/2	1.631E−3	2.51	3.854E−3	2.56	4.584E−2	2.07
	1/4	1.362E−4	3.89	3.009E−4	4.00	2.722E−3	4.43
	1/8	2.640E−6	5.47	7.295E−6	5.16	9.768E−5	4.61
	1/16	2.785E−8	6.67	6.359E−8	6.95	9.705E−7	6.76
Huang et al. [18]	1	2.994E−3	–	5.559E−3	–	3.375E−2	–
	1/2	2.257E−4	3.60	4.969E−4	3.37	5.335E−3	2.57
	1/4	1.170E−5	4.64	2.819E−5	4.50	4.001E−4	4.06
	1/8	5.793E−7	4.17	1.303E−6	4.26	9.437E−6	5.20
	1/16	2.420E−8	4.65	5.503E−8	4.64	5.257E−7	4.23

Table 4: Accuracy test results for the isentropic vortex problem on low-quality triangular meshes.

Schemes	Grid size	L_1 error	Order	L_2 error	Order	L_∞ error	Order
present	1	9.606E−3	–	1.515E−2	–	8.417E−2	–
	1/2	2.524E−3	1.93	3.901E−3	1.96	2.355E−2	1.84
	1/4	1.398E−4	4.17	2.079E−4	4.23	1.460E−3	4.01
	1/8	3.193E−6	5.45	4.747E−6	5.45	2.780E−5	5.72
	1/16	1.553E−7	4.36	2.220E−7	4.42	1.659E−6	4.07
referecne	1	9.183E−3	–	1.650E−2	–	1.064E−1	–
	1/2	2.216E−3	2.05	4.092E−3	2.01	2.895E−2	1.88
	1/4	1.605E−4	3.79	2.563E−4	4.00	2.162E−3	3.74
	1/8	3.363E−6	5.58	5.570E−6	5.52	5.521E−5	5.29
	1/16	1.423E−7	4.56	2.090E−7	4.74	1.659E−6	5.06
Non-optimized	1	1.470E−2	–	3.090E−2	–	2.027E−1	–
	1/2	3.813E−3	1.95	7.032E−3	2.14	6.322E−2	1.68
	1/4	8.192E−4	2.22	1.481E−3	2.25	1.626E−2	1.96
	1/8	3.784E−5	4.44	6.168E−5	4.59	5.710E−4	4.83
	1/16	6.998E−7	5.76	1.158E−6	5.74	9.167E−6	5.96
Huang et al. [18]	1	1.545E−2	–	2.200E−2	–	8.320E−2	–
	1/2	1.281E−2	0.27	2.239E−2	-0.03	1.587E−1	-0.93
	1/4	–	–	–	–	–	–

Table 5: Memory reduction for the low-storage variational finite volume scheme.

mesh	Reconstruction memory reduction	Global memory reduction
quadrilateral	80%	50.4 %
triangle	75%	42.6%

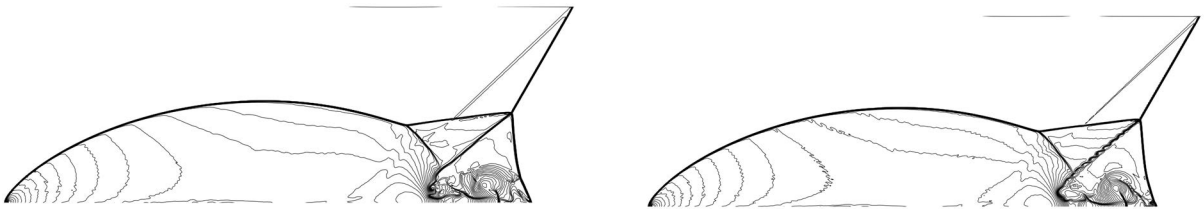
We also use this example to test the storage improvement compared with the traditional VFV scheme with the reference weights on the quadrilateral and triangular meshes with grid size $h = 5/128$ and $h = 1/256$, respectively. The results are shown in Table 5. Here, the "reconstruction memory reduction" refers to the proportion of saved matrices relative to the original number of matrices, while "global memory reduction" denotes the proportion of the low-storage VFV scheme relative to the original traditional VFV scheme with regard to the peak physical memory consumption observed during the actual execution of the program.

Remark 1 *Due to the NaN issue occurring for the low-storage VFV scheme designed with the weights of Huang et al. [18] on the low-quality triangular meshes, subsequent examples will no longer display results using these weights.*

4.2 Double Mach reflection problem

The double-Mach reflection problem [39] is a widely used benchmark for assessing high-resolution numerical schemes. The computational domain is $[0, 4] \times [0, 1]$, with a reflecting wall imposed along the bottom boundary starting from $x = 1/6$. Initially, a right-moving shock with $Ma = 10$ is located at $x = 1/6, y = 0$, inclined 60° with respect to the x-axis. The simulations are performed until $t = 0.2$. The computations are performed with grid size $h = 1/240$ using a physical time step $\Delta t = 0.0002$ on triangular mesh, and with grid size $h = 1/360$ using a physical time step $\Delta t = 0.0001$ on quadrilateral mesh. The CFL number for the local pseudo time step for all meshes is 10 and the inner iteration converges when the density residual drops by 3 orders of magnitude

Figure 8 and 9 show density contour plots generated by the low-storage VFV schemes on the quadrilateral mesh. As shown in these figures, both schemes produce essentially non-oscillatory solutions. However, the optimized scheme resolves the complex flow structures more sharply than the reference one. In particular, it captures a clearer shear layer emanating from the triple point and more detailed vortex structures near the Mach stem. Moreover, the Mach stem is better preserved and less distorted in the optimized result, whereas the reference scheme produces a more smeared shear layer and a slightly distorted Mach stem, indicating reduced shock-capturing accuracy.

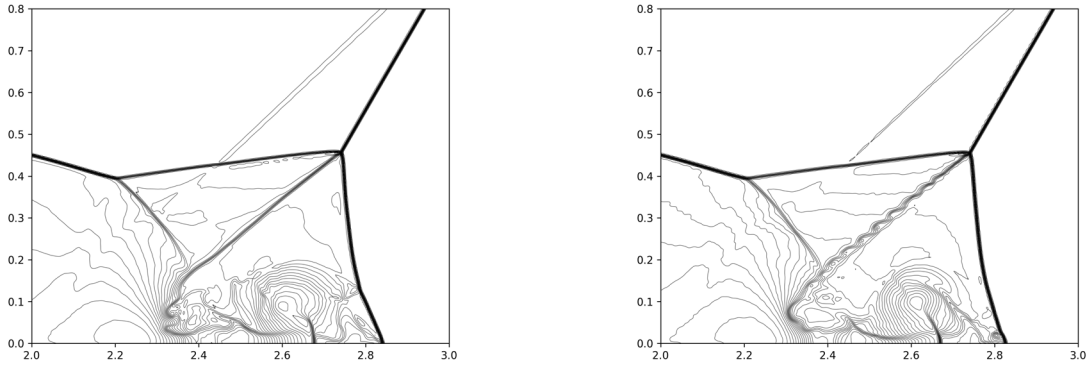


(a) Reference (b) Present

Figure 8: Comparison of density contours for double Mach reflection on the quadrilateral mesh.

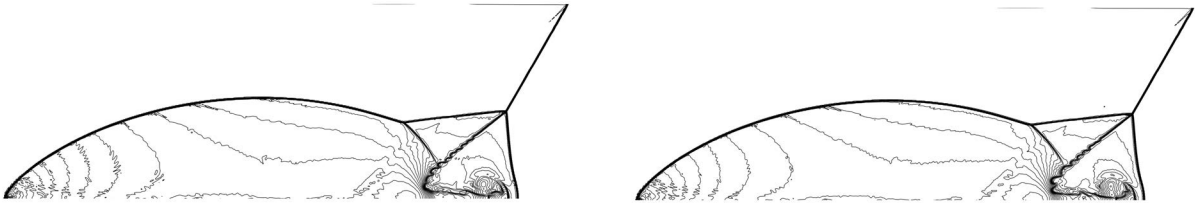
Figure 10 and 11 show density contour plots generated by the low-storage VFV scheme on the triangular mesh. The two schemes give very similar numerical results. The main shock structures and shear-layer patterns are well captured in both cases. While the present scheme shows a slightly more coherent near-wall vortex structure, the overall improvement over the reference scheme is modest.

In Table 6 , we present a comparison of the computational speeds, i.e., the CPU times, of the low-storage VFV schemes defined in different weights for the double-mach reflection problem. The optimized weights for the low-storage VFV schemes use 27.8% and 21.3% less CPU times than the reference scheme on quadrilateral and triangular meshes, respecitvely. Therefore, the optimized schemes achieve improved or at least comparable resolution with reduced computational cost, demonstrating a better balance between accuracy and efficiency than the reference scheme.



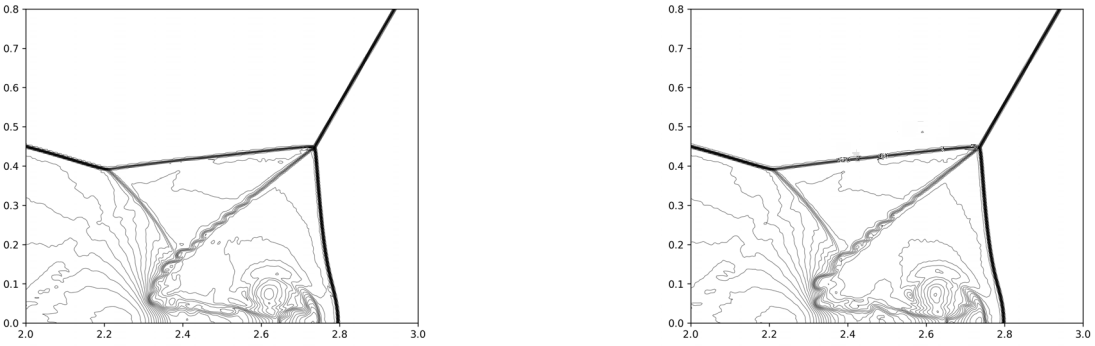
(a) Reference (b) Present

Figure 9: Close-up view around the double Mach stem of Fig. 8.



(a) Reference (b) Present

Figure 10: Comparison of density contours for double Mach reflection on the triangular mesh.



(a) Reference (b) Present

Figure 11: Close-up view around the double Mach stem of Fig. 10

Table 6: CPU time comparison for the double Mach reflection problem.

Mesh	quadrilateral		triangular	
Schemes	Reference	Present	Reference	Present
CPU time (s)	60 916.4	43 963.7	35 993.8	28 345.0

4.3 Subsonic flow past a NACA0012 airfoil

This problem is used to assess the convergence performance of the low-storage VFV schemes for steady-state problems. This test case simulates an inviscid subsonic flow past a NACA0012 airfoil at an angle of attack of $\alpha = 2^\circ$, with a free-stream condition of $Ma_\infty = 0.5$. Only the results for the quadrilateral mesh are shown here. The computational domain is discretized using an O-type quadrilateral mesh comprising 5,143 quadrilateral elements and 5,253 grid points. The airfoil surface contains 150 grid points. The simulation is started with the free-stream flow conditions and the CFL number for the local pseudo time step is 40.

As shown in Figure 12, all the schemes converge to machine zero, demonstrating the superior convergence properties of these schemes. They also use almost the same CPU time to reach the steady-state solution. The L_1, L_2 and L_∞ entropy production errors are list in Table 7 that the optimized scheme yields more accurate solutions than the reference and non-optimized schemes, while requiring almost the same amount of CPU time. This indicates that the optimized scheme is more efficient.

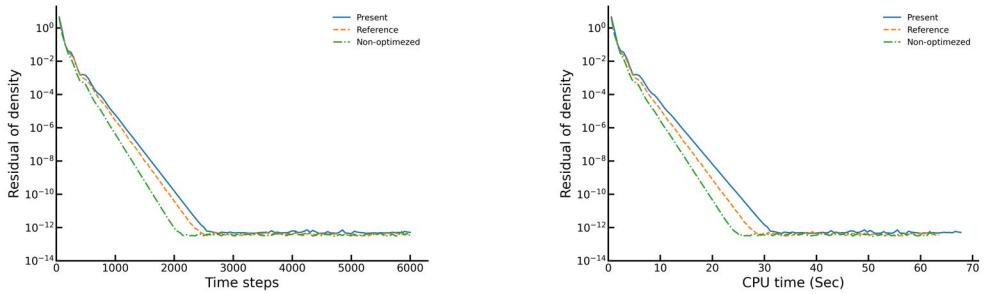


Figure 12: Convergence history comparison for subsonic flow past a NACA0012 airfoil problem.

Table 7: Entropy errors of the steady-state solution.

entropy error	Present	Reference	Non-optimized
L_1	3.20E-4	5.33E-4	1.85E-3
L_2	1.41E-3	2.27E-3	6.17E-3
L_∞	1.93E-2	2.81E-2	6.02E-2

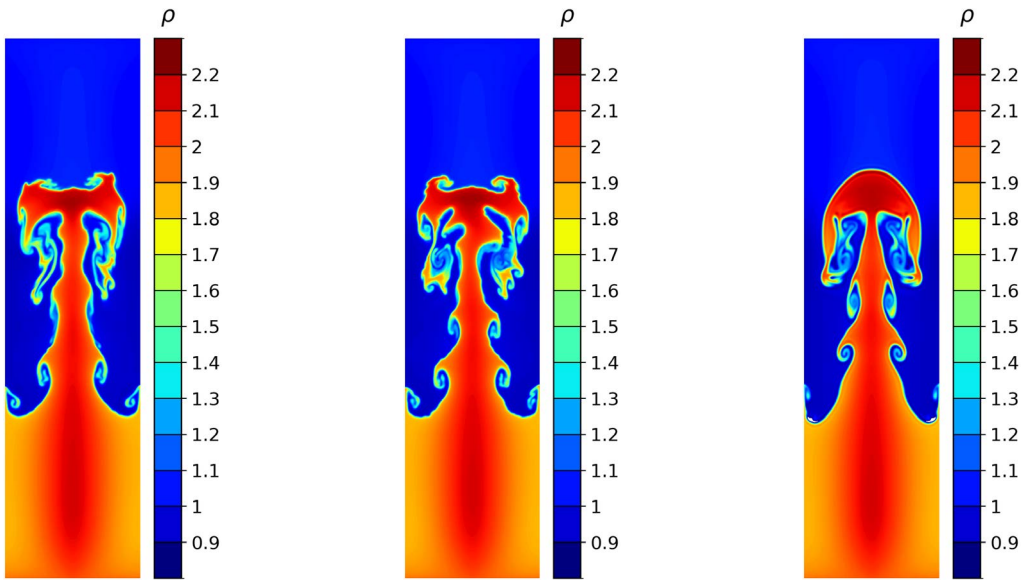
4.4 Rayleigh-Taylor instability

The two-dimensional inviscid Rayleigh–Taylor instability [40], which involves discontinuities and complex flow structures, is a widely used test case to validate the resolution of numerical schemes. The computational domain is $[0, 0.25] \times [0, 1]$. Initially, a contact discontinuity is formed at the $y = 0.5$ interface by imposing two fluids with different densities. The mixing of two fluids is driven by gravity and Rayleigh-Taylor instability occurs at the interface. The initial condition is given by

$$(\rho, u, v, p) = \begin{cases} (2,\ 0,\ -0.025\, c \cos 8\pi x,\ 1 + 2y)\,, & 0 \le y \le 0.5, \\ (1,\ 0,\ -0.025\, c \cos 8\pi x,\ 1.5 + y)\,, & 0.5 < y \le 1.0, \end{cases} \tag{43}$$

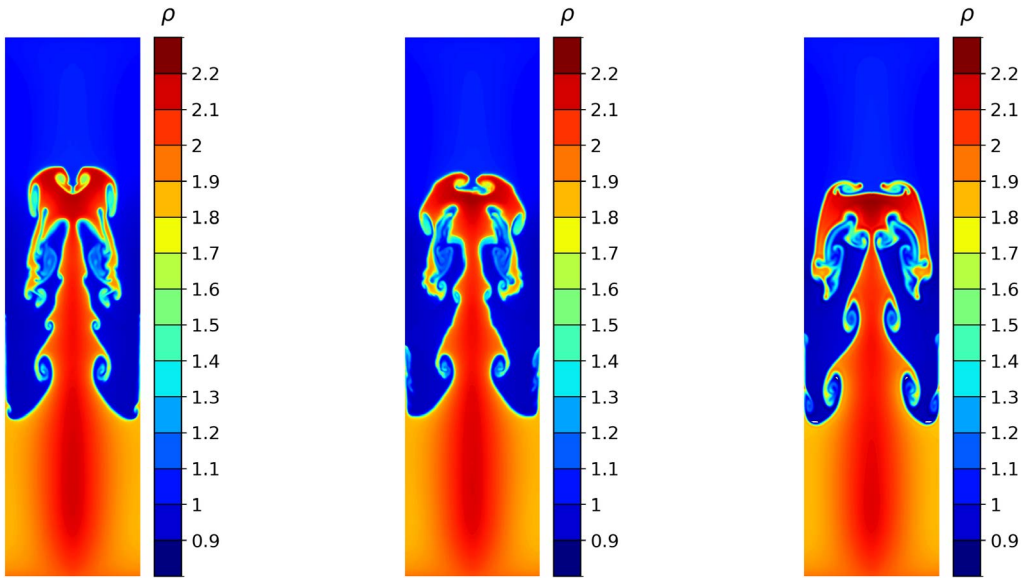
where $c = \sqrt{\gamma p/\rho}$ is the speed of sound. In this test case, the ratio of specific heat is taken as $\gamma = 5/3$. The governing equations are the two-dimensional Euler equations (1) with gravitational source terms on the right-hand-side of y-momentum and energy equations, expressed in terms of ρ and ρv, respectively. Symmetric boundary conditions are imposed on the left and right sides. Dirichlet boundary conditions $(\rho, u, v, p) = (1, 0, 0, 2.5)$ and $(\rho, u, v, p) = (2, 0, 0, 1)$ are imposed on the top and bottom sides, respectively. The quadrilateral computational mesh is performed with grid size $h = 1/800$. The

triangular computational mesh is obtained by splitting each square element of a uniform $h = 1/512$ quadrilateral mesh into two right triangles. The simulation is performed up to $t = 1.95$ with time step $\Delta t = 0.001$. The CFL number for the local pseudo time step is 10.



(a) Reference (b) Present (c) Non-optimized

Figure 13: Comparison of density contours for the Rayleigh-Taylor instability problem at t = 1.95 on quadrilatera mesh, computed by the low-storage VFV schemes with WBAP limiter.



(a) Reference (b) Present (c) Non-optimized

Figure 14: Comparison of density contours for the Rayleigh-Taylor instability problem at t = 1.95 on triangular mesh, computed by the low-storage VFV schemes with WBAP limiter.

The density contours computed by the low-storage VFV schemes are shown in Figure 13 and 14. In both Figure 13 and 14, the reference and optimized results exhibit highly nonlinear flow morphologies, including a mushroom-shaped interface, pronounced interfacial roll-up, and fine-scale vortical structures near the mixing layer. In contrast, the non-optimized result mainly captures the large-scale plume structure but fails to accurately resolve the detailed interfacial dynamics.

Table 8: CPU time comparison for the Rayleigh-Taylor instability problem.

Mesh	quadrilateral			triangular		
Schemes	Reference	Present	Non-optimized	Reference	Present	Non-optimized
CPU time (s)	17 640.4	16 114.4	17 161.6	14 901.5	13 593.1	12 419.3

In Table 8, we present a comparison of the CPU times by the low-storage VFV schemes defined in different weights for the Rayleigh-Taylor instability problem. The optimized weights for the low-storage VFV schemes use less CPU times than the reference schemes on both quadrilateral and triangular meshes. These results demonstrate that the optimized schemes provide accurate and computationally efficient approachs for simulating the Rayleigh–Taylor instability.

5. Conclusion and Future Work

This paper develop a low-storage variational reconstruction for the high-order finite volume method on unstructured grids to reduce storage overhead. In the low-storage variational reconstruction, an interfacial jump function is defined to measure the jumps of reconstruction polynomial and its spatial derivatives between adjacent cells at the interface centers. Furthermore, a posterior optimization based on ensembel Kalman inversion is proposed to optimize the corresponding derivative weights for different geometric partitionings to improve the accuracy, effiency and robustness of the low-sotrage variational finite volume scheme. The reciprocal of efficiency is used to design the objective function for optimization, which is obtained by solving the linear advection equation with a wave pocket composed by a series of weighted sine waves as the initial condition on both high- and low-quality meshes, over a sufficient duration to allow all waves to propagate through the entire computational domain. In the future, we plan to extend this optimization method to three dimensions and address the issue of derivative weights design in anisotropic meshes.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (Grant 12372284).

References

[1] Bram Van Leer. Towards the ultimate conservative difference scheme. *Journal of computational physics*, 135(2):229–248, 1997.

[2] Carl F Ollivier-Gooch. Quasi-eno schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction. *Journal of Computational Physics*, 133(1):6–17, 1997.

[3] Michael Dumbser, Martin Käser, Vladimir A Titarev, and Eleuterio F Toro. Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *Journal of Computational Physics*, 226(1):204–243, 2007.

[4] Qian Wang, Yu-Xin Ren, Jianhua Pan, and Wanai Li. Compact high order finite volume method on unstructured grids iii: Variational reconstruction. *Journal of Computational physics*, 337:1–26, 2017.

[5] Hiroaki Nishikawa and Jeffery A White. An efficient quadratic interpolation scheme for a third-order cell-centered finite-volume method on tetrahedral grids. *Journal of Computational Physics*, 490:112324, 2023.

[6] William H Reed and Thomas R Hill. Triangular mesh methods for the neutron transport equation. Technical report, Los Alamos Scientific Lab., N. Mex.(USA), 1973.

[7] Bernardo Cockburn and Chi-Wang Shu. Runge–kutta discontinuous galerkin methods for convection-dominated problems. *Journal of scientific computing*, 16(3):173–261, 2001.

[8] Michael Dumbser, Dinshaw S Balsara, Eleuterio F Toro, and Claus-Dieter Munz. A unified framework for the construction of one-step finite volume and discontinuous galerkin schemes on unstructured meshes. *Journal of Computational Physics*, 227(18):8209–8253, 2008.

[9] Lingquan Li, Xiaodong Liu, and Hong Luo. A reconstructed discontinuous galerkin method based on variational formulation for compressible flows. *Journal of Computational Physics*, 466:111406, 2022.

[10] Laiping Zhang, Liu Wei, He Lixin, Deng Xiaogang, and Zhang Hanxin. A class of hybrid dg/fv methods for conservation laws i: Basic formulation and one-dimensional systems. *Journal of Computational Physics*, 231(4):1081–1103, 2012.

[11] Rémi Abgrall and Mohamed Mezine. Construction of second order accurate monotone and stable residual distribution schemes for unsteady flow problems. *Journal of Computational Physics*, 188(1):16–55, 2003.

[12] Hung T Huynh. A flux reconstruction approach to high-order schemes including discontinuous galerkin methods. In *18th AIAA computational fluid dynamics conference*, page 4079, 2007.

[13] Peter E Vincent, Patrice Castonguay, and Antony Jameson. A new class of high-order energy stable flux reconstruction schemes. *Journal of Scientific Computing*, 47(1):50–72, 2011.

[14] Zhi Jian Wang and Haiyang Gao. A unifying lifting collocation penalty formulation including the discontinuous galerkin, spectral volume/difference methods for conservation laws on mixed grids. *Journal of Computational Physics*, 228(21):8161–8186, 2009.

[15] Qian Wang, Yu-Xin Ren, and Wanai Li. Compact high order finite volume method on unstructured grids i: Basic formulations and one-dimensional schemes. *Journal of Computational Physics*, 314:863–882, 2016.

[16] Qian Wang, Yu-Xin Ren, and Wanai Li. Compact high order finite volume method on unstructured grids ii: Extension to two-dimensional euler equations. *Journal of Computational Physics*, 314:883–908, 2016.

[17] Yu-Si Zhang, Yu-Xin Ren, and Qian Wang. Compact high order finite volume method on unstructured grids iv: Explicit multi-step reconstruction schemes on compact stencil. *Journal of Computational Physics*, 396:161–192, 2019.

[18] Qian-Min Huang, Yu-Xin Ren, Qian Wang, and Jian-Hua Pan. High-order compact finite volume schemes for solving the reynolds averaged navier-stokes equations on the unstructured mixed grids with a large aspect ratio. *Journal of Computational Physics*, 467:111458, 2022.

[19] Chong-Bo Zhou, Qian Wang, and Yu-Xin Ren. Machine learning optimization of compact finite volume methods on unstructured grids. *Journal of Computational Physics*, 500:112746, 2024.

[20] Hanyu Zhou and Qian Wang. Positivity-preserving implicit finite volume methods on unstructured grids for compressible flows. *Journal of Computational Physics*, page 115014, 2026.

[21] PAN Jianhua, WANG Qian, REN Yuxin, et al. High-order compact finite volume methods on unstructured grids with adaptive mesh refinement for solving inviscid and viscous flows. *Chinese Journal of Aeronautics*, 31(9):1829–1841, 2018.

[22] Marco A Iglesias, Kody JH Law, and Andrew M Stuart. Ensemble kalman methods for inverse problems. *Inverse Problems*, 29(4):045001, 2013.

[23] Claudia Schillings and Andrew M Stuart. Analysis of the ensemble kalman filter for inverse problems. *SIAM Journal on Numerical Analysis*, 55(3):1264–1290, 2017.

[24] Philip L Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2):357–372, 1981.

[25] Richard Sanders, Eric Morano, and Marie-Claude Druguet. Multidimensional dissipation for upwind schemes: stability and applications to gas dynamics. *Journal of Computational Physics*, 145(2):511–537, 1998.

[26] Rémi Abgrall. On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. *Journal of Computational Physics*, 114(1):45–58, 1994.

[27] Oliver Friedrich. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *Journal of computational physics*, 144(1):194–212, 1998.

[28] Hong Luo, Joseph D Baum, and Rainald Löhner. A discontinuous galerkin method based on a taylor basis for the compressible flows on arbitrary grids. *Journal of Computational Physics*, 227(20):8875–8893, 2008.

[29] Wanai Li, Yu-Xin Ren, Guodong Lei, and Hong Luo. The multi-dimensional limiters for solving hyperbolic conservation laws on unstructured grids. *Journal of Computational Physics*, 230(21):7775–7795, 2011.

[30] Wanai Li and Yu-Xin Ren. The multi-dimensional limiters for solving hyperbolic conservation laws on unstructured grids ii: extension to high order finite volume schemes. *Journal of Computational Physics*, 231(11):4053–4077, 2012.

[31] Wanai Li and Yu-Xin Ren. High-order k-exact weno finite volume schemes for solving gas dynamic euler equations on unstructured grids. *International Journal for Numerical Methods in Fluids*, 70(6):742–763, 2012.

[32] Antony Jameson and Eli Turkel. Implicit schemes and lu decompositions. *Mathematics of Computation*, 37(156):385–397, 1981.

[33] Hong Luo, Joseph D Baum, and Rainald Löhner. A fast, matrix-free implicit method for compressible flows on unstructured grids. *Journal of Computational Physics*, 146(2):664–690, 1998.

[34] Luca Ferracina and Marc N Spijker. Strong stability of singly-diagonally-implicit runge–kutta methods. *Applied Numerical Mathematics*, 58(11):1675–1686, 2008.

[35] Zhen-Sheng Sun, Yu-Xin Ren, Cédric Larricq, Shi-ying Zhang, and Yue-cheng Yang. A class of finite difference schemes with low dispersion and controllable dissipation for dns of compressible turbulence. *Journal of computational physics*, 230(12):4616–4635, 2011.

[36] QiuJu Wang, YuXin Ren, ZhenSheng Sun, and YuTao Sun. Low dispersion finite volume scheme based on reconstruction with minimized dispersion and controllable dissipation. *Science China Physics, Mechanics and Astronomy*, 56(2):423–431, 2013.

[37] Marco Iglesias and Yuchen Yang. Adaptive regularisation for ensemble kalman inversion. *Inverse Problems*, 37(2):025008, 2021.

[38] Changqing Hu and Chi-Wang Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150(1):97–127, 1999.

[39] Paul Woodward and Phillip Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of computational physics*, 54(1):115–173, 1984.

[40] Zhengfu Xu and Chi-Wang Shu. Anti-diffusive flux corrections for high order finite difference weno schemes. *Journal of Computational Physics*, 205(2):458–485, 2005.