A Quasi-Conservative Discontinuous Galerkin Method for Multi-component Flows Using the Non-oscillatory Kinetic Flux II: ALE Framework

Dongmi Luo[1] · Shiyi Li[1] · Weizhang Huang[2] · Jianxian Qiu[3] · Yibing Chen[1]

Abstract

A high-order quasi-conservative discontinuous Galerkin (DG) method is proposed for the numerical simulation of compressible multi-component flows. A distinct feature of the method is a predictor-corrector strategy to define the grid velocity. A Lagrangian mesh is first computed based on the flow velocity and then used as an initial mesh in a moving mesh method (the moving mesh partial differential equation or MMPDE method) to improve its quality. The fluid dynamic equations are discretized in the direct arbitrary Lagrangian-Eulerian framework using DG elements and the non-oscillatory kinetic flux while the species equation is discretized using a quasi-conservative DG scheme to avoid numerical oscillations near material interfaces. A selection of one- and two-dimensional examples are presented to verify the convergence order and the constant-pressure-velocity preservation property of the method. They also demonstrate that the incorporation of the Lagrangian meshing with the

✉ Yibing Chen
  chen_yibing@iapcm.ac.cn

  Dongmi Luo
  dongmiluo@stu.xmu.edu.cn

  Shiyi Li
  lishiyi14@tsinghua.org.cn

  Weizhang Huang
  whuang@ku.edu

  Jianxian Qiu
  jxqiu@xmu.edu.cn

[1] Institute of Applied Physics and Computational Mathematics, Beijing 100088, China

[2] Department of Mathematics, University of Kansas, Lawrence, KS 66045, USA

[3] School of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computing, Xiamen University, Xiamen 361005, Fujian, China

MMPDE moving mesh method works well to concentrate mesh points in regions of shocks and material interfaces.

1 Introduction

The numerical simulation of compressible multi-component flows plays a significant role in the computational fluid dynamics due to their application in a wide range of fields, including inertial confinement fusion, combustion, and flame propagation in detonation. Those flows contain rich physical features and structures such as shock waves, contact waves, and material interfaces. How to resolve these structures accurately and efficiently presents a challenge in the numerical simulation of compressible multi-component flows.

There exist two types of frameworks to describe fluid motion, i.e., the Eulerian framework and the Lagrangian framework. In the former framework, the mesh is fixed, which makes methods based on the framework suitable for problems with large deformations. But contact discontinuities are smeared typically in the Eulerian framework. On the other hand, methods based on the Lagrangian framework or called Lagrangian methods (e.g., see [11,12,40]), where the mesh moves with the fluid, are more suitable for flows with contact discontinuities. Unfortunately, the corresponding computation can easily fail due to mesh distortion caused by large flow deformations. To overcome this problem, an arbitrary Lagrangian-Eulerian (ALE) method was first proposed by Hirt et al. [21] where the grid velocity can be chosen independently from the local flow velocity and therefore the mesh points can be moved to maintain the mesh quality. Roughly speaking, ALE methods can be split into two major groups, indirect ALE methods (e.g., see [32,39]) and direct ALE methods (e.g., see [5,31,49]). An indirect ALE method typically includes three steps, i.e., the Lagrangian, rezoning, and remapping steps. In the Lagrangian step, the physical variables and the mesh are updated at the same time and the mesh update is based on the flow velocity. The mesh quality is improved using a remeshing strategy in the rezoning step. Finally, the physical solutions are projected from the old mesh to the new one in the remapping step. In a direct ALE method, the mesh movement is taken into account directly in the discretization of the governing equations and the remapping step is avoided. More high-order methods have been developed with the direct ALE approach since it avoids the remapping step and the need for high-order conservative remapping schemes. For example, a high-order moving mesh kinetic scheme has been proposed [58] for compressible flows on unstructured meshes based on the direct ALE approach. Moreover, a new class of high-order ALE one-step finite volume schemes has been developed in [5] for two- and three-dimensional hyperbolic partial differential equations (PDEs).

Discontinuous Galerkin (DG) methods have been incorporated into the ALE framework; e.g, see [19,31,36,42–44,49,60]. DG methods, first applied to neutron transport equations [48] and later extended to general nonlinear systems of hyperbolic conservation laws in a series of papers by Cockburn and Shu and coworkers [13–16], have been successfully used for multi-material flows [10,20,38,45,46,50,56,61]. They are known to be high-order and compact and can readily be combined with various mesh adaptation strategies. Examples

of DG-ALE methods for compressible multiphase flows include that by Pandare et al. [43] where the HLLC numerical flux is employed but a constant or a linearly varying grid velocity is assumed in the computation. Recently, an ALE Runge-Kutta DG method was proposed by Zhao et al. [60] for multi-material flows where a conservative equation related to a γ-model is coupled with the system of fluid equations and the grid velocity is obtained by a variational approach.

In the last four decades a variety of r-adaptive or (adaptive) moving mesh methods have been developed in the context of the numerical solution of general PDEs; e.g., see the books/review articles [3,4,6,26,55] and references therein. Moving mesh methods share many common features with ALE methods and in particular, their rezoning version and quasi-Lagrangian version can be compared directly with indirect and direct ALE methods, respectively; see discussion in [26]. The main difference between moving mesh methods and ALE methods lies in how adaptive meshes are generated. As mentioned earlier, an adaptive mesh is obtained in ALE methods through the flow velocity (Lagrangian meshing) and a remeshing strategy. This approach has the advantage that mesh points (viewed as fluid particles) are naturally moved to and concentrated in regions of shocks and contact/material interfaces. Its main disadvantage is that a Lagrangian mesh can often be distorted and even tangling and thus cannot be used directly in the numerical solution of PDEs. This is also the reason why a remeshing strategy must be used to effectively improve the quality of the mesh while maintaining a level of mesh concentration in the Lagrangian mesh. On the other hand, an adaptive mesh is generated in moving mesh methods by either integrating a set of moving mesh equations or minimizing a meshing functional. This approach has the advantage that it generally generates a mesh of good quality. The mesh adaptation in this approach is typically associated with an error estimate instead of the flow velocity, which can be viewed as a drawback of the approach when applied to the computational fluid dynamics.

The objective of the current work is to study a high-order DG-ALE method for multi-component flows by incorporating the Lagrangian meshing with a moving mesh method. The incorporation is realized with a predictor-corrector strategy, i.e., a Lagrangian mesh is first computed and then used as an initial mesh for a moving mesh method to improve mesh quality. We use the moving mesh PDE (MMPDE) method [24–26,37] for this purpose. The MMPDE method moves a mesh through a set of ordinary differential equations (cf. (3.7)) while concentrating points according to a user-supplied metric tensor that provides the magnitude and directional information to control the size, shape, and orientation of mesh elements throughout the physical domain. With a proper choice of the metric tensor, the MMPDE method is expected to maintain a similar level of point concentration as in the Lagrangian mesh while improving the mesh quality. Moreover, it has been shown analytically and numerically in [23] that a formulation of the MMPDE method, at semi-discrete or fully discrete level, leads to a non-singular (tangling-free) mesh for any domain in any dimension when the metric tensor is bounded and the initial mesh is non-singular. This provides a justification on the use of the MMPDE method to improve the quality of the Lagrangian mesh. With the incorporation of the Lagrangian meshing and the MMPDE method, we hope that the proposed DG-ALE method can have the advantages of both methods and is able to track rapid changes in the flow field including shocks and material interfaces. It is noted that this computation of the grid velocity is different from those in [43,60]. (The mesh is assumed to move at a constant or a linearly varying mesh velocity in [43] while the grid velocity is obtained by a variational approach in [60].) We adopt the non-oscillatory kinetic (NOK) flux [8,35,38] for the DG discretization of the Euler equations since it provides more physical information of the flow (than other numerical fluxes such as the commonly used Lax-Friedrich flux) and avoids the need to construct any Riemann solver. Moreover, we

use a quasi-conservative DG discretization [1] for the species equation to avoid numerical oscillations near material interfaces.

So far there exist many DG-ALE works for single material but only a few for multi-component flows. For example, Pandare et al. [43] proposed a DG-ALE method for compressible multiphase flows. However, a constant or linearly varying grid velocity was assumed in the computation and the physical field was not taken into consideration. Thus, the method cannot track shocks exactly. Recently, an ALE Runge-Kutta DG method was developed by Zhao et al. [60] for multi-material flows. Since it does not take a Lagrange step, the method may have difficulty to track material interfaces. The method presented in this work overcomes these drawbacks by employing a predictor-corrector approach to compute the mesh velocity and taking the advantages of both Lagrangian and moving mesh methods. It can keep good quality of the mesh while tracking discontinuities including shocks and material interfaces in the flow field. Moreover, the method is high-order and non-oscillatory at shocks and interfaces.

The organization of the paper is as follows. The DG method for conservative systems in the ALE framework and the discretization of the species equation are described in Sect. 2. In Sect. 3, the computation of the grid velocity and the MMPDE moving mesh method are discussed. One- and two-dimensional numerical examples are presented to demonstrate the accuracy and the mesh adaptation capability of the proposed DG-ALE method in Sect. 4. Conclusions are drawn in Sect. 5.

2 A DG-ALE Method for Multi-component Flows

2.1 The Govorning Equations

In this work we consider the 4-equation model for a compressible multi-component flow in the dimensionless form

$$\begin{cases} W_t + \nabla \cdot F(W) = 0, \\ \frac{\partial Y}{\partial t} + U\frac{\partial Y}{\partial x} + V\frac{\partial Y}{\partial y} = 0, \end{cases} \tag{2.1}$$

where

$$W = \begin{pmatrix} \rho \\ \rho U \\ \rho V \\ E \end{pmatrix}, \quad F(W) = \begin{pmatrix} \rho U, & \rho V \\ \rho U^2 + P, & \rho U V \\ \rho U V, & \rho V^2 + P \\ U(E+P), & V(E+P) \end{pmatrix},$$

ρ is the density, P is the pressure, $\mathbf{U} = (U, V)$ is the flow velocity, $E = \rho e + \frac{1}{2}\rho(U^2 + V^2)$ is the total energy, ρe is the internal energy, and Y is the volume fraction. We consider the stiffened gas equation of state (EOS)

$$\rho e = \frac{P + \gamma B}{\gamma - 1}, \tag{2.2}$$

where γ is the ratio of specific heats and B is a prescribed pressure-like constant. In this work we consider a two-component flow and assume that the material parameters for fluid 1 and fluid 2 are γ_1, B_1 and γ_2, B_2, respectively. The material parameters for mixing cells are

computed as (e.g., see [53])

$$\frac{1}{\gamma - 1} = \frac{Y}{\gamma_1 - 1} + \frac{1 - Y}{\gamma_2 - 1}, \tag{2.3}$$

$$\frac{\gamma B}{\gamma - 1} = \frac{Y \gamma_1 B_1}{\gamma_1 - 1} + \frac{(1 - Y)\gamma_2 B_2}{\gamma_2 - 1}. \tag{2.4}$$

Finally, the mixing sound speed is computed as

$$c = \sqrt{\frac{\gamma (P + B)}{\rho}}.$$

2.2 DG Discretization of the Fluid Dynamic Equations on Moving Meshes

We now consider the DG discretization of the first four equations in (2.1) on a moving triangular mesh with an arbitrary but given grid velocity. (See Sect. 3 for the determination of the grid velocity.) The resulting scheme reduces to the Eulerian and Lagrangian forms when the grid velocity is taken to be zero and the flow velocity, respectively.

For the moment we assume that a moving triangular mesh $\mathscr{T}_h$ for the domain Ω is given at time instants

$$0 = t_0 < t_1 < \cdots < t_n < t_{n+1} < \cdots \le T.$$

The appearances are denoted by $\mathscr{T}_h^n$, $n = 0, 1, \cdots$. Since they belong to the same mesh, they have the same number of elements (N) and vertices (N_v) and the same connectivity, and differ only in the location of the vertices. Denote the coordinates of the vertex of $\mathscr{T}_h^n$ by $\mathbf{x}_j^n$, $j = 1, 2, \cdots, N_v$. For any $t \in [t_n, t_{n+1}]$, the coordinates and velocities of the vertices of the mesh $\mathscr{T}_h(t)$ are defined as

$$\mathbf{x}_j(t) = \frac{t - t_n}{\Delta t_n} \mathbf{x}_j^{n+1} + \frac{t_{n+1} - t}{\Delta t_n} \mathbf{x}_j^n, \qquad j = 1, 2, \cdots, N_v \tag{2.5}$$

$$\dot{\mathbf{x}}_j(t) = \frac{\mathbf{x}_j^{n+1} - \mathbf{x}_j^n}{\Delta t_n}, \qquad j = 1, 2, \cdots, N_v \tag{2.6}$$

where $\Delta t_n = t_{n+1} - t_n$. Then the DG finite element space is defined as

$$V_h^k = \left\{ p(\mathbf{x}, t) : p|_K \in P^k(K), \forall K \in \mathscr{T}_h(t) \right\},$$

where $P^k(K)$ is the set of polynomials defined on K of degree no more than k. The semi-discrete DG approximation of (2.1) is to find $W_h(\cdot, t) \in V_h^k, t \in (0, T]$ such that

$$\int_K \frac{\partial W_h}{\partial t} \psi \, dx dy + \int_{\partial K} F \cdot \mathbf{n} \psi \, ds - \int_K F \cdot \nabla \psi \, dx dy = 0, \quad \forall \psi \in P^k(K), \quad \forall K \in \mathscr{T}_h(t) \tag{2.7}$$

where $\mathbf{n} = (n_x, n_y)$ is the outward unit normal vector of the triangular boundary ∂K. It is not difficult to show [30] that

$$\frac{\partial \psi(\mathbf{x}, t)}{\partial t} = -\nabla \psi(\mathbf{x}, t) \cdot \dot{X}(\mathbf{x}, t), \quad \forall \psi \in P^k(K)$$

where $\dot{X}(\mathbf{x}, t)$ is the piecewise linear interpolant of the nodal velocities. From this and the Reynolds transport theorem, we have

$$
\begin{aligned}
\frac{d}{dt} \int_K W_h \psi \, dx dy &= \int_K \frac{\partial (W_h \psi)}{\partial t} dx dy + \int_{\partial K} W_h \dot{X} \cdot \mathbf{n} \psi \, ds \\
&= \int_K \frac{\partial W_h}{\partial t} \psi \, dx dy + \int_K \frac{\partial \psi}{\partial t} W_h \, dx dy + \int_{\partial K} W_h \dot{X} \cdot \mathbf{n} \psi \, ds \\
&= \int_K \frac{\partial W_h}{\partial t} \psi \, dx dy - \int_K W_h \dot{X} \cdot \nabla \psi \, dx dy + \int_{\partial K} W_h \dot{X} \cdot \mathbf{n} \psi \, ds,
\end{aligned}
$$

which yields

$$
\int_K \frac{\partial W_h}{\partial t} \psi \, dx dy = \frac{d}{dt} \int_K W_h \psi \, dx dy + \int_K W_h \dot{X} \cdot \nabla \psi \, dx dy - \int_{\partial K} W_h \dot{X} \cdot \mathbf{n} \psi \, ds.
$$

Inserting this to (2.7) yields

$$
\frac{d}{dt} \int_K W_h \psi \, dx dy + \int_{\partial K} H(W_h) \psi \, ds - \int_K H_1(W_h) dx dy = 0, \tag{2.8}
$$

where

$$
H(W_h) = (F(W_h) - W_h \dot{X}) \cdot \mathbf{n} = \begin{pmatrix} \rho(\mathbf{U} - \dot{X}) \cdot \mathbf{n} \\ \rho U(\mathbf{U} - \dot{X}) \cdot \mathbf{n} + P n_x \\ \rho V(\mathbf{U} - \dot{X}) \cdot \mathbf{n} + P n_y \\ E(\mathbf{U} - \dot{X}) \cdot \mathbf{n} + P \mathbf{U} \cdot \mathbf{n} \end{pmatrix},
$$

$$
H_1(W_h) = (F(W_h) - W_h \dot{X}) \cdot \nabla \psi = \begin{pmatrix} \rho(\mathbf{U} - \dot{X}) \cdot \nabla \psi \\ \rho U(\mathbf{U} - \dot{X}) \cdot \nabla \psi + P \psi_x \\ \rho V(\mathbf{U} - \dot{X}) \cdot \nabla \psi + P \psi_y \\ E(\mathbf{U} - \dot{X}) \cdot \nabla \psi + P \mathbf{U} \cdot \nabla \psi \end{pmatrix}.
$$

Recall that these expressions reduce to the Eulerian form when $\dot{X} = 0$ and the Lagrangian form when $\dot{X} = \mathbf{U}$.

Applying a Gaussian quadrature rule to the second and third terms, we get

$$
\int_{\partial K} H(W_h) \psi \, ds \approx \sum_e \sum_{G_e} H(W_h(\mathbf{x}_{G_e})) \psi(\mathbf{x}_{G_e}) w_{G_e} |e|,
$$

$$
\int_K H_1(W_h) dx dy \approx \sum_G H_1(W_h(\mathbf{x}_G)) w_G |K|,
$$

where e denotes an edge of the element K, $|K|$ is the volume of K, $|e|$ is the length of e, $\mathbf{x}_G$ and $\mathbf{x}_{G_e}$ denote the Gauss points on K and e, respectively, and the summations $\sum_e$, $\sum_G$, and $\sum_{G_e}$ are taken over the edges of ∂K, Gauss points on K, and Gauss points on e, respectively. Replacing the flux H by a numerical flux $\hat{H}$, we obtain

$$
\frac{d}{dt} \int_K W_h \psi \, dx dy + \sum_e \sum_{G_e} \hat{H} \psi(\mathbf{x}_{G_e}^{int}) w_{G_e} |e| - \sum_G H_1(W_h(\mathbf{x}_G)) w_G |K| = 0,
$$

$$
\forall K \in \mathscr{T}_h, \ \psi \in V_h^k. \tag{2.9}
$$

Generally speaking, a numerical flux has the form $\hat{H} = \hat{H}(W_h(\mathbf{x}_G^{int}), W_h(\mathbf{x}_G^{ext}))$, where $W_h(\mathbf{x}_G^{int})$ and $W_h(\mathbf{x}_G^{ext})$ are defined as the values of W_h at $\mathbf{x} = \mathbf{x}_G$ from the interior and exterior of K, respectively.

In this work, we use the non-oscillatory kinetic (NOK) flux [8,9,38] that is a generalization of the gas kinetic scheme (GKS) [57] that avoids the need to construct Riemann solvers for the numerical solution of hyperbolic PDEs. Different from traditional Godunov-type methods, GKS is based on the Boltzmann equation and describes the flux function of the governing equations by particle collision of the transport process. Once the particle distribution function on a cell interface is obtained, the numerical flux can be calculated directly. The GKS has been successfully applied to multi-component flows [57]. Combined with a full conservative scheme, the GKS can work well when the difference between two species is small although oscillations may occur near material interfaces. Chen and Jiang [9] proposed a quasi-conservative scheme (called the non-oscillatory kinetic scheme or NOK) to overcome the problem and further extended it to more general material problems in [8].

The NOK flux is defined using the local coordinates. Denote

$$\tilde{U} = (\mathbf{U} - \dot{X}) \cdot \mathbf{n}, \quad \tilde{V} = (\mathbf{U} - \dot{X}) \cdot \mathbf{s}, \quad \tilde{E} = \rho e + \frac{1}{2}\rho(\tilde{U}^2 + \tilde{V}^2),$$

where $\mathbf{s} = (-n_y, n_x)$. Notice that $\tilde{U}$ and $\tilde{V}$ are the components of the vector $\mathbf{U} - \dot{X}$ along the normal and tangential directions of an edge e, respectively. Denoting $\dot{X} = (U_g, V_g)$, we have

$$U = U_g + \tilde{U}n_x - \tilde{V}n_y, \quad V = V_g + \tilde{U}n_y + \tilde{V}n_x,$$
$$U^2 + V^2 = (U_g^2 + V_g^2) + 2(U_g n_x + V_g n_y)\tilde{U} + 2(V_g n_x - U_g n_y)\tilde{V} + \tilde{U}^2 + \tilde{V}^2,$$
$$E = \tilde{E} + \frac{1}{2}\rho(U_g^2 + V_g^2) + \rho(U_g n_x + V_g n_y)\tilde{U} + \rho(V_g n_x - U_g n_y)\tilde{V}.$$

Introducing the new variable

$$\xi = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{pmatrix} = \begin{pmatrix} \rho\tilde{U} \\ \rho\tilde{U}^2 + P \\ \rho\tilde{U}\tilde{V} \\ \tilde{U}(\tilde{E} + P) \end{pmatrix}, \tag{2.10}$$

we can rewrite the flux along e as

$$\hat{H}(W_h) = \begin{pmatrix} \xi_1 \\ U_g\xi_1 + n_x\xi_2 - n_y\xi_3 \\ V_g\xi_1 + n_y\xi_2 + n_x\xi_3 \\ \frac{1}{2}(U_g^2 + V_g^2)\xi_1 + (U_g n_x + V_g n_y)\xi_2 + (V_g n_x - U_g n_y)\xi_3 + \xi_4 \end{pmatrix}.$$

This implies that $\hat{H}(W_h)$ can be computed on any edge e if we can compute the variable ξ on the edge. Moreover, recalling that $\tilde{U}$ and $\tilde{V}$ are the components of the vector $\mathbf{U} - \dot{X}$ along the normal and tangential directions, we know that $\tilde{U}$ has a jump whereas $\tilde{V}$ is continuous across the edge. Thus, the computation of ξ on e relies essentially on the computation of $\tilde{U}$ on e, which is a one-dimensional problem.

In the NOK flux, ξ is computed as

$$\xi = \eta\xi^K + (1 - \eta)\xi^E,$$

where ξ^E and ξ^K are the equilibrium and non-equilibrium parts to be defined below and $\eta \in [0, 1]$ is a relaxation parameter to determine the speed that a system evolves into an equilibrium state and be a function of local flow variables. For instance, η is defined in [34]

as a function of the flow jump around the cell interface, i.e.,

$$\eta = 1 - \exp\left[-C\frac{|P^+ - P^-|}{P^+ + P^-}\right],$$

where C is a problem-dependent positive constant ranging from 10^3 to 10^5. The parameter η should be set close to one for challenging problems such as those involving strong shocks.

The non-equilibrium part is

$$\xi^K = \xi_L^+ + \xi_R^-,$$

where the subscripts L and R denote the values of the corresponding variable from the interior and exterior of K, respectively, and

$$\xi_L^+ = \langle u^1 \rangle_+ \begin{pmatrix} \rho \\ \rho\tilde{U} \\ \rho\tilde{V} \\ \tilde{E} \end{pmatrix}_L + \begin{pmatrix} 0 \\ P_L\langle u^0 \rangle_+ \\ 0 \\ \frac{1}{2}P_L\langle u^1 \rangle_+ + \frac{1}{2}P_L\tilde{U}_L\langle u^0 \rangle_+ \end{pmatrix}, \tag{2.11}$$

$$\xi_R^- = \langle u^1 \rangle_- \begin{pmatrix} \rho \\ \rho\tilde{U} \\ \rho\tilde{V} \\ \tilde{E} \end{pmatrix}_R + \begin{pmatrix} 0 \\ P_R\langle u^0 \rangle_- \\ 0 \\ \frac{1}{2}P_R\langle u^1 \rangle_- + \frac{1}{2}P_R\tilde{U}_R\langle u^0 \rangle_- \end{pmatrix}, \tag{2.12}$$

$$\lambda = \min\left\{\frac{1}{c_L^2}, \frac{1}{c_R^2}\right\}, \tag{2.13}$$

$$\langle u^0 \rangle_+ = \frac{1}{2}\text{erfc}(-\sqrt{\lambda}\tilde{U}_L), \qquad \langle u^0 \rangle_- = \frac{1}{2}\text{erfc}(\sqrt{\lambda}\tilde{U}_R), \tag{2.14}$$

$$\langle u^1 \rangle_+ = \tilde{U}_L\langle u^0 \rangle_+ + \frac{1}{2}\frac{e^{-\lambda\tilde{U}_L^2}}{\sqrt{\pi\lambda}}, \qquad \langle u^1 \rangle_- = \tilde{U}_R\langle u^0 \rangle_- - \frac{1}{2}\frac{e^{-\lambda\tilde{U}_R^2}}{\sqrt{\pi\lambda}}, \tag{2.15}$$

c is the sound speed, $\text{erfc}(\cdot)$ is the complementary error function, and $\langle\cdot\rangle$ represents an average over the Maxwell-Boltzmann distribution.

In order to avoid oscillations of the pressure and velocity near a contact discontinuity, the equilibrium part should satisfy the consistent condition [8,9]. The primitive variables are computed by

$$\begin{pmatrix} \bar{\rho} \\ \tilde{U} \\ \tilde{V} \\ \bar{P} \\ \bar{Y} \end{pmatrix} = \begin{pmatrix} \rho_L <u^0>_+ + \rho_R <u^0>_- \\ <u^1>_+ + <u^1>_- \\ \tilde{V}_L <u^0>_+ + \tilde{V}_R <u^0>_- \\ P_L <u^0>_+ + P_R <u^0>_- \\ Y_L <u^0>_+ + Y_R <u^0>_- \end{pmatrix}. \tag{2.16}$$

Then we take

$$\xi^E = \begin{pmatrix} \bar{\rho}\tilde{U} \\ \bar{\rho}\tilde{U}^2 + \bar{P} \\ \bar{\rho}\tilde{U}\tilde{V} \\ \tilde{U}(\tilde{E} + \bar{P}) \end{pmatrix}, \tag{2.17}$$

and $\tilde{E}$ is determined by EOS. The reader is referred to [8] for the derivation of these formulas.

The traditional BGK scheme takes different values of $\lambda^L = \frac{1}{c_L^2}$ and $\lambda^R = \frac{1}{c_R^2}$ for the interior and exterior of K, which is known (e.g., see [8,9]) to introduce oscillations in velocity and pressure. On the other hand, the same value of λ, (2.13), is used for both the interior and exterior of K in the above NOK formula. This treatment is known [8] to overcome the oscillatory issue.

Finally we remark that the scheme (2.9) involves the grid velocity $\dot{X} = (U_g, V_g)$ which is defined as the piecewise linear interpolant of the nodal grid velocity. The latter is computed as the change rate of the location of vertices between the old and new meshes $\mathscr{T}_h^n$ and $\mathscr{T}_h^{n+1}$; cf. (2.6). The generation of the new mesh will be discussed in Sect. 3.

2.3 DG-ALE for Species Equation

In this subsection we consider the DG discretization of the species equation in (2.1). Numerical experiments show that a direct DG discretization of the equation can lead to oscillations in the pressure and velocity near material interfaces. The analysis by Abgrall [1] shows that those oscillations can be prevented if the scheme preserves constant velocity and constant pressure. We follow the procedure of Abgrall [1] and develop such a DG discretization of the species equation on a moving mesh in the following.

To start with, we assume that the velocity and the pressure are constant, i.e., $\mathbf{U} = \mathbf{U}_0 = (U_0, V_0)$ and $P = P_0$. Denote $\tilde{\mathbf{U}}_0 = \mathbf{U}_0 - \dot{X}$. We can rewrite $H(W_h)$ and $H_1(W_h)$ as

$$H(W_h) = \begin{pmatrix} \rho\tilde{\mathbf{U}}_0 \cdot \mathbf{n} \\ \rho U_0\tilde{\mathbf{U}}_0 \cdot \mathbf{n} + P_0 n_x \\ \rho V_0\tilde{\mathbf{U}}_0 \cdot \mathbf{n} + P_0 n_y \\ E\tilde{\mathbf{U}}_0 \cdot \mathbf{n} + P_0\mathbf{U}_0 \cdot \mathbf{n} \end{pmatrix}, \quad H_1(W_h) = \begin{pmatrix} \rho\tilde{\mathbf{U}}_0 \cdot \nabla\psi \\ \rho U_0\tilde{\mathbf{U}}_0 \cdot \nabla\psi + P_0\psi_x \\ \rho V_0\tilde{\mathbf{U}}_0 \cdot \nabla\psi + P_0\psi_y \\ E\tilde{\mathbf{U}}_0 \cdot \nabla\psi + P_0\mathbf{U}_0 \cdot \nabla\psi \end{pmatrix}. \quad (2.18)$$

Then, the continuity equation in (2.8) becomes

$$\frac{d}{dt}\int_K \rho\psi dxdy = -\int_{\partial K} \rho\tilde{\mathbf{U}}_0 \cdot \mathbf{n}\psi ds + \int_K \rho\tilde{\mathbf{U}}_0 \cdot \nabla\psi dxdy. \quad (2.19)$$

Similarly, the momentum equations read as

$$\frac{d}{dt}\int_K \rho U\psi dxdy = -\int_{\partial K} (\rho U_0\tilde{\mathbf{U}}_0 \cdot \mathbf{n} + P_0 n_x)\psi ds + \int_K (\rho U_0\tilde{\mathbf{U}}_0 \cdot \nabla\psi + P_0\psi_x)dxdy,$$

$$\frac{d}{dt}\int_K \rho V\psi dxdy = -\int_{\partial K} (\rho V_0\tilde{\mathbf{U}}_0 \cdot \mathbf{n} + P_0 n_y)\psi ds + \int_K (\rho V_0\tilde{\mathbf{U}}_0 \cdot \nabla\psi + P_0\psi_y)dxdy.$$

From the divergence theorem, we have $\int_K P_0\nabla\psi dxdy = \int_{\partial K} P_0\psi\mathbf{n}ds$. Inserting this into the above equations, we get

$$\frac{d}{dt}\int_K \rho U\psi dxdy = -\int_{\partial K} \rho U_0\tilde{\mathbf{U}}_0 \cdot \mathbf{n}\psi ds + \int_K \rho U_0\tilde{\mathbf{U}}_0 \cdot \nabla\psi dxdy,$$

$$\frac{d}{dt}\int_K \rho V\psi dxdy = -\int_{\partial K} \rho V_0\tilde{\mathbf{U}}_0 \cdot \mathbf{n}\psi ds + \int_K \rho V_0\tilde{\mathbf{U}}_0 \cdot \nabla\psi dxdy.$$

Comparing these with (2.19), we have

$$\frac{d}{dt}\int_K \rho U\psi dxdy = U_0\frac{d}{dt}\int_K \rho\psi dxdy, \quad \frac{d}{dt}\int_K \rho V\psi dxdy = V_0\frac{d}{dt}\int_K \rho\psi dxdy.$$
$$(2.20)$$

From the Reynolds transport theorem, we have

$$
\begin{aligned}
\frac{d}{dt} \int_K \rho U \psi \, dx dy &= \int_K U \frac{\partial(\rho \psi)}{\partial t} dx dy + \int_K \frac{\partial U}{\partial t} \rho \psi \, dx dy + \int_{\partial K} U \rho \psi \mathbf{n} \cdot \dot{X} ds \\
&= \int_K U_0 \frac{\partial(\rho \psi)}{\partial t} dx dy + \int_K \frac{\partial U}{\partial t} \rho \psi \, dx dy + \int_{\partial K} U_0 \rho \psi \mathbf{n} \cdot \dot{X} ds \\
&= U_0 \frac{d}{dt} \int_K \rho \psi \, dx dy + \int_K \frac{\partial U}{\partial t} \rho \psi \, dx dy.
\end{aligned}
$$

Similarly, we have

$$
\frac{d}{dt} \int_K \rho V \psi \, dx dy = V_0 \frac{d}{dt} \int_K \rho \psi \, dx dy + \int_K \frac{\partial V}{\partial t} \rho \psi \, dx dy.
$$

Combining the above results with (2.20), we get

$$
\int_K \rho \frac{\partial U}{\partial t} \psi \, dx dy = 0, \quad \int_K \rho \frac{\partial V}{\partial t} \psi \, dx dy = 0,
$$

which, from the arbitrariness of ψ and the positivity of ρ, implies $\frac{\partial U}{\partial t} = 0$ and $\frac{\partial V}{\partial t} = 0$. Thus, the scheme (2.8) preserves constant velocity solutions.

We now consider the preservation of constant pressure. The energy equation in (2.8) becomes

$$
\frac{d}{dt} \int_K E \psi \, dx dy = -\int_{\partial K} (E \tilde{\mathbf{U}}_0 \cdot \mathbf{n} + P_0 \mathbf{U}_0 \cdot \mathbf{n}) \psi ds + \int_K (E \tilde{\mathbf{U}}_0 \cdot \nabla \psi + P_0 \mathbf{U}_0 \cdot \nabla \psi) dx dy,
$$

which can be simplified into

$$
\frac{d}{dt} \int_K E \psi \, dx dy = -\int_{\partial K} E \tilde{\mathbf{U}}_0 \cdot \mathbf{n} \psi ds + \int_K E \tilde{\mathbf{U}}_0 \cdot \nabla \psi \, dx dy. \tag{2.21}
$$

Let $\kappa = \frac{1}{\gamma-1}$ and $\chi = \frac{\gamma B}{\gamma-1}$. Then we can rewrite the equation of state (2.2) into $\rho e = \kappa P + \chi$. From the definition of E, we have $E = \kappa P + \chi + \frac{1}{2} \rho (U^2 + V^2)$. Inserting this into (2.21), we get

$$
\begin{aligned}
\frac{d}{dt} \int_K \kappa P \psi \, dx dy &+ \frac{d}{dt} \int_K \chi \psi \, dx dy + \frac{d}{dt} \int_K \frac{1}{2} \rho (U^2 + V^2) \psi \, dx dy \\
&= -\int_{\partial K} \kappa P \tilde{\mathbf{U}}_0 \cdot \mathbf{n} \psi ds + \int_K \kappa P \tilde{\mathbf{U}}_0 \cdot \nabla \psi \, dx dy \\
&\quad - \int_{\partial K} \chi \tilde{\mathbf{U}}_0 \cdot \mathbf{n} \psi ds + \int_K \chi \tilde{\mathbf{U}}_0 \cdot \nabla \psi \, dx dy \\
&\quad - \int_{\partial K} \frac{1}{2} \rho (U^2 + V^2) \tilde{\mathbf{U}}_0 \cdot \mathbf{n} \psi ds + \int_K \frac{1}{2} \rho (U^2 + V^2) \tilde{\mathbf{U}}_0 \cdot \nabla \psi \, dx dy.
\end{aligned}
$$

Using the assumption that $U = U_0$, $V = V_0$, and $P = P_0$, the fact that $\frac{\partial U}{\partial t} = 0$ and $\frac{\partial V}{\partial t} = 0$, and the equation (2.19), we can derive from the above equation that

$$
\int_K \kappa \frac{\partial P}{\partial t} \psi \, dx dy = 0, \quad \text{or} \quad \frac{\partial P}{\partial t} = 0,
$$

provided that

$$\frac{d}{dt} \int_K \kappa \psi \, dx dy = -\int_{\partial K} \kappa \tilde{\mathbf{U}}_0 \cdot \mathbf{n} \psi \, ds + \int_K \kappa \tilde{\mathbf{U}}_0 \cdot \nabla \psi \, dx dy, \tag{2.22}$$

$$\frac{d}{dt} \int_K \chi \psi \, dx dy = -\int_{\partial K} \chi \tilde{\mathbf{U}}_0 \cdot \mathbf{n} \psi \, ds + \int_K \chi \tilde{\mathbf{U}}_0 \cdot \nabla \psi \, dx dy. \tag{2.23}$$

In other words, the scheme (2.8) also preserves constant pressure if the above two conditions are satisfied.

We notice from (2.3) and (2.4) that both κ and χ are linear functions of Y. Thus, (2.22) and (2.23) can be satisfied if the following equation for Y holds,

$$\frac{d}{dt} \int_K Y_h \psi \, dx dy = -\int_{\partial K} Y_h \tilde{\mathbf{U}}_0 \cdot \mathbf{n} \psi \, ds + \int_K Y_h \tilde{\mathbf{U}}_0 \cdot \nabla \psi \, dx dy. \tag{2.24}$$

It is not difficult to see that the above equation is actually a DG discretization of the equation

$$Y_t + (UY)_x + (VY)_y = 0.$$

Since the species equation (cf. (2.1)) can be written as

$$Y_t + UY_x + VY_y = Y_t + (UY)_x + (VY)_y - YU_x - YV_y = 0,$$

we obtain a DG discretization for the species equation as

$$\frac{d}{dt} \int_K Y_h \psi \, dx dy = -\int_{\partial K} Y_h (\mathbf{U} - \dot{X}) \cdot \mathbf{n} \psi \, ds + \int_K Y_h (\mathbf{U} - \dot{X}) \cdot \nabla \psi \, dx dy$$
$$+ Y_h(\mathbf{x}_b) \left(\int_{\partial K} \mathbf{U} \cdot \mathbf{n} \psi \, ds - \int_K \mathbf{U} \cdot \nabla \psi \, dx dy \right), \quad \forall K \in \mathscr{T}_h, \ \psi \in V_h^k \tag{2.25}$$

where $\mathbf{x}_b$ is the barycenter of the element K. Noticing that the above equation reduces to (2.24) for a solution with constant velocity and constant pressure, we know that (2.25), along with (2.8) (or (2.9)), preserves constant velocity and constant pressure. Moreover, (2.25) is only quasi-conservative since the last term is non-conservative.

The preservation of constant velocity and pressure by the DG-ALE scheme implies its preservation of a uniform flow or some geometric identities. The later is commonly called the geometric conservation law(s) (GCL) [33]. Satisfaction of GCL is considered as an important property for moving mesh methods [33]. By design, the above described DG-ALE scheme satisfies GCL.

2.4 Temporal Discretization and Limiting

Finally, the semi-discrete schemes (2.9) and (2.25) are discretized in time. Here, we use an explicit, the third order TVD Runge-Kutta scheme [51]. Casting (2.9) and (2.25) in the form

$$\frac{d(\mathbf{M}\mathbf{U}_h)}{dt} = L_h(\mathbf{U}_h, t),$$

where $\mathbf{M}$ is the mass matrix. Noticing that the mesh motion of the method is part of Lagrangian solution, we use the same temporal discretization to update the physical solution, the mesh coordinates, and other geometrical quantities such as the cell volume and the edge length. The scheme reads as

Stage 1

$$
\begin{aligned}
\mathbf{x}_j^{(1)} &= \mathbf{x}_j^n + \Delta t_n \dot{X}_j, \\
\mathbf{M}^{(1)}\mathbf{U}_h^{(1)} &= \mathbf{M}^n \mathbf{U}_h^n + \Delta t_n L_h(\mathbf{U}_h^n, t_n).
\end{aligned}
\tag{2.26}
$$

Stage 2

$$
\begin{aligned}
\mathbf{x}_j^{(2)} &= \frac{3}{4}\mathbf{x}_j^n + \frac{1}{4}(\mathbf{x}_j^{(1)} + \Delta t_n \dot{X}_j), \\
\mathbf{M}^{(2)}\mathbf{U}_h^{(2)} &= \frac{3}{4}\mathbf{M}^n \mathbf{U}_h^n + \frac{1}{4}(\mathbf{M}^{(1)}\mathbf{U}_h^{(1)} + \Delta t_n L_h(\mathbf{U}_h^{(1)}, t_n + \Delta t_n)).
\end{aligned}
\tag{2.27}
$$

Stage 3

$$
\begin{aligned}
\mathbf{x}_j^{n+1} &= \frac{1}{3}\mathbf{x}_j^n + \frac{2}{3}(\mathbf{x}_j^{(2)} + \Delta t_n \dot{X}_j), \\
\mathbf{M}^{n+1}\mathbf{U}_h^{n+1} &= \frac{1}{3}\mathbf{M}^n \mathbf{U}_h^n + \frac{2}{3}\left(\mathbf{M}^{(2)}\mathbf{U}_h^{(2)} + \Delta t_n L_h(\mathbf{U}_h^{(2)}, t_n + \frac{1}{2}\Delta t_n)\right).
\end{aligned}
\tag{2.28}
$$

We need to use a nonlinear limiter to control spurious oscillations in the numerical solution when strong shocks are present. The process typically contains identification of troubled cells and modification of the DG approximations on those cells. Here, we employ the minmod-type TVB limiter [13–16,37,47] to detect troubled cells and use the multi-resolution WENO limiter developed in [62,63] for the limiting process. Moreover, we implement this limiting process to the primitive variables component-wisely to keep the pressure non-oscillatory. Since the process is the same for schemes on a fixed mesh, we omit the detail here and refer the reader to [38].

The application of the forward Euler scheme to the continuity equation (2.19) gives

$$
\left(\int_K \rho\psi \, dxdy\right)^{n+1} = \int_K \rho\psi \, dxdy - \Delta t \int_{\partial K} \rho\tilde{\mathbf{U}}_0 \cdot \mathbf{n}\psi \, ds + \Delta t \int_K \rho\tilde{\mathbf{U}}_0 \cdot \nabla\psi \, dxdy,
\tag{2.29}
$$

where the superscripts are omitted. Similarly, for the momentum equations we have

$$
\begin{aligned}
\left(\int_K \rho U\psi \, dxdy\right)^{n+1} &= \int_K \rho U_0 \psi \, dxdy - \Delta t \int_{\partial K}(\rho U_0 \tilde{\mathbf{U}}_0 \cdot \mathbf{n} + P_0 n_x)\psi \, ds \\
&\quad + \Delta t \int_K (\rho U_0 \tilde{\mathbf{U}}_0 \cdot \nabla\psi + P_0 \psi_x) dxdy, \\
\left(\int_K \rho V\psi \, dxdy\right)^{n+1} &= \int_K \rho V_0 \psi \, dxdy - \Delta t \int_{\partial K}(\rho V_0 \tilde{\mathbf{U}}_0 \cdot \mathbf{n} + P_0 n_y)\psi \, ds \\
&\quad + \Delta t \int_K (\rho V_0 \tilde{\mathbf{U}}_0 \cdot \nabla\psi + P_0 \psi_y) dxdy.
\end{aligned}
$$

Thus, we get

$$
\left(\int_K \rho U\psi \, dxdy\right)^{n+1} = U_0 \left(\int_K \rho\psi \, dxdy\right)^{n+1},
\tag{2.30}
$$

$$
\left(\int_K \rho V\psi \, dxdy\right)^{n+1} = V_0 \left(\int_K \rho\psi \, dxdy\right)^{n+1}.
\tag{2.31}
$$

From the arbitrariness of ψ and the positivity of ρ, we have $U^{n+1} = U_0$ and $V^{n+1} = V_0$. Similarly, we get $P^{n+1} = P_0$. Therefore, we have the following remarks.

Remark 1 When the forward Euler scheme is employed for the time discretization, then the DG-ALE scheme preserves constant velocity and constant pressure.

Remark 2 When an explicit third-order TVD Runge-Kutta scheme is employed for the time discretization, the DG-ALE scheme preserves constant velocity and constant pressure. This is because an explicit third-order Runge-Kutta scheme can be expressed as a linear combination of the forward Euler scheme.

Remark 3 the DG-ALE scheme with an explicit third-order TVD Runge-Kutta scheme preserves a uniform flow and thus satisfies the GCL.

3 The Computation of the Grid Velocity

We consider the computation of the grid velocity in this section. To take the advantages of both the Lagrangian meshing and the moving mesh method, we employ a predictor-corrector strategy to define the grid velocity. More specifically, we first compute the Lagrangian velocity and obtain a Lagrangian mesh (cf. Sect. 3.1). Then we use the MMPDE method [24–26,37] to improve the quality of the Lagrangian mesh. The MMPDE method is described in Sect. 3.2.

3.1 The Lagrangian Meshing

There are many possible ways to determine the vertex velocity. Here, we adopt a simple yet robust least-squares algorithm [2,7]. The first step of the algorithm is to find the flow velocity U_e^* for each edge e. This has been done in [2,7] by solving a Riemann problem in the normal direction of the edge. Here we propose to use the idea of the NOK flux (cf. Sect. 2.2) to compute U_e^*. Specifically, from (2.15) and (2.14) (and taking $\dot{X} = 0$) we compute U_e^* as

$$U_e^* = \langle u^1 \rangle_+ + \langle u^1 \rangle_-, \tag{3.1}$$

where

$$\langle u^1 \rangle_+ = (\mathbf{U}_L \cdot \mathbf{n})\langle u^0 \rangle_+ + \frac{1}{2}\frac{e^{-\lambda(\mathbf{U}_L \cdot \mathbf{n})^2}}{\sqrt{\pi\lambda}}, \qquad \langle u^1 \rangle_- = (\mathbf{U}_R \cdot \mathbf{n})\langle u^0 \rangle_- - \frac{1}{2}\frac{e^{-\lambda(\mathbf{U}_R \cdot \mathbf{n})^2}}{\sqrt{\pi\lambda}}, \tag{3.2}$$

$$\langle u^0 \rangle_+ = \frac{1}{2}\mathrm{erfc}(-\sqrt{\lambda}(\mathbf{U}_L \cdot \mathbf{n})), \qquad \langle u^0 \rangle_- = \frac{1}{2}\mathrm{erfc}(\sqrt{\lambda}(\mathbf{U}_R \cdot \mathbf{n})), \tag{3.3}$$

$\lambda = \min\{\frac{1}{c_L^2}, \frac{1}{c_R^2}\}$, c is the sound speed, and $\mathbf{U}_R$ and $\mathbf{U}_L$ denote the values of $\mathbf{U}$ from the positive-$\mathbf{n}$ side and the negative-$\mathbf{n}$ side of e, respectively. Note that the velocity U_e^* is defined at the midpoint of the edge.

The second step is to obtain the velocity at each vertex using the least squares method. Specifically, for a given vertex j, we would like the normal projection of the vertex velocity on any of the connecting edges to be equal to the corresponding Riemann velocity, viz.,

$$\dot{X}_j^L \cdot \mathbf{n}_{e_i} = U_{e_i}^*, \quad i = 1, 2, \cdots, n_j$$

where n_j denotes the number of the edges connected to the vertex j. The system can be overdetermined or underdetermined and thus we solve it by minimizing a weighted sum of the squares of the residuals with respect to $\dot{X}_j^L$,

$$\sum_i \alpha_{e_i} (\dot{X}_j^L \cdot \mathbf{n}_{e_i} - U_{e_i}^*)^2,$$

where α_{e_i} is a weight taken as the mean of the density on each side of the edge in our computation. Having computed the Lagrangian velocity for all vertices (with proper modifications for boundary vertices so that they stay on the domain), we can obtain the Lagrangian mesh $\mathscr{T}_h^{n+1,L}$ by updating the location of the vertices as

$$\mathbf{x}_j^L = \mathbf{x}_j^n + \Delta t_n \cdot \dot{X}_j^L, \quad j = 1, ..., N_v.$$

3.2 The MMPDE Moving Mesh Method

Generally speaking, a Lagrangian mesh can be severely distorted and thus cannot be used directly for the numerical solution of PDEs. In this subsection we describe the MMPDE method that is employed to improve the quality of Lagrangian meshes.

To start with, we assume a reference computational mesh $\hat{\mathscr{T}}_c = \{\hat{\boldsymbol{\xi}}_j\}_{j=1}^{N_v}$ has been chosen, which is a deformation of the physical mesh and is fixed for the whole computation. Usually we should choose it to be as uniform as possible and can take it as the initial physical mesh. We also need to use the computational mesh $\mathscr{T}_c = \{\boldsymbol{\xi}_j\}_{j=1}^{N_v}$, which is also a deformation of the physical mesh and will be used as an intermediate variable.

The MMPDE method views any nonuniform mesh as a uniform one in some metric specified by a metric tensor $\mathbb{M} = \mathbb{M}(\mathbf{x})$ that is a symmetric and positive definite matrix for each $\mathbf{x}$ and uniformly positive definite on Ω. Such a uniform mesh $\mathscr{T}_h$ in $\mathbb{M}$ satisfies the so-called equidistribution and alignment conditions [27]

$$|K|\sqrt{\det(\mathbb{M}_K)} = \frac{\sigma_h |K_c|}{|\Omega_c|}, \qquad \forall K \in \mathscr{T}_h \tag{3.4}$$

$$\frac{1}{d}\text{tr}((F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T}) = \det((F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T})^{\frac{1}{d}}, \qquad \forall K \in \mathscr{T}_h \tag{3.5}$$

where d is the dimension of the spatial domain ($d = 1$ for one dimension and $d = 2$ for two dimensions), K_c is the element in $\mathscr{T}_c$ corresponding to K, F_K is the affine mapping from K_c to K and F_K' is its Jacobian matrix, $\mathbb{M}_K$ is the average of $\mathbb{M}$ over K, $\text{tr}(\cdot)$ and $\det(\cdot)$ denote the trace and determinant of a matrix, respectively, and

$$|\Omega_c| = \sum_{K_c \in \mathscr{T}_c} |K_c|, \quad \sigma_h = \sum_{K \in \mathscr{T}_h} |K|\det(\mathbb{M}_K)^{\frac{1}{2}}.$$

The equidistribution condition (3.4) requires that all of the elements have the same size. It determines the size of elements through the metric tensor $\mathbb{M}$: $|K|$ is smaller when $\det(\mathbb{M}_K)$ is larger and vice versa. On the other hand, the alignment condition (3.5) requires that element K, measured in the metric $\mathbb{M}_K$, be similar to K_c. It determines the shape and orientation of K through the combination of $\mathbb{M}_K$ and the shape and orientation of K_c. In the case where K_c is equilateral, the shape and orientation of K is then determined completely by $\mathbb{M}_K$ (through its eigenvalues and eigen-directions).

The objective of the MMPDE method is to generate a mesh satisfying the above two conditions as closely as possible. This is done by minimizing the energy function

$$I_h(\mathscr{T}_h, \mathscr{T}_c) = \sum_{K \in \mathscr{T}_h} |K| \sqrt{\det(\mathbb{M}_K)} (\text{tr}((F_K')^{-1} \mathbb{M}_K^{-1} (F_K')^{-T}))^{\frac{3d}{4}}$$

$$+ d^{\frac{3d}{4}} \sum_{K \in \mathscr{T}_h} |K| \sqrt{\det(\mathbb{M}_K)} \left(\frac{|K_c|}{|K| \sqrt{\det(\mathbb{M}_K)}} \right)^{\frac{3}{2}}, \qquad (3.6)$$

which is a Riemann sum of a continuous functional [28] based on equidistribution and alignment for variational mesh adaptation. Here we use the ξ-formulation where $\mathscr{T}_h$ is taken as a currently available physical mesh (denoted by $\tilde{\mathscr{T}}_h$, its choice will be discussed later) and $I_h(\tilde{\mathscr{T}}_h, \mathscr{T}_c)$ is minimized with respect to $\{\boldsymbol{\xi}_j\}$. The minimization is realized by solving the gradient system (i.e., the mesh equation)

$$\frac{d\boldsymbol{\xi}_j}{dt} = -\frac{\det(\mathbb{M}(\mathbf{x}_j))^{\frac{1}{4}}}{\tau} \left(\frac{\partial I_h}{\partial \boldsymbol{\xi}_j} \right)^T, \quad j = 1, 2, \cdots, N_v \qquad (3.7)$$

where $\frac{\partial I_h}{\partial \boldsymbol{\xi}_j}$ is considered as a row vector and $\tau > 0$ is a parameter used to adjust the response time of mesh movement to changes in $\mathbb{M}$. The analytical formula for $\frac{\partial I_h}{\partial \boldsymbol{\xi}_j}$ can be found in [22].

The mesh equation (3.7) (with proper modifications for boundary vertices) can be integrated from t_n to t_{n+1}, starting with the reference computational mesh $\hat{\mathscr{T}}_c$ as an initial mesh, to obtain a new computational mesh $\mathscr{T}_c^{n+1}$. This new mesh forms a correspondence Ψ_h with the physical mesh $\tilde{\mathscr{T}}_h$ having the property $\tilde{\mathbf{x}}_j = \Psi_h(\boldsymbol{\xi}_j^{n+1})$, $j = 1, 2, \cdots, N_v$. The new physical mesh $\mathscr{T}_h^{n+1}$ is then defined as $\mathbf{x}_j^{n+1} = \Psi_h(\hat{\boldsymbol{\xi}}_j)$, $j = 1, 2, \cdots, N_v$, which can be computed using linear interpolation.

In the context of the MMPDE method, it is common to take $\tilde{\mathscr{T}}_h$ as the physical mesh at the current step, i.e., $\tilde{\mathscr{T}}_h = \mathscr{T}_h^n$. In our current situation, however, we choose $\tilde{\mathscr{T}}_h = \mathscr{T}_h^{n+1,L}$ to take the advantages of the Lagrangian mesh $\mathscr{T}_h^{n+1,L}$. In this setting, the MMPDE method plays a role of improving the quality of the Lagrangian mesh. It has been shown in [23] analytically and numerically that the x-formulation of the MMPDE method, where $I_h(\mathscr{T}_h, \hat{\mathscr{T}}_c)$ (with $\mathscr{T}_c = \hat{\mathscr{T}}_c$) is minimized by solving its gradient system with respect to $\{\mathbf{x}_j\}$, produces non-singular (tangling-free) meshes when the metric tensor is bounded and the initial mesh is non-singular. Although the ξ-formulation (that we use in this work) is different from the x-formulation, they are numerical approximations to the same continuous optimization problem and thus we can expect the ξ-formulation also leads to non-singular meshes especially when the mesh is sufficiently fine. Indeed, our experience shows that the ξ-formulation is as robust as the x-formulation in terms of generating non-singular meshes. Moreover, with a proper metric tensor (see its definition below), the MMPDE method can be expected to concentrate mesh points near regions of shocks and material interfaces. Thus, we hope that the use of the MMPDE method to improve quality of the Lagrangian mesh will not (at least not significantly) move points away from shocks and material interfaces. In this sense, the MMPDE maintains a level of mesh concentration of the Lagrangian mesh.

Having obtained the new physical mesh $\mathscr{T}_h^{n+1}$, we can compute the nodal grid velocity using (2.6) and then $\dot{X} = (U_g, V_g)$ as the piecewise linear interpolant of the nodal grid velocity.

A key to the success for the MMPDE moving mesh method is the selection of a proper metric tension $\mathbb{M}$ that provides the information needed to control the size, shape and orien-

tation of the mesh elements throughout the domain. In this work, we employ a metric tensor that is known to be optimal for the L^2 norm of linear interpolation error [29] and has been used for conservation laws in [37], i.e.,

$$\mathbb{M} = \det(\mathbb{I} + |H(u_h^n)|)^{-\frac{1}{d+4}} (\mathbb{I} + |H(u_h^n)|), \tag{3.8}$$

where u denotes a physical variable, $\mathbb{I}$ is the $d \times d$ identity matrix, $H(u_h^n)$ is a recovered Hessian from the numerical solution u_h^n, and $|H(u_h^n)| = Q\mathrm{diag}(|\lambda_1|, \cdots, |\lambda_d|)Q^T$ with $Q\mathrm{diag}(\lambda_1, \cdots, \lambda_d)Q^T$ being the eigen-decomposition of $H(u_h^n)$. The Hessian is recovered using the quadratic least squares fitting [59]. For a smoother mesh, a low-pass-filter smoothing algorithm [26] is applied to the metric tensor several sweeps every time it is computed.

For multi-component flows, we take u to be the quantity

$$S = 1 + \beta_1 \left(\frac{\rho}{\|\rho\|_\infty} \right)^2 + \beta_2 \left(\frac{P}{\|P\|_\infty} \right)^2 + \beta_3 \left(\frac{Y}{\|Y\|_\infty} \right)^2, \tag{3.9}$$

where β_i, $i = 1, 2, 3$ are positive parameters and $\| \cdot \|_\infty$ is the maximum norm. In the computation of the metric tensors, the nodal values of ρ, P, and Y are needed. They are calculated as the volume average of the cell average of these variables in the neighboring cells of each vertex.

4 Numerical Examples

In this section we present numerical results obtained with the DG-ALE method described in the previous sections for a selection of one- and two-dimensional examples. Recall that the method has been described in two dimensions. Its implementation in one dimension is similar. The CFL number in time step selection is set to be 0.3 for P^1 elements, 0.15 for P^2 elements. The parameter τ in (3.7) is taken as 0.1 for accuracy test problems, 10^{-3} and 10^{-4} for one- and two-dimensional systems with discontinuities, respectively. The parameter β_i's in (3.9) are taken as 1.0, 1.0, and 0 for all the computations unless otherwise stated. The results obtained with the DG-ALE method will be marked as "ALEMM" in figures.

4.1 One-dimensional Examples

Example 1 To verify the convergence order of the new method, we first consider a sine wave composed of two components with $\gamma_1 = 1.4$, $\gamma_2 = 1.9$, $B_1 = 1$, and $B_2 = 0$. The initial condition is given by

$$\rho(x, 0) = 1 + 0.2\sin(\pi x), \quad v(x, 0) = 1, \quad P(x, 0) = 1, \quad Y(x, 0) = 0.5 + 0.5\sin(\pi x).$$

A periodic boundary condition is used. We take the computational domain as (0, 2) and compute the solution up to $t = 0.5$.

The error of the density is listed in Table 1, which shows the second-order convergence for P^1 elements and the third-order convergence for P^2 elements for the DG-ALE method.

Example 2 To demonstrate the non-oscillatory property for the pressure and velocity fields, in this example we consider the interface only problem with the initial condition

$$(\rho, v, P, \gamma, B) = \begin{cases} (1, 1, 1, 1.4, 1), & x \le 0 \\ (0.125, 1, 1, 1.9, 0), & x > 0. \end{cases}$$

Table 1 Example 1: The solution error in the density at $t = 0.5$

k	N	40	80	160	320	640	1280
1	L^1	1.128e-4	2.826e-5	7.150e-6	1.818e-6	4.615e-7	1.166e-7
	Order		1.997	1.983	1.975	1.978	1.985
	L^2	1.443e-4	3.636e-5	9.296e-6	2.388e-6	6.111e-7	1.552e-7
	Order		1.989	1.968	1.961	1.966	1.977
	L_∞	4.521e-4	1.125e-4	2.925e-5	7.592e-6	1.938e-6	4.889e-7
	Order		2.006	1.944	1.946	1.970	1.987
2	L^1	9.595e-6	1.225e-6	1.498e-7	1.801e-8	2.168e-9	2.640e-10
	Order		2.969	3.032	3.056	3.054	3.038
	L^2	1.380e-5	1.784e-6	2.175e-7	2.579e-8	3.049e-9	3.656e-10
	Order		2.951	3.036	3.076	3.080	3.060
	L_∞	5.748e-5	7.204e-6	9.491e-7	1.167e-7	1.347e-8	1.524e-9
	Order		2.996	2.924	3.023	3.115	3.144

The computational domain is taken as (-5,5) and the inflow/outflow boundary conditions are employed at the ends.

The numerical results obtained with $N = 100$ at $T = 2$ are plotted in Fig. 1. The figure shows that the DG-ALE method preserves the constant velocity and pressure and appears to be free of oscillations near the material interface. Moreover, the results show that the method is able to concentrate mesh points in the regions of the material interface, which is a strong indication that the incorporation of the Lagrangian meshing with the MMPDE moving mesh method works well. Furthermore, the close-up of the density at the interface demonstrates that P^2-DG gives more accurate results than P^1-DG.

To show the method's preservation of the constant velocity and pressure, the absolute errors of the velocity and pressure are plotted in Fig. 2, where one can observe that all the errors are close to the machine zero.

Example 3 To test the numerical viscosity of the method, the Shu-Osher problem, which was first introduced for a single material in [52], is considered for two components [35] in this example (where each of γ and B takes two different values on the domain). This problem contains both shocks and complex smooth region structures. We solve the Euler equations with a moving shock interacting with a sine wave in density. The initial data is

$$(\rho, v, P, \gamma, B) = \begin{cases} (3.857143, 2.629369, \frac{31}{3}, 1.4, 1), & x \leq -4 \\ (1 + 0.2\sin(5x), 0, 1, 1.9, 0), & x > -4. \end{cases}$$

The inflow and outflow boundary conditions are employed at the ends. The physical domain is taken as (-5, 5).

The computed density obtained with $N = 150$ at $t = 1.8$ is plotted in Fig. 3 where the reference solution is obtained by the second-order DG method with 2000 uniform points. The figure shows that both the shocks and the complex structures in the smooth regions are resolved well by the DG-ALE scheme while P^2-DG is more accurate than P^1-DG. The mesh trajectories Fig. 3 also show that the mesh points are concentrated near the moving shocks and smooth region structures for all time in the time range of the computation.
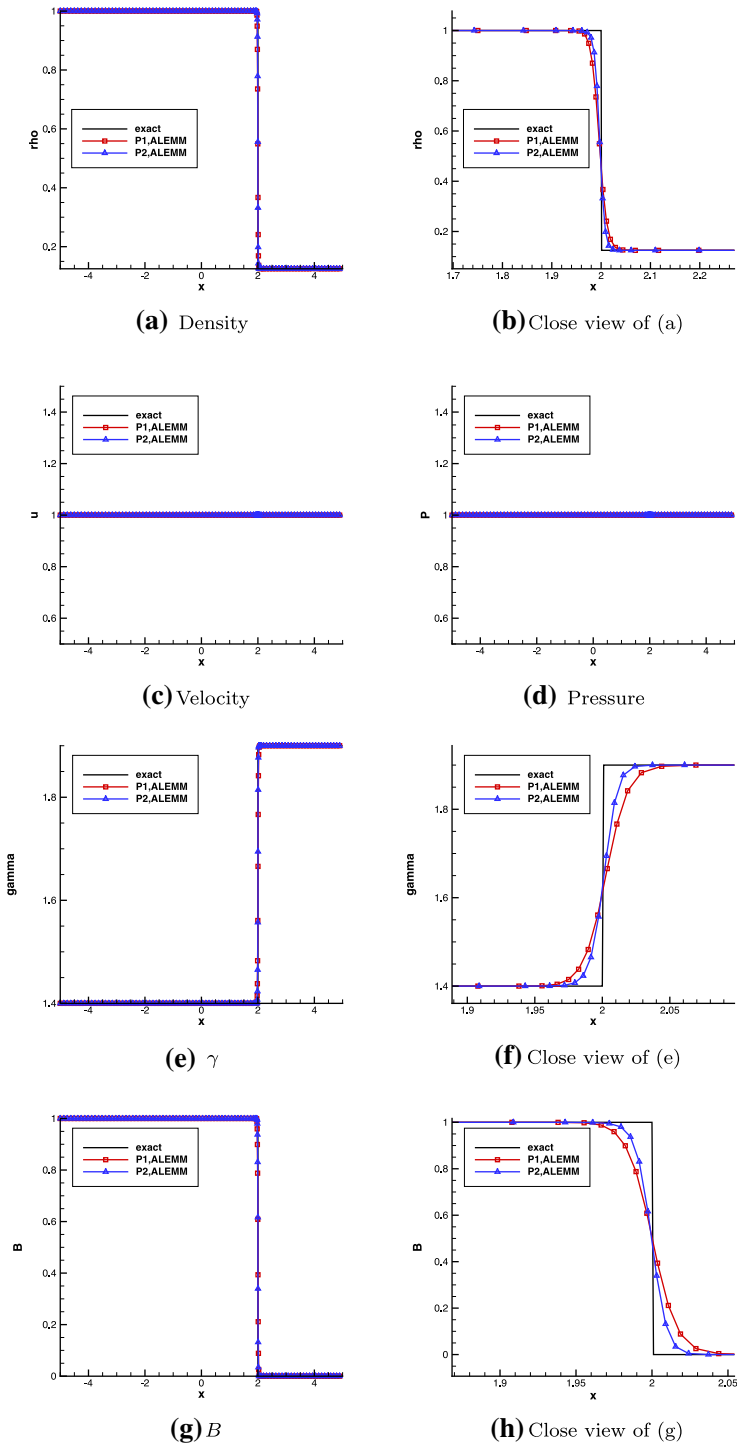
(a) Density

(b) Close view of (a)

(c) Velocity

(d) Pressure

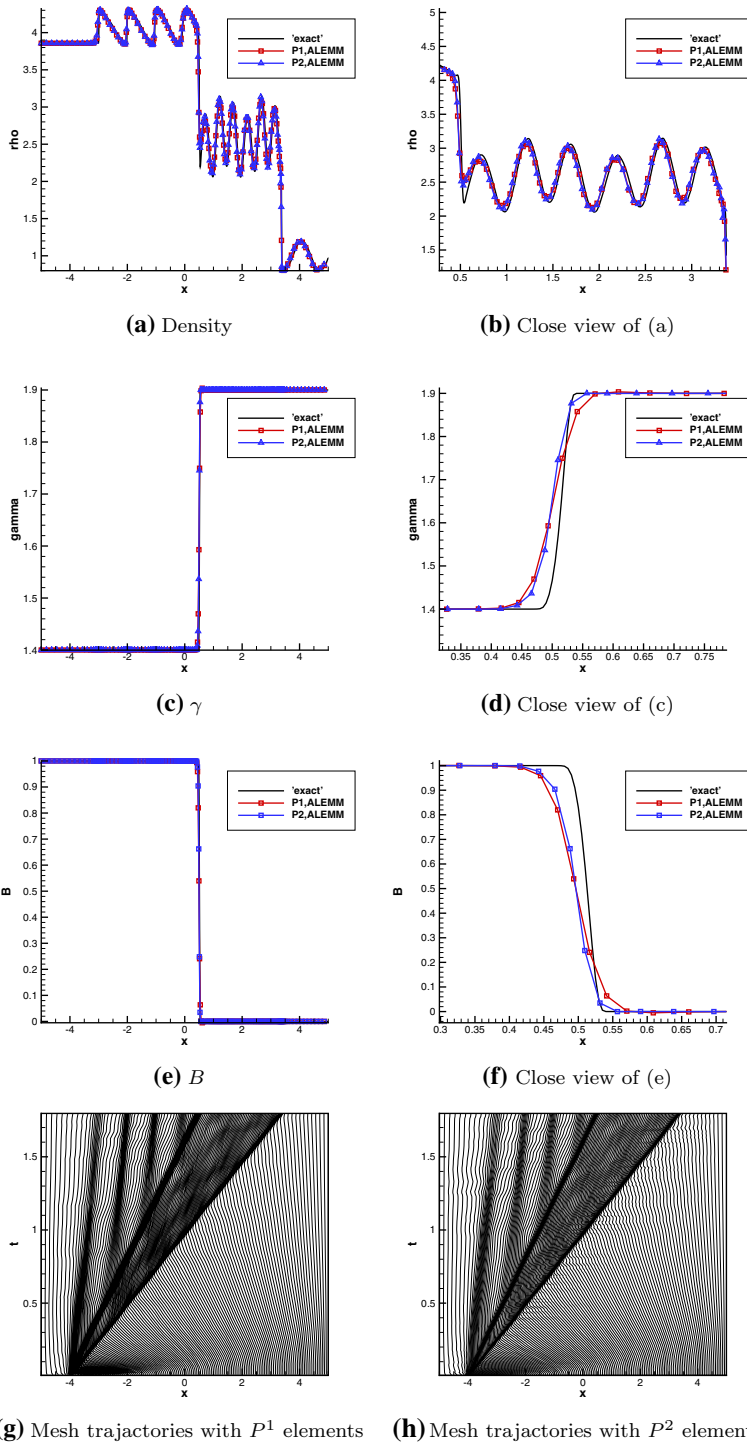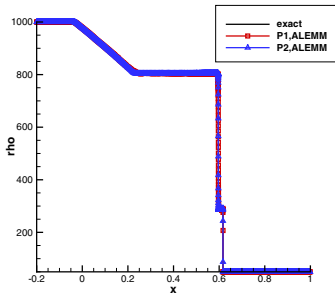**(e)** γ

(f) Close view of (e)

(g) B

(h) Close view of (g)

Fig. 1 Example 2 The DG-ALE method with $N = 100$

(a) error of velocity

(b) error of pressure

(c) Mesh trajactories with P^1 elements

(d) Mesh trajactories with P^2 elements
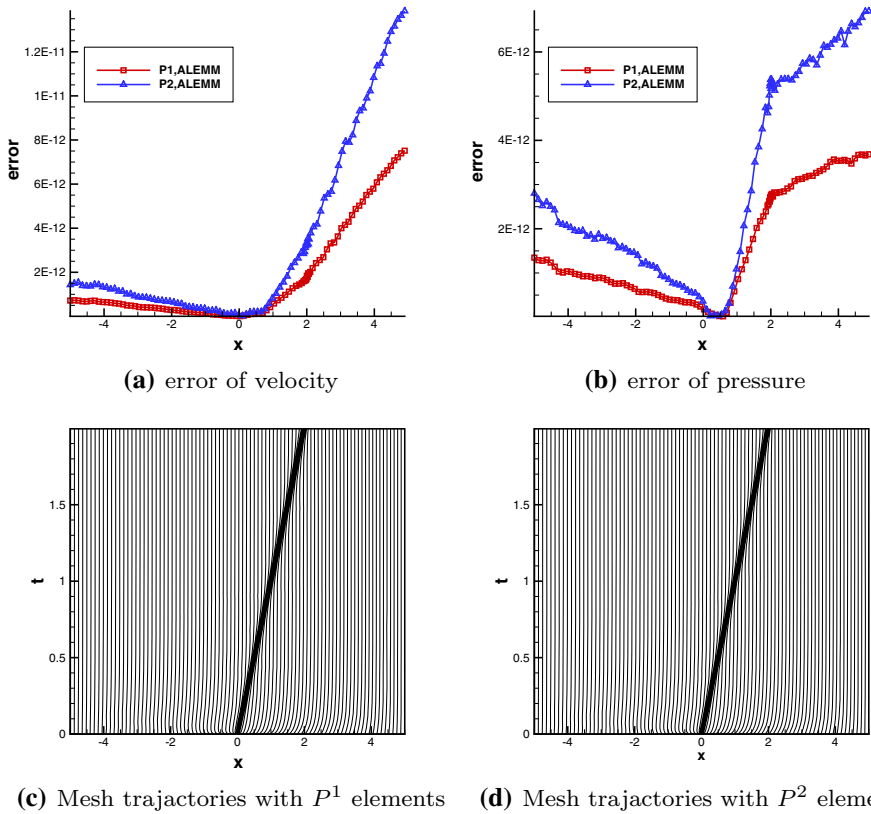
Fig. 2 Example 2 The DG-ALE method with $N = 100$

Example 4 In this example the gas-liquid shock tube test with a strong shock wave is considered. This test is challenging since the shock and the material interface are close to each other and the pressure ratio is excessively high. The initial condition is

$$(\rho, v, P, \gamma, B) = \begin{cases} (10^3, 0, 10^9, 4.4, 6 \times 10^8), & x \leq 0.5 \\ (50, 0, 10^5, 1.4, 0), & x > 0.5 \end{cases}$$

and the computational domain is (-0.2, 1). The inflow and outflow boundary conditions are employed at the ends. The final time is $t = 0.0002$. For this problem, an analytical solution is available [17].

The numerical results with $N = 2000$ are plotted in Fig. 4. One can see that both the shock and the material interface are resolved by the scheme. Moreover, P^2-DG produces more accurate solutions than P^1-DG.

For comparison purpose, in Figs. 5, 6 and 7 plots are shown the results of the current scheme against those with the pure Lagrangian DG method (marked by "LAG"), an ALE with Winslow smoothing (marked using "ALEWIN") where the Winslow method is used to smooth and improve the Lagrangian mesh, and the moving mesh DG method (marked by "MM") which basically is the DG-ALE method without using the Lagrangian mesh, respectively. From these figures, we can see that the current DG-ALE method gives more accurate results than ALEWIN, MM, and LAG. Once again, this demonstrates that the incorporation of the

(a) Density

(b) Close view of (a)

(c) γ

(d) Close view of (c)

(e) B

(f) Close view of (e)

(g) Mesh trajactories with P^1 elements   **(h)** Mesh trajactories with P^2 elements

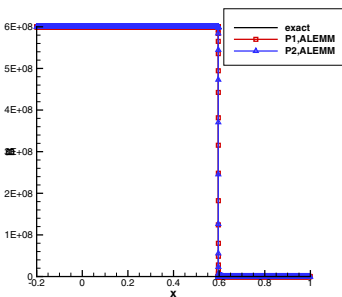**Fig. 3** Example 3 The DG-ALE method with $N = 150$

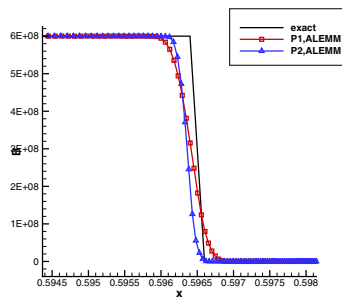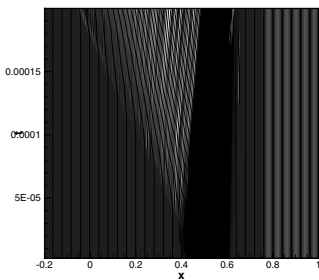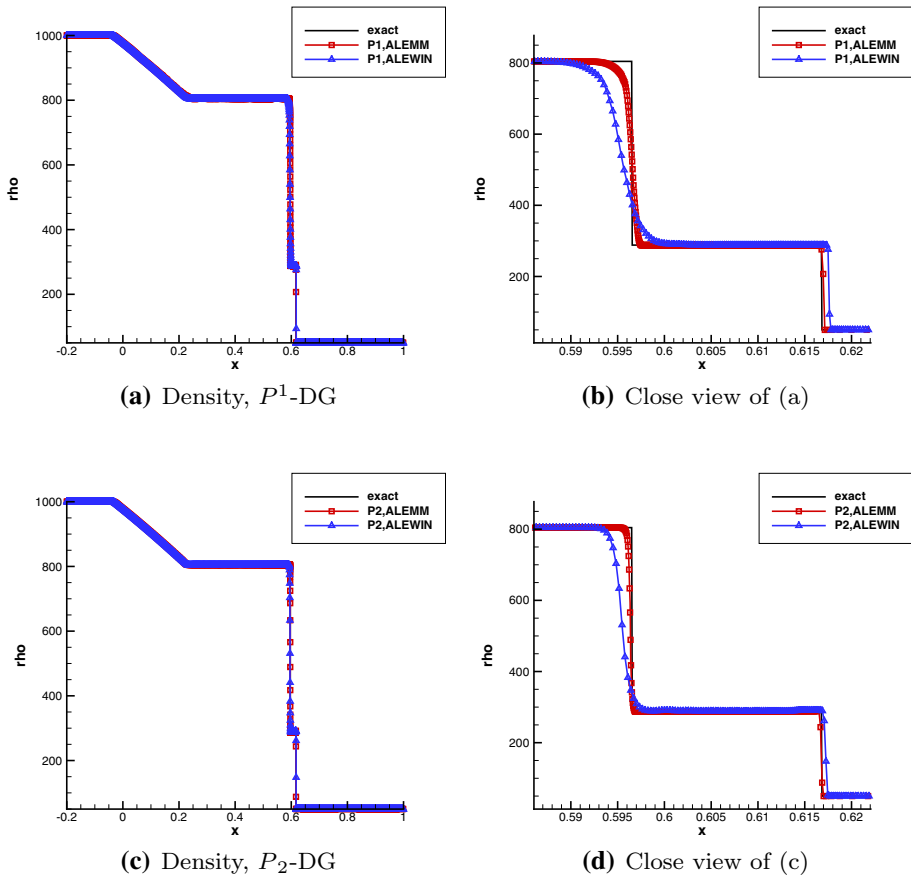**(a)** Density
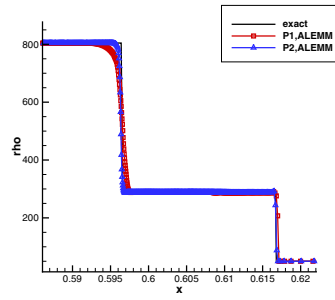
(b) Close view of (a)
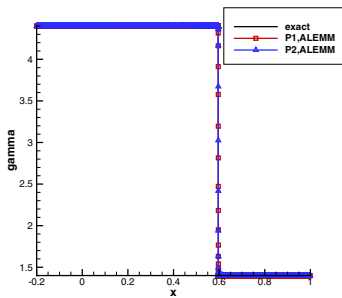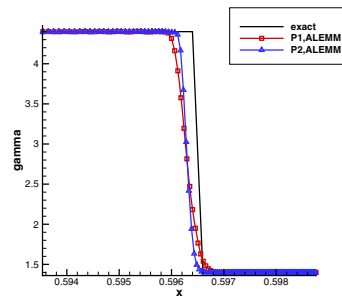
(c) γ

(d) Close view of (c)

(e) B

(f) Close view of (e)

(g) Mesh trajactories with P^1 elements
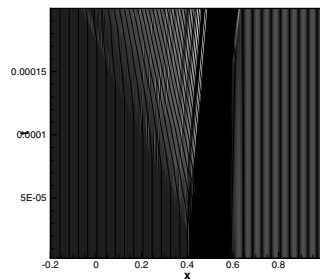
(h) Mesh trajactories with P^2 elements

Fig. 4 Example 4 The DG-ALE method with $N = 2000$

(a) Density, P^1-DG

(b) Close view of (a)

(c) Density, P_2-DG

(d) Close view of (c)

Fig. 5 Example 4 The DG-ALE method with moving mesh is compared with DG-ALE with Winslow smoothing. $N = 2000$

Lagrangian meshing with the MMPDE moving mesh method used in the current DG-ALE method works really well to concentrate mesh points in regions of shocks and material interfaces. Moreover, the solutions obtained with LAG are oscillatory at the discontinuities.

The solutions obtained for a longer time are shown in Fig. 8.

4.2 Two-dimensional Examples

For two-dimensional examples, an initial triangular mesh is obtained by dividing any rectangular element into four triangular elements; see Fig. 9 for an example. A moving mesh associated with a mesh in Fig. 9 will be denoted by $N = 10 \times 10 \times 4$.

Example 5 To verify the convergence order of the DG-ALE method in two dimensions, we consider a sine-wave problem and choose the parameters $\gamma_1 = 1.4$, $\gamma_2 = 1.9$, $B_1 = 1$, $B_2 = 0$. The initial condition is given by

$$\rho(x, y, 0) = 1 + 0.2\sin(\pi(x+y)), \quad U(x, y, 0) = 1, \quad V(x, y, 0) = 1,$$
$$P(x, y, 0) = 1, \quad Y(x, y, 0) = 0.5 + 0.5\sin(\pi(x+y)).$$

(a) Density, P^1-DG

(b) Close view of (a)

(c) Density, P^2-DG

(d) Close view of (c)
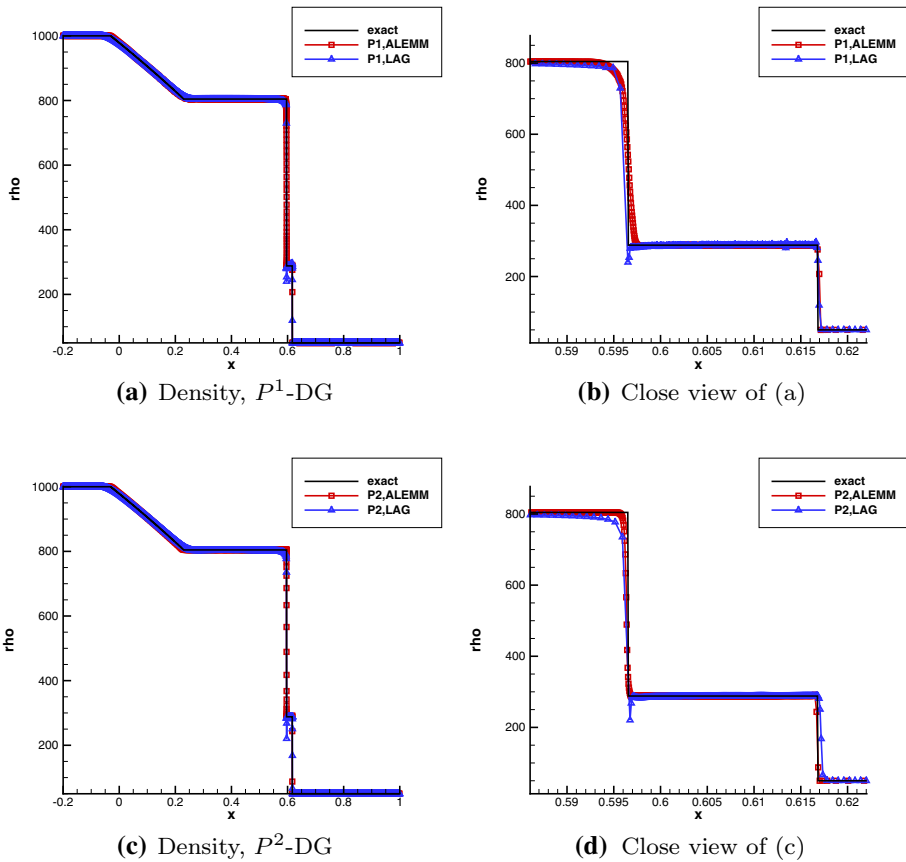
Fig. 6 Example 4 The DG-ALE method with moving mesh is compared with the Lagrangian DG method. $N = 2000$

Periodic boundary conditions are used in both directions. We take the computational domain as $(0, 2) \times (0, 2)$.

We compute the solution up to $t = 1$. The error in the density is listed in Table 2, which shows the second-order convergence for P^1 elements and the third-order convergence for P^2 elements for the DG-ALE method.

Example 6 We test a 2D interface only problem with the initial condition

$$(\rho, \mu, \nu, P, \gamma, B) = \begin{cases} (2, 1, 1, 1, 4.4, 1), & (x - 0.2)^2 + (y - 0.2)^2 \leq 0.01 \\ (1, 1, 1, 1, 1.4, 0), & (x - 0.2)^2 + (y - 0.2)^2 > 0.01. \end{cases}$$

For this example, nonreflecting boundary conditions are used for the top and bottom boundary and inflow and outflow boundary conditions are used for the left and right boundary. The computational domain is taken as $(0, 1) \times (0, 1)$ and the numerical computation is stopped at $t = 0.3$.

The results obtained with $N = 100 \times 100 \times 4$ are plotted in Fig. 10. One can observe that the quasi-conservative DG-ALE method can preserve the constant pressure and velocity and seems to be free of oscillations near the material interfaces. The density along the line
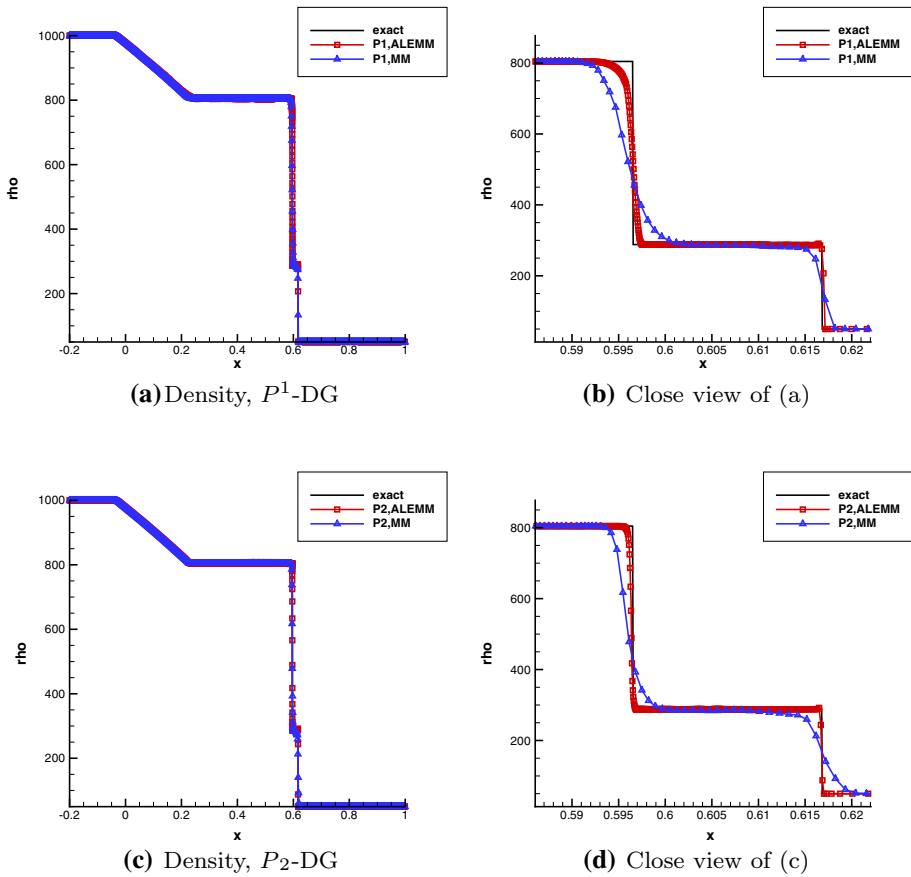
(a) Density, P^1-DG

(b) Close view of (a)

(c) Density, P_2-DG

(d) Close view of (c)

Fig. 7 Example 4 The DG-ALE method with moving mesh is compared with the moving mesh DG method (i.e., the DG-ALE method without using the Lagrangian mesh). $N = 2000$

$y = 0.5$ and the final adaptive meshes for both P^1-DG and P^2-DG are plotted in Fig. 11. It is clear that the mesh points are concentrated near the material interface and the results with P^2-DG have better resolution than the ones with P^1-DG.

The absolute errors of the velocity and pressure are plotted in Fig. 12, in which one can observe that all the errors are machine zeros.

Example 7 We consider an air shock impacting on a helium bubble [46]. The domain of this problem is $(-3, 4) \times (-3, 3)$ and the initial condition is

$$(\rho, U, V, P, \gamma, B) = \begin{cases} (0.138, 0, 0, 1, \frac{5}{3}, 0), & x^2 + y^2 \le 1 \\ (1.3764, 0.394, 0, 1.5698, 1.4, 0), & x < -1.2 \\ (1, 0, 0, 1, 1.4, 0), & x \ge -1.2 \text{ and } x^2 + y^2 > 1. \end{cases}$$
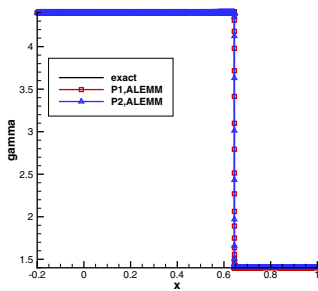
Reflective boundary conditions are used for the top and bottom boundary and inflow and outflow boundary conditions are adopted for the left and right boundary. The final time is $t = 4$. The parameters β_i's in (3.9) are all taken as 1 in this example.
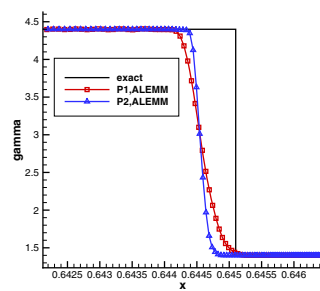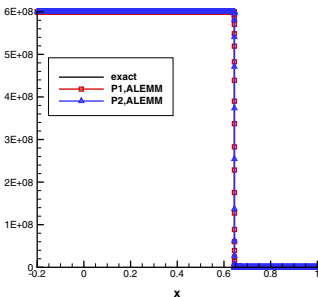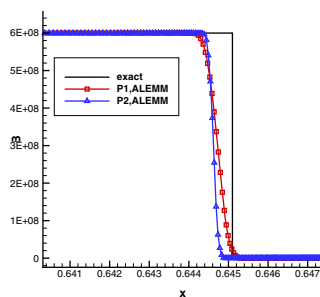
(a) Density
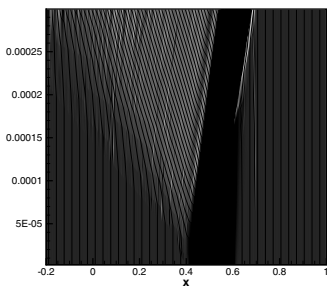
(b) Close view of (a)

(c) γ

(d) Close view of (c)
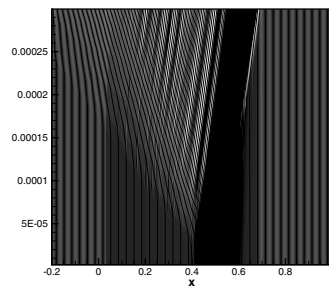
(e) B
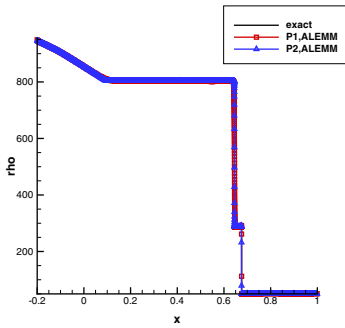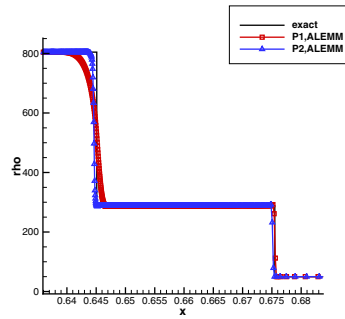
(f) Close view of (e)

(g) Mesh trajectories with P^1 elements   **(h)** Mesh trajectories with P^2 elements

Fig. 8 Example 4 The DG-ALE method with $N = 2000$

Fig. 9 An initial triangular mesh used in two dimensional computation. A moving mesh associated with this mesh is denoted by $N = 10 \times 10 \times 4$
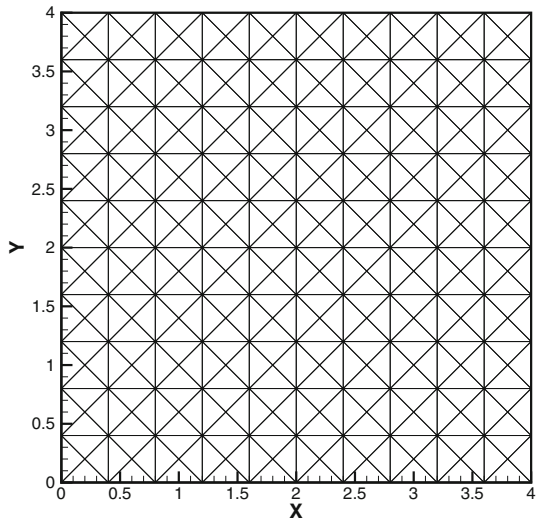
Table 2 Example 5: The error in the density at $t = 1$

k	N	$4 \times 4 \times 4$	$8 \times 8 \times 4$	$16 \times 16 \times 4$	$32 \times 32 \times 4$	$64 \times 64 \times 4$	$128 \times 128 \times 4$
1	L^1	1.498e-2	3.159e-3	7.358e-4	1.767e-4	4.454e-5	1.140e-5
	Order		2.245	2.102	2.058	1.988	1.966
	L^2	1.959e-2	4.149e-3	9.762e-4	2.325e-4	5.966e-5	1.553e-5
	Order		2.239	2.088	2.070	1.962	1.942
	L_∞	5.348e-2	1.540e-2	3.419e-3	8.166e-4	2.135e-4	5.421e-5
	Order		1.796	2.171	2.066	1.936	1.977
2	L^1	2.057e-3	3.360e-4	5.516e-5	1.112e-5	1.685e-6	1.876e-7
	Order		2.614	2.607	2.311	2.722	3.168
	L^2	2.709e-3	4.020e-4	6.957e-5	1.511e-5	2.468e-6	2.938e-7
	Order		2.752	2.531	2.203	2.614	3.070
	L_∞	8.688e-3	1.102e-3	2.707e-4	7.037e-5	1.163e-5	1.491e-6
	Order		2.980	2.025	1.944	2.597	2.963

We plot contours of the density and the adaptive mesh of $N = 70 \times 60 \times 4$ at $t = 0.5$, 1, 2, and 4 in Figs. 13 and 14. One can see that the mesh points are concentrated at the material interface, an indication that the combination of the Lagrangian meshing and the MMPDE moving mesh method works well. Moreover, P^2-DG produces sharper contours of the density near the material interface than P^1-DG.

Example 8 To demonstrate the performance of the current DG-ALE method with high pressure ratio in two dimensions, we consider a model underwater explosion problem [38,54]. In this test, the computation domain is taken as $(-2, 2) \times (-1.5, 1)$. Initially, the horizontal air-water interface is located at the $y = 0$ and the center of a circular gas bubble with the radius 0.12 in the water is located at $(0, -0.3)$. Above the air-water interface, the fluid is a
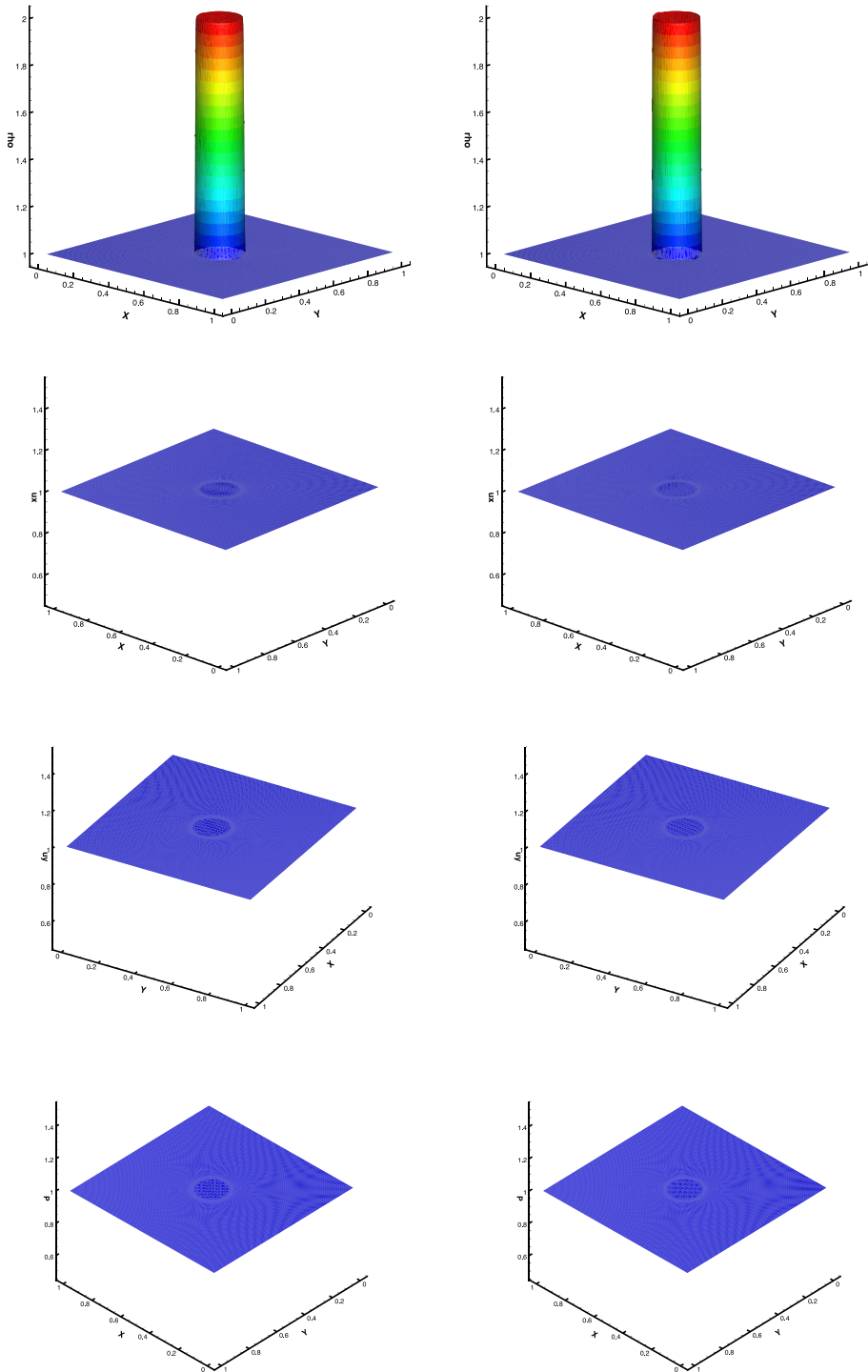
Fig. 10 Example 6. From top row to bottom row: density, velocity in the x-direction, velocity in the y-direction, and pressure. Left column: P^1-DG; Right column: P^2-DG. $N = 100 \times 100 \times 4$

(a) The density along the line $y = 0.5$

(b) Zoom of (a)

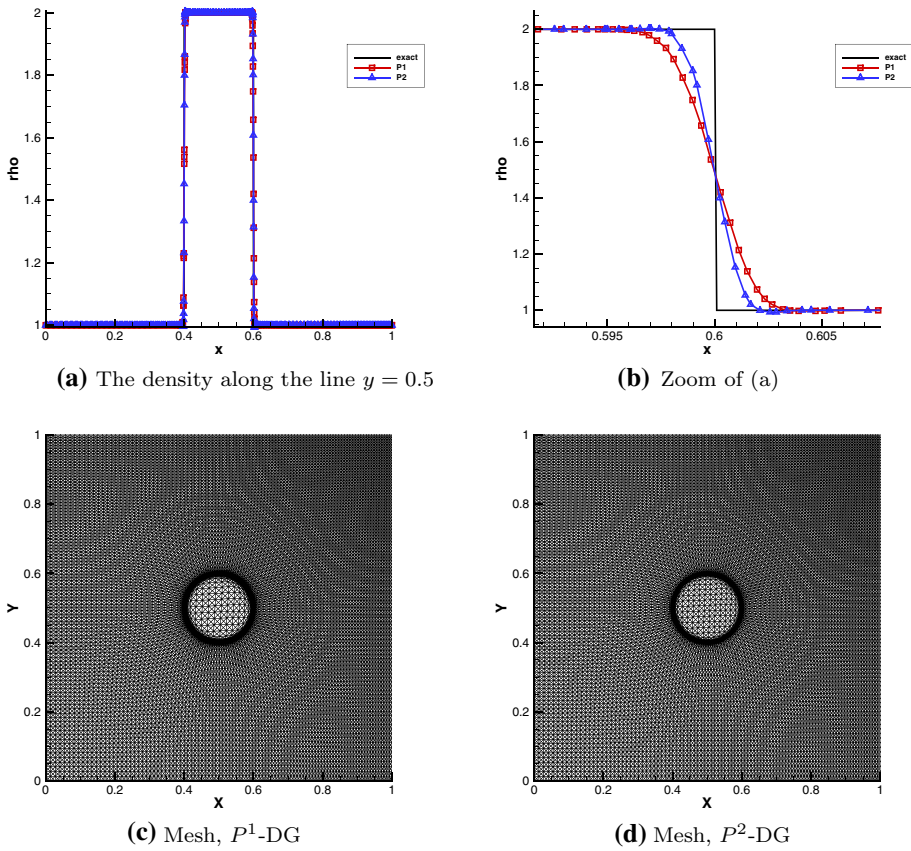**(c)** Mesh, P^1-DG

(d) Mesh, P^2-DG

Fig. 11 Example 6. The density along $y = 0.5$ and the final meshes of $N = 100 \times 100 \times 4$

perfect gas at the standard atmospheric condition and below the air-water interface in region outside the gas bubble the fluid is water. Thus the initial condition is

$$(\rho, U, V, P, \gamma, B) = \begin{cases} (1.225, 0, 0, 101325, 1.4, 0), & y > 0 \\ (1250, 0, 0, 10^9, 1.4, 0), & x^2 + (y + 0.3)^2 \leq 0.12^2 \text{ and } y \leq 0 \\ (1000, 0, 0, 101325, 4.4, 6 \times 10^8), & \text{otherwise.} \end{cases}$$

A reflecting boundary condition is employed on the bottom of the domain and non-reflecting boundary conditions are used for the remaining boundary. The parameters β_i's in (3.9) are all taken as 1 in this example.

At the beginning of the process, both the gas and water are in a stationary position. Due to the pressure difference between the fluids, the bubble begins to break and this results in a circularly outward-going shock wave in water, an inward-going rarefaction wave in gas, and an interface lying in between those waves that separates the gas and the water. Soon after, the shock wave is diffracted through the nearby air-water surface, causing the subsequent deformation of the interface topology from a circle to an oval-like shape.

The contours of the density and the adaptive mesh of $N = 120 \times 75 \times 4$ are plotted at $t = 0.2, 0.4, 0.8,$ and 1.2 in Figs. 15 and 16. Once again, the mesh points are concentrated
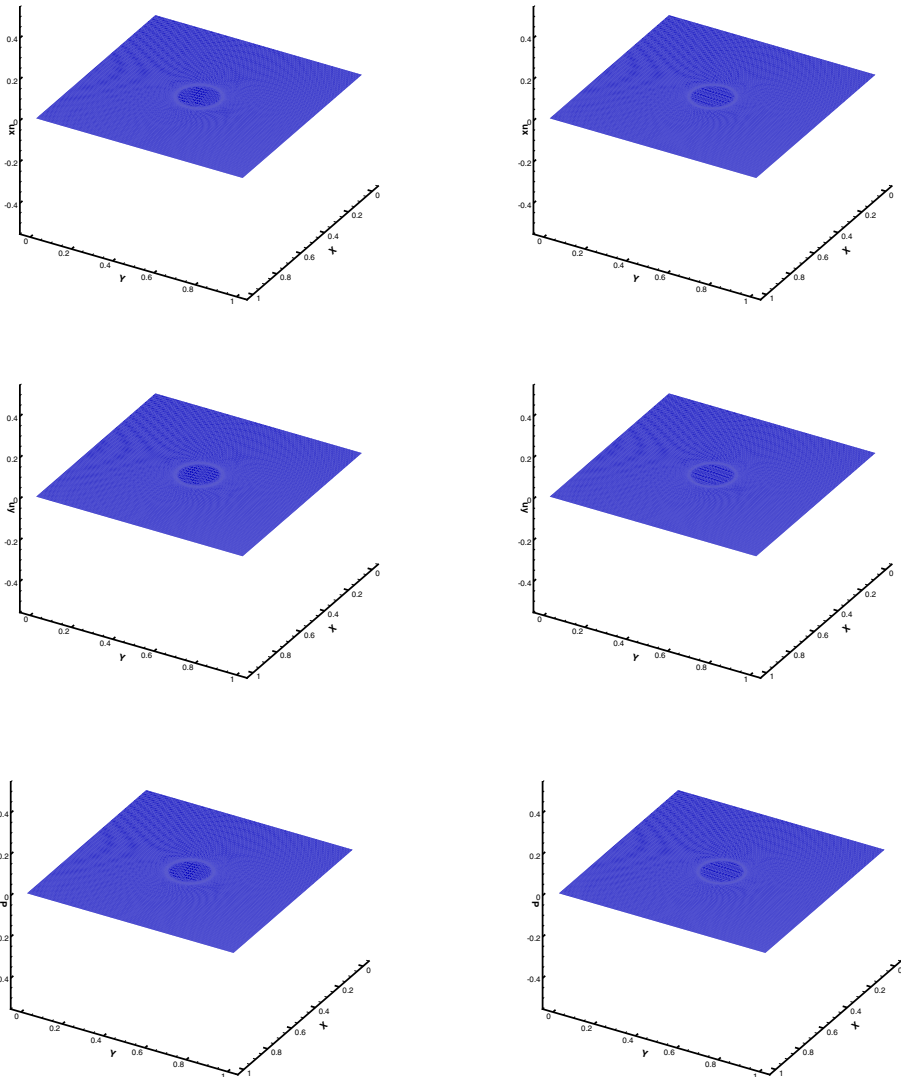
Fig. 12 Example 6. From top row to bottom row: the deviations of x- and y-components of the velocity and the pressure. Left column: P^1-DG; Right column: P^2-DG. $N = 100 \times 100 \times 4$

correctly near the material interface and shocks and P^2-DG appears to give higher resolution than P^1-DG.

5 Conclusions

A quasi-conservative DG-ALE method for multi-component flows has been proposed in the previous sections. The non-oscillatory kinetic flux is adopted to compute the numerical flux, which avoids the need to construct a Riemann solver. The main difference between the current
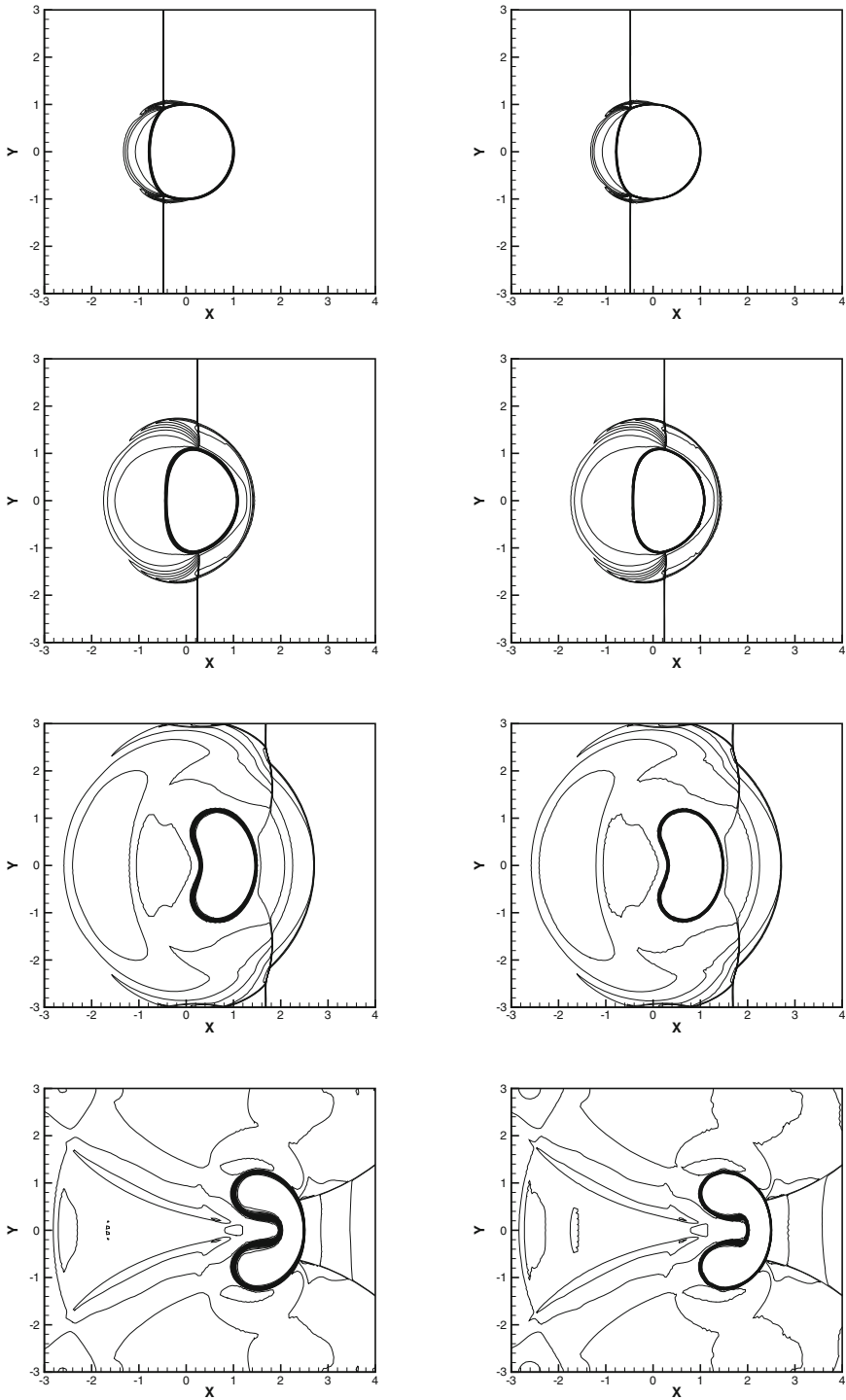
Fig. 13 Example 7 Density contours at $t = 0.5$, 1, 2, and 4 (from top to bottom). Left column: P^1-DG; Right column: P^2-DG. $N = 70 \times 60 \times 4$
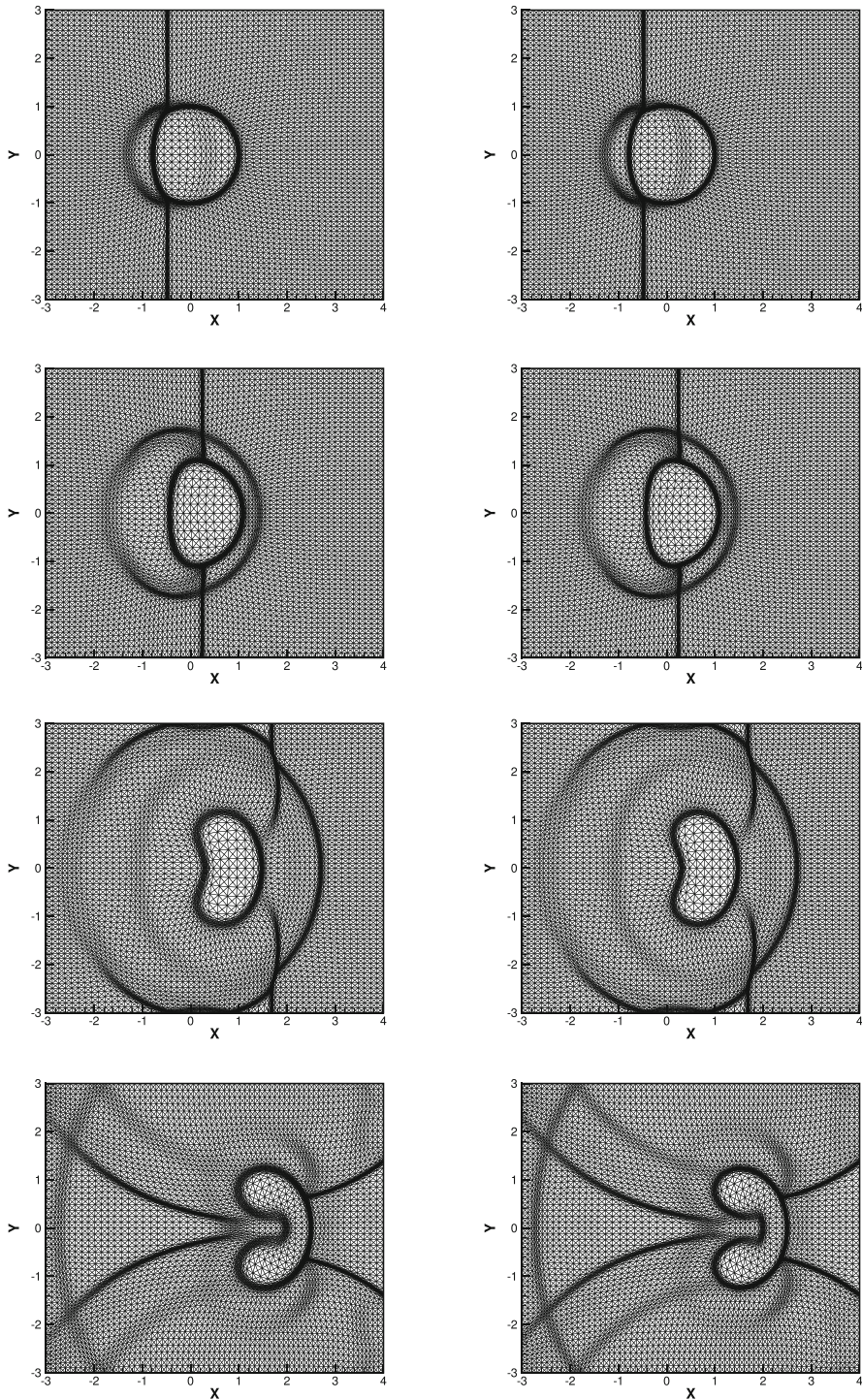
Fig. 14 Example 7 The adaptive moving mesh of $N = 70 \times 60 \times 4$ at time $t = 0.5, 1, 2,$ and 4 (from top to bottom). Left column: P^1-DG; Right column: P^2-DG
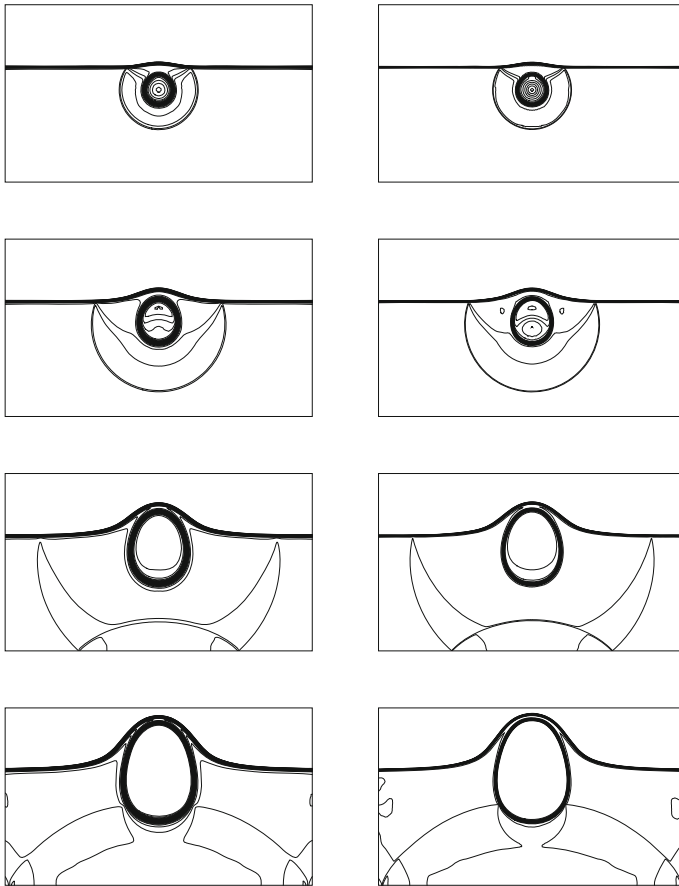
Fig. 15 Example 8 Contours of the density at $t = 0.2, 0.4, 0.8$, and 1.2 ms (from top to bottom). Left column: P^1-DG; Right column: P^2-DG. $N = 120 \times 75 \times 4$

method and other DG-ALE methods is that we use here a predictor-corrector strategy to define the grid velocity, i.e., a Lagrangian step is used to predict the new mesh and then the MMPDE moving mesh method is employed to improve the quality of the Lagrangian mesh. In this way, the method can take the advantages of both the Lagrangian meshing and the MMPDE moving mesh method and keep good quality of the mesh while tracking discontinuities such as shocks and material interfaces in the flow field. The quasi-conservative DG method [38] is adopted to solve the extended Euler equations of compressible multi-component flows on moving meshes. Numerical results in one and two dimensions have demonstrated the high-order accuracy of the method for smooth problems and its ability to capture shocks and material interfaces. Particularly the results demonstrate that the incorporation of the Lagrangian meshing with the MMPDE moving mesh method works well to concentrate mesh points in regions of shocks and material interfaces. In the future, we plan to extend the method to compressible multi-component flows with more general equations of state such as the Mie-Grüneisen equation of state.
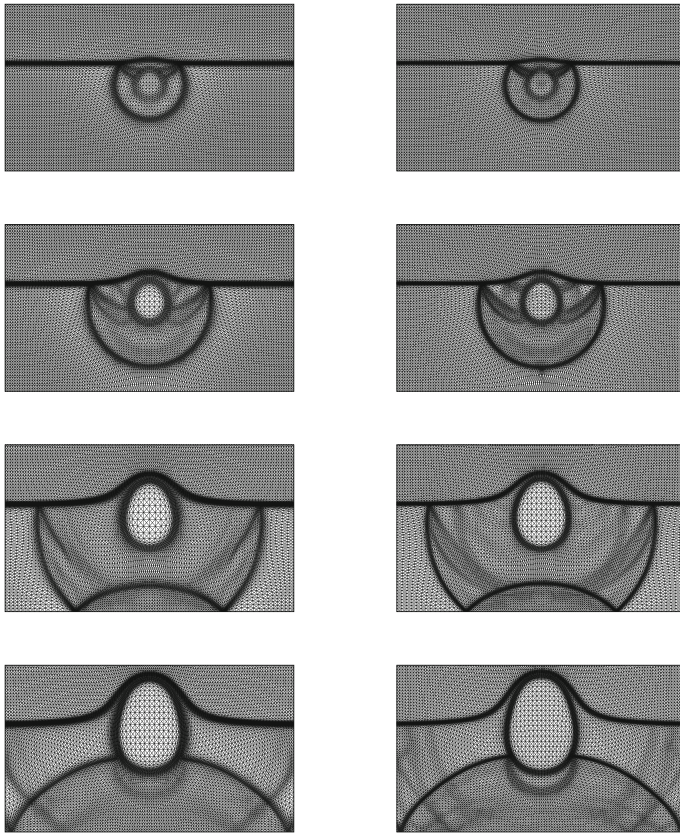
Fig. 16 Example 8 The adaptive moving mesh of $N = 120 \times 75 \times 4$ at $t = 0.2, 0.4, 0.8$, and 1.2 (from top to bottom). Left column: P^1-DG; Right column: P^2-DG

References

1. Abgrall, R.: How to prevent pressure oscillations in multicomponent flow calculations: A quasi conservative approach. J. Comput. Phys. **125**, 150–160 (1996)
2. Addessio, F. L., Carroll, D. E., Dukowicz, J. K., Harlow, F. H., Johnson, J. N., Kashiwa, B. A., Maltrud, M. E., Ruppel, H.: CAVEAT: A computer code for fluid dynamics problems with large distortion and internal slip, Los Alamos National Laboratory LA-10613-MSREVISED, (1990)
3. Baines, M.J.: Moving Finite Elements. Oxford University Press, Oxford (1994)
4. Baines, M.J., Hubbard, M.E., Jimack, P.K.: Velocity-based moving mesh methods for nonlinear partial differential equations. Comm. Comput. Phys. **10**, 509–576 (2011)
5. Boscheri, W.: High order direct Arbitrary-Lagrangian-Eulerian (ALE) finite volume schemes for hyperbolic systems on unstructured meshes. Arch. Comput. Methods Eng. **24**, 751–801 (2017)
6. Budd, C.J., Huang, W., Russell, R.D.: Adaptivity with moving grids. Acta Numer. **18**, 111–241 (2009)
7. Chen, Y.: A study of GKS method for multi-component flows (in Chinese), Ph.D thesis, Beijing, China Academy of Engineering Physics (2010)
8. Chen, Y., Jiang, S.: A non-oscillatory kinetic scheme for multi-component flows with the equation of state for a stiffened gas. J. Comput. Math. **29**, 661–683 (2011)
9. Chen, Y., Jiang, S.: Modified kinetic flux vector splitting schemes for compressible flows. J. Comput. Phys. **228**, 3582–3604 (2009)
10. Cheng, J., Zhang, F., Liu, T.G.: A discontinuous Galerkin method for the simulation of compressible gas-gas and gas-water two-medium flows. J. Comput. Phys. **403**, 109059 (2020)

11. Cheng, J., Shu, C.-W.: A high order ENO conservative Lagrangian type scheme for the compressible Euler equations. J. Comput. Phys. **227**, 1567–1596 (2007)
12. Cheng, J., Shu, C.-W.: Positivity-preserving Lagrangian scheme for multi-material compressible flow. J. Comput. Phys. **257**, 143–168 (2014)
13. Cockburn, B., Hou, S., Shu, C.-W.: The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. Math. Comp. **54**, 545–581 (1990)
14. Cockburn, B., Lin, S.-Y., Shu, C.-W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One dimensional systems. J. Comput. Phys. **84**, 90–113 (1989)
15. Cockburn, B., Shu, C.-W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. Math. Comp. **52**, 411–435 (1989)
16. Cockburn, B., Shu, C.-W.: The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. J. Comput. Phys. **141**, 199–224 (1998)
17. del Razo, M.J., Leveque, R.J.: Numerical methods for interface coupling of compressible and almost incompressible media. SIAM J. Sci. Comput. **39**, B486–B507 (2017)
18. Farhat, C., Geuzaine, P., Granndmont, C.: The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. J. Comput. Phys. **174**, 669–694 (2001)
19. Fu, P., Schnücke, G., Xia, Y.: Arbitrary Lagrangian-Eulerian discontinuous Galerkin method for conservation laws on moving simplex meshes. Math. Comp. **88**, 2221–2255 (2019)
20. Henry de Frahan, M. T., Varadan, S., Johnsen, E.: A new limiting procedure for discontinuous Galerkin methods applied to compressible multiphase flows with shocks and interfaces, J. Comput. Phys. 280, 489-509 (2015)
21. Hirt, C.W., Amsden, A.A., Cook, J.L.: An arbitrary Lagrangian-Eulerian computing method for all flow speed. J. Comput. Phys. **14**, 227–253 (1974)
22. Huang, W., Kamenski, L.: A geometric discretization and a simple implementation for variational mesh generation and adaptation. J. Comput. Phys. **301**, 322–337 (2015)
23. Huang, W., Kamenski, L.: On the mesh nonsingularity of the moving mesh PDE method. Math. Comput. **87**, 1887–1911 (2018)
24. Huang, W., Ren, Y., Russell, R.D.: Moving mesh methods based on moving mesh partial differential equations. J. Comput. Phys. **113**, 279–290 (1994)
25. Huang, W., Ren, Y., Russell, R.D.: Moving mesh partial differential equations (MMPDEs) based upon the equidistribution principle. SIAM J. Numer. Anal. **31**, 709–730 (1994)
26. Huang, W., Russell, R.D.: Adaptive moving mesh methods. Springer-Verlag, New York (2011)
27. Huang, W.: Mathematical principles of anisotropic mesh adaptation. Comm. Comput. Phys. **1**, 276–310 (2006)
28. Huang, W.: Variational mesh adaptation: isotropy and equidistribution. J. Comput. Phys. **174**, 903–924 (2001)
29. Huang, W.: Variational mesh adaptation II: error estimates and monitor functions. J. Comput. Phys. **184**, 619–648 (2003)
30. Jimack, P., Wathen, A.: Temporal derivatives in the finite-element method on continuously deforming grids. SIAM J. Numer. Anal. **28**, 990–1003 (1991)
31. Klingenberg, C., Schnücke, G., Xia, Y.: Arbitrary Lagrangian-Eulerian discontinuous Galerkin method for conservation laws: analysis and application in one dimension. Math. Comp. **86**, 1203–1232 (2017)
32. Kucharik, M., Shashkov, M.J.: One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian-Eulerian methods. J. Comput. Phys. **231**, 2851–2864 (2012)
33. Lesoinne, M., Farhat, C.: Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. Comput. Methods Appl. Mech. Engrg. **134**, 71–90 (1996)
34. Liu, H., Xu, K.: A Runge-Kutta discontinuous Galerkin method for viscous flow equations. J. Comput. Phys. **224**, 1223–1242 (2007)
35. Liu, N., Xu, X., Chen, Y.: High-order spectral volume scheme for multi-component flows using non-oscillatory kinetic flux. Comput. Fluids **152**, 120–133 (2017)
36. Lomtev, I., Kirby, R.M., Karniadakis, G.E.: A discontinuous Galerkin ALE method for compressible viscous flows in moving domains. J. Comput. Phys. **155**, 128–159 (1999)
37. Luo, D., Huang, W., Qiu, J.X.: A quasi-Lagrangian moving mesh discontinuous Galerkin method for hyperbolic conservation laws. J. Comput. Phys. **396**, 544–578 (2019)
38. Luo, D., Qiu, J.X., Zhu, J., Chen, Y.: A quasi-conservative discontinuous Galerkin method for multi-component flows using the non-oscillatory kinetic flux. J. Sci. Comput. **87**, 1–32 (2021)
39. Maire, P.H., Breil, J., Galera, S.: A cell-centered arbitrary Lagrangian-Eulerian (ALE) method. Int. J. Numer. Meth. Fluids **56**, 1161–1166 (2008)

40. Maire, P.H., Abgrall, R., Breil, J., Ovadia, J.: A cell-centered Lagrangian scheme for two-dimensional compressible flow problems. SIAM J. Sci. Comput. **29**, 1781–1824 (2007)

41. Miller, K., Miller, R.N.: Moving finite elements I. SIAM J. Numer. Anal. **18**, 1019–1032 (1981)

42. Nguyen, V.T.: An arbitrary Lagrangian-Eulerian discontinuous Galerkin method for simulations of flows over variable geometries. J. Fluids Struct. **26**, 312–329 (2010)

43. Pandare, A. K., Wang, C., Luo, H.: An arbitrary Lagrangian-Eulerian reconstructed discontinuous Galerkin method for compressible multiphase flows, 46th AIAA Fluid Dynamics Conference (2016)

44. Persson, P.O., Bonet, J., Peraire, J.: Discontinuous Galerkin solution of the Navier-Stokes equations on deformable domains. Comput. Methods Appl. Mech. Eng. **198**, 1585–1595 (2009)

45. Qiu, J.X., Liu, T.G., Khoo, B.C.: Runge-Kutta discontinuous Galerkin methods for compressible two-medium flow simulations: one-dimensional case. J. Comput. Phys. **222**, 353–373 (2007)

46. Qiu, J.X., Liu, T.G., Khoo, B.C.: Simulations of compressible two-medium flow by Runge-Kutta discontinuous Galerkin methods with the ghost fluid method. Comm. Comput. Phys. **3**, 479–504 (2008)

47. Qiu, J.X., Shu, C.-W.: Runge-Kutta discontinuous Galerkin method using WENO limiters. SIAM J. Sci. Comput. **26**, 907–929 (2005)

48. Reed, W. H., Hill, T. R.: Triangular mesh methods for neutron transport equation, Los Alamos Scientific Laboratory Report LA-UR-73-479 (1973)

49. Ren, X., Xu, K., Shyy, W.: A multi-dimensional high-order DG-ALE method based on gas-kinetic theory with application to oscillating bodies. J. Comput. Phys. **316**, 700–720 (2016)

50. Saleem, M.R., Ali, I., Qamar, S.: Application of discontinuous Galerkin method for solving a compressible five-equation two-phase flow model. Results Phys. **8**, 379–390 (2018)

51. Shu, C.-W.: Total-variation-diminishing time discretizations. SIAM J. Sci. Stat. Comput. **9**, 1073–1084 (1988)

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

53. Shyue, K.M.: An efficient shock-capturing algorithm for compressible multicomponent problems. J. Comput. Phys. **142**, 208–242 (1998)

54. Shyue, K.M.: A wave-propagation based volume tracking method for compressible multicomponent flow in two space dimensions. J. Comput. Phys. **215**, 219–244 (2006)

55. Tang, T.: Moving mesh methods for computational fluid dynamics flow and transport, Recent Advances in Adaptive Computation (Hangzhou, 2004), Volume 383 of AMS Contemporary Mathematics, pages 141-173. Amer. Math. Soc., Providence, RI, (2005)

56. Wang, C.-W., Shu, C.-W.: An interface treating technique for compressible multi-medium flow with Runge-Kutta discontinuous Galerkin method. J. Comput. Phys. **229**, 8823–8843 (2010)

57. Xu, K.: BGK-based scheme for multicomponent flow calculations. J. Comput. Phys. **134**, 122–133 (1997)

58. Xu, X., Ni, G., Jiang, S.: A high-order moving mesh kinetic scheme based on WENO reconstruction for compressible flows on unstructured meshes. J. Sci. Comput. **57**, 278–299 (2013)

59. Zhang, Z., Naga, A.: A new finite element gradient recovery method: superconvergence property. SIAM J. Sci. Comput. **26**, 1192–1213 (2005)

60. Zhao, X., Yu, X., Qiu, M., Qing, F., Zou, S.: An arbitrary Lagrangian-Eulerian RKDG method for multi-material flows on adaptive unstructured meshes. Comput. Fluids **207**, 104589 (2020)

61. Zhu, J., Qiu, J.X., Liu, T.G., Khoo, B.C.: High-order RKDG methods with WENO type limiters and conservative interfacial procedure for one-dimensional compressible multi-medium flow simulations. Appl. Numer. Math. **61**, 554–580 (2011)

62. Zhu, J., Qiu, J.X., Shu, C.-W.: High-order Runge-Kutta discontinuous Galerkin methods with a new type of multi-resolution WENO limiters. J. Comput. Phys. **404**, 109105 (2020)

63. Zhu, J., Shu, C.-W., Qiu, J.X.: High-order Runge-Kutta discontinuous Galerkin methods with a new type of multi-resolution WENO limiters on triangular meshes. Appl. Numer. Math. **153**, 519–539 (2020)