



# Single-Step Arbitrary Lagrangian–Eulerian Discontinuous Galerkin Method for 1-D Euler Equations

Jayesh Badwaik<sup>1</sup> · Praveen Chandrashekar<sup>2</sup> · Christian Klingenberg<sup>1</sup> 

Received: 6 March 2019 / Revised: 17 September 2019 / Accepted: 31 October 2019 /  
Published online: 8 January 2020  
© Shanghai University 2020

## Abstract

We propose an explicit, single-step discontinuous Galerkin method on moving grids using the arbitrary Lagrangian–Eulerian approach for one-dimensional Euler equations. The grid is moved with the local fluid velocity modified by some smoothing, which is found to considerably reduce the numerical dissipation introduced by Riemann solvers. The scheme preserves constant states for any mesh motion and we also study its positivity preservation property. Local grid refinement and coarsening are performed to maintain the mesh quality and avoid the appearance of very small or large cells. Second, higher order methods are developed and several test cases are provided to demonstrate the accuracy of the proposed scheme.

**Keywords** Discontinuous Galerkin method · Moving meshes · Arbitrary Lagrangian–Eulerian · Euler equations

**Mathematics Subject Classification** 65M60 · 35L04

## 1 Introduction

Finite volume schemes based on exact or approximate Riemann solvers are used for solving hyperbolic conservation laws like the Euler equations governing compressible flows. These schemes are able to compute discontinuous solutions in a stable manner since they have implicit dissipation built into them due to the upwind nature of the schemes. Higher

**Electronic supplementary material** The online version of this article (<https://doi.org/10.1007/s42967-019-00054-5>) contains supplementary material, which is available to authorized users.

✉ Christian Klingenberg  
klingenberg@mathematik.uni-wuerzburg.de

Jayesh Badwaik  
badwaik.jayesh@gmail.com

Praveen Chandrashekar  
praveen@math.tifrbng.res.in

<sup>1</sup> Department of Mathematics, University of Würzburg, Würzburg, Germany

<sup>2</sup> TIFR Center for Applicable Mathematics, Bangalore, India

order schemes are constructed following a reconstruction approach combined with a high order time integration scheme. Discontinuous Galerkin methods can be considered as higher order generalizations of finite volume methods which also make use of Riemann solver technology but do not need a reconstruction step since they evolve a polynomial solution inside each cell. While these methods are formally high order accurate on smooth solutions, they can still introduce too much numerical dissipation in some situations. Sprin-gel [1] gives the example of a Kelvin–Helmholtz instability in which adding a large constant velocity to both states leads to suppression of the instability due to excessive numerical dissipation. This behaviour is attributed to the fact that fixed grid methods based on upwind schemes are not Galilean invariant. Upwind schemes, even when they are formally high order accurate, are found to be too dissipative when applied to turbulent flows [2] since the numerical viscosity can overwhelm the physical viscosity.

For the linear convection equation  $u_t + au_x = 0$ , the first-order upwind scheme has the modified partial differential equation

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = \frac{1}{2} |a| h (1 - \nu) \frac{\partial^2 u}{\partial x^2} + O(h^2), \quad \nu = \frac{|a| \Delta t}{h},$$

which shows that the numerical dissipation is proportional to  $|a|$  which is the wave speed. In case of Euler equations simulated with a Riemann solver, e.g., the Roe scheme, the wave speeds are related to the eigenvalues of the flux Jacobian and the numerical dissipation would be proportional to the absolute values of the eigenvalues, e.g.,  $|v - c|$ ,  $|v|$ ,  $|v + c|$  where  $v$  is the fluid velocity and  $c$  is the sound speed. This type of the numerical viscosity is not Galilean invariant since the fluid velocity depends on the coordinate frame adopted for the description of the flow. Adding a large translational velocity to the coordinate frame will increase the numerical viscosity and reduce the accuracy of the numerical solution. Such high numerical viscosity can be eliminated or minimized if the grid moves along with the flow as in Lagrangian methods [3–5]. However, pure Lagrangian methods encounter the issue of large grid deformations that occur in highly sheared flows as in the Kelvin–Helmholtz problem requiring some form of re-meshing. A related approach is to use the arbitrary Lagrangian–Eulerian approach [6, 7] where the mesh velocity can be chosen to be close to the local fluid velocity but may be regularized to maintain the mesh quality. Even in the ALE approach, it may be necessary to perform some local remeshing to prevent the grid quality from degrading. In [1], the mesh is regenerated after every time step based on a Delaunay triangulation, which allows it to maintain good mesh quality even when the fluid undergoes large shear deformation. However, these methods have been restricted to second-order accuracy as they rely on unstructured finite volume schemes on general polygonal/polyhedral cells, where achieving higher order accuracy is much more difficult compared to structured grids.

Traditionally, ALE methods have been used for problems involving moving boundaries as in wing flutter, store separation and other problems involving fluid structure interaction [8–12]. In these applications, the main reason to use ALE is not to minimize the dissipation in upwind schemes but to account for the moving boundaries and, hence, the grid velocities are chosen based on boundary motion and with a view to maintain good mesh quality. Another class of methods solves the PDE on moving meshes where the mesh motion is determined based on a monitor function which is designed to detect regions of large gradients in the solution, see [13, 14] and the references therein. These methods achieve automatic clustering of grid points in regions of large gradients. ALE schemes have been used to compute multi-material flows as in [15], since they are useful to accurately track the material interface.

The mesh velocity was chosen to be equal to the contact speed but away from the material contact, the velocity was chosen by linear interpolation and was not close to Lagrangian. There are other methods for choosing the mesh velocity which have been studied in [16, 17]. Lax–Wendroff type ALE schemes for compressible flows have been developed in [18]. Finite volume schemes based on the ADER approach have been developed on unstructured grids [19–21]. The theoretical analysis of ALE-DG schemes in the framework of Runge–Kutta time stepping for conservation laws has been done in [22].

In the present work, we consider only the one-dimensional problem to set down the fundamental principles with which in an upcoming work, we shall solve the multi-dimensional problem. The numerical method developed here will be usable in the multiple dimensions, but additional work is required in multiple dimensions to maintain a good mesh quality under fluid flow deformation. We develop an explicit discontinuous Galerkin scheme that is conservative on moving meshes and automatically satisfies the geometric conservation law. The scheme is a single-step method which is achieved using a predictor computed from a Runge–Kutta scheme that is local to each cell in the sense that it does not require any data from neighbouring cells and belongs to the class of schemes called the ADER method. Due to the single-step nature of the scheme, the TVD limiter has to be applied only once in each time step unlike in multi-stage Runge–Kutta schemes where the limiter is applied after each stage update. This nature of the ADER scheme can reduce its computational expense, especially in multi-dimensional problems and while performing parallel computations. The mesh velocity is specified at each cell face as the local velocity with some smoothing. We analyze the positivity of the first-order scheme using the Rusanov flux and derive a CFL condition. The scheme is shown to be exact for steady moving contact waves and the solutions are invariant to the motion of the coordinate frame. Due to the Lagrangian nature, the Roe scheme does not require any entropy fix. However, we identify the possibility of spurious contact waves arising in some situations. This is due to the vanishing of the eigenvalue corresponding to the contact wave. While the cell averages are well predicted, the higher moments of the solution can be inaccurate. This behaviour of Lagrangian DG schemes does not seem to have been reported in the literature. We propose a fix for the eigenvalue in the spirit of the entropy fix of Harten [23] that prevents the spurious contact waves from occurring in the solution. The methodology developed here will be extended to multi-dimensional flows in a future work with a view towards handling complex sheared flows.

The rest of the paper is organized as follows. Section 2 introduces the Euler equation model that is used in the rest of the paper. In Sect. 3, we explain the derivation of the scheme on a moving mesh together with the quadrature approximations and computation of mesh velocity. The computation of the predicted solution is detailed in Sect. 4. The TVD type limiter is presented in Sect. 5 for a non-uniform mesh, Sect. 6 shows the positivity of the first-order scheme and Sect. 7 shows the preservation of constant states. The grid coarsening and refinement strategy are explained in Sect. 8, while Sect. 9 presents a series of numerical results.

## 2 Euler Equations

The Euler equations model the conservation of mass, momentum and energy, and can be written as a system of coupled partial differential equations laws of the form

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial f(\mathbf{u})}{\partial x} = 0, \quad (1)$$

where  $\mathbf{u}$  is called the vector of conserved variables and  $\mathbf{f}(\mathbf{u})$  is the corresponding flux given by

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho v \\ E \end{bmatrix}, \quad \mathbf{f}(\mathbf{u}) = \begin{bmatrix} \rho v \\ p + \rho v^2 \\ \rho H v \end{bmatrix}.$$

In the above expressions,  $\rho$  is the density,  $v$  is the fluid velocity,  $p$  is the pressure and  $E$  is the total energy per unit volume, which for an ideal gas is given by  $E = p/(\gamma - 1) + \rho v^2/2$ , with  $\gamma > 1$  being the ratio of specific heats at constant pressure and volume, and  $H = (E + p)/\rho$  is the enthalpy. The Euler equations form a hyperbolic system; the flux Jacobian  $A(\mathbf{u}) = \mathbf{f}'(\mathbf{u})$  has real eigenvalues and linearly independent eigenvectors. The eigenvalues are  $v - c$ ,  $v$ ,  $v + c$  where  $c = \sqrt{\gamma p/\rho}$  is the speed of sound and the corresponding right eigenvectors are given by

$$r_1 = \begin{bmatrix} 1 \\ v - c \\ H - vc \end{bmatrix}, \quad r_2 = \begin{bmatrix} 1 \\ v \\ \frac{1}{2}v^2 \end{bmatrix}, \quad r_3 = \begin{bmatrix} 1 \\ v + c \\ H + vc \end{bmatrix}. \tag{2}$$

The hyperbolic property implies that  $A$  can be diagonalized as  $A = R\Lambda R^{-1}$  where  $R$  is the matrix formed by the right eigenvectors as the columns and  $\Lambda$  is the diagonal matrix of eigenvalues.

### 3 Discontinuous Galerkin Method

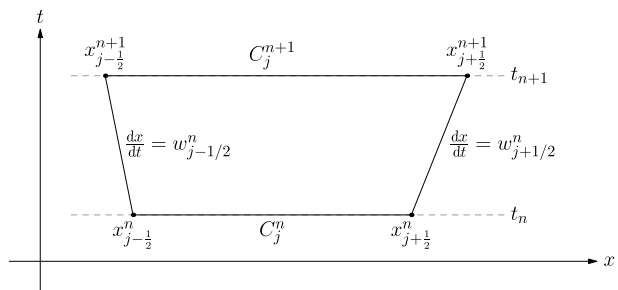
#### 3.1 Mesh and Solution Space

Consider a partition of the domain into disjoint cells with the  $j$ th cell being denoted by  $C_j(t) = [x_{j-\frac{1}{2}}(t), x_{j+\frac{1}{2}}(t)]$ . As the notation shows, the cell boundaries are time dependent which means that the cell is moving in some specified manner. The time levels are denoted by  $t_n$  with the time step  $\Delta t_n = t_{n+1} - t_n$ . The boundaries of the cells move with a constant velocity in the time interval  $(t_n, t_{n+1})$  given by

$$w_{j+\frac{1}{2}}(t) = w_{j+\frac{1}{2}}^n, \quad t_n < t < t_{n+1},$$

which defines a cell in space–time as shown in Fig. 1. The algorithm to choose the mesh velocity  $w_{j+\frac{1}{2}}^n$  is explained in a later section. The location of the cell boundaries is given by

**Fig. 1** Example of a space–time cell in the time interval  $(t_n, t_{n+1})$



$$x_{j+\frac{1}{2}}(t) = x_{j+\frac{1}{2}}^n + (t - t_n)w_{j+\frac{1}{2}}^n, \quad t_n \leq t \leq t_{n+1}.$$

Let  $x_j(t)$  and  $h_j(t)$  denote the center of the cell  $C_j(t)$  and its length, i.e.,

$$x_j(t) = \frac{1}{2}(x_{j-\frac{1}{2}}(t) + x_{j+\frac{1}{2}}(t)), \quad h_j(t) = x_{j+\frac{1}{2}}(t) - x_{j-\frac{1}{2}}(t).$$

Let  $w(x, t)$  be the continuous linear interpolation of the mesh velocity which is given by

$$w(x, t) = \frac{x_{j+\frac{1}{2}}(t) - x}{h_j(t)}w_{j-\frac{1}{2}}^n + \frac{x - x_{j-\frac{1}{2}}(t)}{h_j(t)}w_{j+\frac{1}{2}}^n, \quad x \in C_j(t), \quad t \in (t_n, t_{n+1}).$$

We will approximate the solution of the conservation law by piecewise polynomials which are allowed to be discontinuous across the cell boundaries as shown in Fig. 2. For a given degree  $k \geq 0$ , the solution in the  $j$ th cell is given by

$$u_h(x, t) = \sum_{m=0}^k u_{j,m}(t)\varphi_m(x, t), \quad x \in C_j(t),$$

where  $\{u_{j,m} \in \mathbb{R}^3, 0 \leq m \leq k\}$  are the degrees of freedom associated with the  $j$ th cell. The basis functions  $\varphi_m$  are defined in terms of Legendre polynomials

$$\varphi_m(x, t) = \hat{\varphi}_m(\xi) = \sqrt{2m + 1}P_m(\xi), \quad \xi = \frac{x - x_j(t)}{\frac{1}{2}h_j(t)},$$

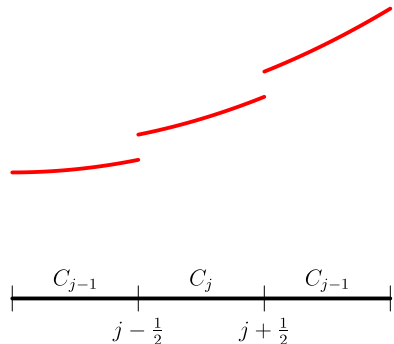
where  $P_m : [-1, +1] \rightarrow \mathbb{R}$  is the Legendre polynomial of degree  $m$ . The above definition of the basis functions implies the following orthogonality property:

$$\int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)} \varphi_l(x, t)\varphi_m(x, t)dx = h_j(t)\delta_{lm}, \quad 0 \leq l, m \leq k. \tag{3}$$

We will sometimes also write the solution in the  $j$ th cell in terms of the reference coordinates  $\xi$  as

$$u_h(\xi, t) = \sum_{m=0}^k u_{j,m}(t)\hat{\varphi}_m(\xi),$$

**Fig. 2** Example of a discontinuous piecewise polynomial solution



and we will use the same notation  $\mathbf{u}$  to denote both functions.

### 3.2 Derivation of the Scheme

To derive the DG scheme on a moving mesh, let us introduce the change of variable  $(x, t) \rightarrow (\xi, \tau)$  given by

$$\tau = t, \quad \xi = \frac{x - x_j(t)}{\frac{1}{2}h_j(t)}. \tag{4}$$

For any  $0 \leq l \leq k$ , we now calculate the rate of change of the  $l$ th moment of the solution starting from

$$\begin{aligned} \frac{d}{dt} \int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)} \mathbf{u}(x, t) \varphi_l(x, t) dx &= \frac{d}{d\tau} \int_{-1}^{+1} \mathbf{u}(\xi, \tau) \hat{\varphi}_l(\xi) \frac{1}{2} h_j(\tau) d\xi \\ &= \frac{1}{2} \int_{-1}^{+1} \left[ h_j(\tau) \frac{\partial \mathbf{u}}{\partial \tau} + \mathbf{u} \frac{dh_j}{d\tau} \right] \hat{\varphi}_l(\xi) d\xi, \end{aligned}$$

wherein we used the change of variables given by (4). But we also have the inverse transform

$$t = \tau, \quad x = x_j(\tau) + \frac{\xi}{2} h_j(\tau),$$

and hence

$$\frac{\partial t}{\partial \tau} = 1, \quad \frac{\partial x}{\partial \tau} = \frac{dx_j}{d\tau} + \frac{\xi}{2} \frac{dh_j}{d\tau} = \frac{1}{2}(w_{j-\frac{1}{2}} + w_{j+\frac{1}{2}}) + \frac{\xi}{2}(w_{j+\frac{1}{2}} - w_{j-\frac{1}{2}}) = w(x, t).$$

Using the above relations, we can easily show that

$$\frac{\partial \mathbf{u}}{\partial \tau}(\xi, \tau) = \frac{\partial \mathbf{u}}{\partial t}(x, t) + w(x, t) \frac{\partial \mathbf{u}}{\partial x}(x, t).$$

Moreover

$$\frac{dh_j}{d\tau} = w_{j+\frac{1}{2}} - w_{j-\frac{1}{2}} = h_j \frac{\partial w}{\partial x},$$

since  $w(x, t)$  is linear in  $x$  and hence  $\frac{\partial w}{\partial x}$  is constant inside each cell. Hence, the  $l$ th moment evolves according to

$$\begin{aligned} \frac{d}{dt} \int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)} \mathbf{u}(x, t) \varphi_l(x, t) dx &= \int_{-1}^{+1} \left( \frac{\partial \mathbf{u}}{\partial t} + w \frac{\partial \mathbf{u}}{\partial x} + \mathbf{u} \frac{\partial w}{\partial x} \right) \hat{\varphi}_l(\xi) \frac{1}{2} h_j d\xi \\ &= \int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)} \left( -\frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} + \frac{\partial}{\partial x}(w\mathbf{u}) \right) \varphi_l(x, t) dx, \end{aligned}$$

where we have transformed back to the physical coordinates and made use of the conservation law (1) to replace the time derivative of the solution with the flux derivative. Define the ALE flux

$$g(\mathbf{u}, w) = f(\mathbf{u}) - wu. \tag{5}$$

Performing an integration by parts in the  $x$  variable, we obtain

$$\begin{aligned} \frac{d}{dt} \int_{x_{j-\frac{1}{2}}^-(t)}^{x_{j+\frac{1}{2}}^+(t)} \mathbf{u}_h(x, t) \varphi_l(x, t) dx &= \int_{x_{j-\frac{1}{2}}^-(t)}^{x_{j+\frac{1}{2}}^+(t)} g(\mathbf{u}_h, w) \frac{\partial}{\partial x} \varphi_l(x, t) dx \\ &\quad + \hat{g}_{j-\frac{1}{2}}(\mathbf{u}_h(t)) \varphi_l(x_{j-\frac{1}{2}}^+, t) - \hat{g}_{j+\frac{1}{2}}(\mathbf{u}_h(t)) \varphi_l(x_{j+\frac{1}{2}}^-, t), \end{aligned}$$

where we have introduced the numerical flux

$$\hat{g}_{j+\frac{1}{2}}(\mathbf{u}_h(t)) = \hat{g}(\mathbf{u}_{j+\frac{1}{2}}^-(t), \mathbf{u}_{j+\frac{1}{2}}^+(t), w_{j+\frac{1}{2}}(t)),$$

which provides an approximation to the ALE flux, see Appendix A. Integrating over the time interval  $(t_n, t_{n+1})$  and using (3), we obtain

$$\begin{aligned} h_j^{n+1} \mathbf{u}_{j,l}^{n+1} &= h_j^n \mathbf{u}_{j,l}^n + \int_{t_n}^{t_{n+1}} \int_{x_{j-\frac{1}{2}}^-(t)}^{x_{j+\frac{1}{2}}^+(t)} g(\mathbf{u}_h, w) \frac{\partial}{\partial x} \varphi_l(x, t) dx dt \\ &\quad + \int_{t_n}^{t_{n+1}} [\hat{g}_{j-\frac{1}{2}}(t) \varphi_l(x_{j-\frac{1}{2}}^+, t) - \hat{g}_{j+\frac{1}{2}}(t) \varphi_l(x_{j+\frac{1}{2}}^-, t)] dt. \end{aligned}$$

The above scheme has an implicit nature since the unknown solution  $\mathbf{u}_h$  appears on the right-hand side integrals whereas we only know the solution at time  $t_n$ . To obtain an explicit scheme, we assume that we have available with us a predicted solution  $\mathbf{U}_h$  in the time interval  $(t_n, t_{n+1})$ , which is used in the time integrals to obtain an explicit scheme. Moreover, the integrals are computed using quadrature in space and time leading to the fully discrete scheme

$$\begin{aligned} h_j^{n+1} \mathbf{u}_{j,l}^{n+1} &= h_j^n \mathbf{u}_{j,l}^n \\ &\quad + \Delta t_n \sum_r \theta_r h_j(\tau_r) \sum_q \eta_q g(\mathbf{U}_h(x_q, \tau_r), w(x_q, \tau_r)) \frac{\partial}{\partial x} \varphi_l(x_q, \tau_r) \\ &\quad + \Delta t_n \sum_r \theta_r [\hat{g}_{j-\frac{1}{2}}(\mathbf{U}_h(\tau_r)) \varphi_l(x_{j-\frac{1}{2}}^+, \tau_r) - \hat{g}_{j+\frac{1}{2}}(\mathbf{U}_h(\tau_r)) \varphi_l(x_{j+\frac{1}{2}}^-, \tau_r)], \end{aligned} \tag{6}$$

where  $\theta_r$  is the weight for time quadrature and  $\eta_q$  is the weight for spatial quadrature. For the spatial integral, we will use  $q = k + 1$  point Gauss quadrature. For the time integral, we will use the mid-point rule for  $k = 1$  and two-point Gauss quadrature for  $k = 2, 3$ . Since the mesh is moving, the spatial quadrature points  $x_q$  depend on the quadrature time  $\tau_r$ , though this is not clear from the notation. In practice, the integrals are computed by mapping the cell to the reference cell, and the basis functions and its derivatives are also evaluated on the reference cell. The quadrature points in the reference cell are independent of time due to the linear mesh evolution.

### 3.3 Mesh Velocity

The mesh velocity must be close to the local fluid velocity to have a Lagrangian character to the scheme. Since the solution is discontinuous, there is no unique fluid velocity at the mesh boundaries. Some researchers, especially in the context of Lagrangian methods, solve a Riemann problem at the cell face to determine the face velocity. Since we use an ALE formulation, we do not require the exact fluid velocity which is anyway not available to use since we only have a predicted solution. Following the exact trajectory of the fluid would also lead to curved trajectories for the grid point, which is an unnecessary complication. In our work, we make two different choices for the mesh velocities.

- i. The first choice is to take an average of the two velocities at every face. In the numerical results, we refer to this as ADG

$$\tilde{w}_{j+\frac{1}{2}}^n = \frac{1}{2} \left[ v \left( x_{j+\frac{1}{2}}^-, t_n \right) + v \left( x_{j+\frac{1}{2}}^+, t_n \right) \right].$$

- ii. The second choice is to solve a linearized Riemann problem at the face at time  $t_n$ . In the numerical results, we refer to this as RDG. For simplicity of notation, let the solution to the left of the face  $x_{j+\frac{1}{2}}$  be represented as  $u_{j+\frac{1}{2}}^-$  and the solution to the right be represented as  $u_{j+\frac{1}{2}}^+$ . Then,

$$\tilde{w}_{j+\frac{1}{2}}^n = \frac{\rho_j^n c_j^n v_j^n + \rho_{j+1}^n c_{j+1}^n v_{j+1}^n}{\rho_j^n c_j^n + \rho_{j+1}^n c_{j+1}^n} + \frac{p_j^n - p_{j+1}^n}{\rho_j^n c_j^n + \rho_{j+1}^n c_{j+1}^n}.$$

We will also perform some smoothing of the mesh velocity, e.g., the actual face velocity is computed from

$$w_{j+\frac{1}{2}}^n = \frac{1}{3} \left( \tilde{w}_{j-\frac{1}{2}}^n + \tilde{w}_{j+\frac{1}{2}}^n + \tilde{w}_{j+\frac{3}{2}}^n \right).$$

Note that our algorithm to choose the mesh velocity is very local and hence easy and efficient to implement as it does not require the solution of any global problems. In Springel [1], the mesh velocity is adjusted so that the cells remain nearly isotropic which leads to smoothly varying cell sizes. Such an approach leads to many parameters that need to be selected and we did not find a good way to make this choice that works well for a range of problems. Instead, we will make use of mesh refinement and coarsening to maintain the quality of cells, i.e., to prevent very small or large cells from occurring in the grid. The use of a DG scheme makes it easy to perform such local mesh adaptation without loss of accuracy.

**Remark 1** Consider the application of the proposed ALE-DG scheme to the linear advection equation  $u_t + au_x = 0$ . In this case, the mesh velocity is equal to the advection velocity  $w_{j+\frac{1}{2}} = a$ , i.e., the cells move along the characteristics. This implies that the ALE flux  $g(u, w) = au - wu = 0$  and also the numerical flux  $\hat{g}_{j+\frac{1}{2}} = 0$ . Thus, the DG scheme reduces to

$$\int_{x_{j-\frac{1}{2}}^{n+1}}^{x_{j+\frac{1}{2}}^{n+1}} u_h(x, t_{n+1}) \varphi_l(x, t_{n+1}) dx = \int_{x_{j-\frac{1}{2}}^n}^{x_{j+\frac{1}{2}}^n} u_h(x, t_n) \varphi_l(x, t_n) dx, \quad l = 0, 1, \dots, k,$$



so that the solution at time  $t_n$  has been advected exactly to the solution at time  $t_{n+1}$ . Note that there is no time step restriction involved in this case and the accuracy of the predicted solution is also not relevant. If the initial condition has a discontinuity coinciding with a cell face, then the scheme advects the discontinuity exactly without any diffusion.

### 4 Computing the Predictor

The predicted solution is used to approximate the flux integrals over the time interval  $(t_n, t_{n+1})$  and the method to compute this must be local, i.e., it must not require solution from neighbouring cells. Several methods for computing the predictor have been reviewed in [24]. The simplest approach is to use a Taylor expansion in space and time. Since the cells are moving, the Taylor expansion has to be performed along the trajectory of the mesh motion. For a second-order scheme, an expansion retaining only linear terms in  $t$  and  $x$  is sufficient. Consider a quadrature point  $(x_q, \tau_r)$ , the Taylor expansion of the solution around the cell center  $x_j^n$  and time level  $t_n$  yields

$$\begin{aligned} \mathbf{u}_h(x_q, \tau_r) &= \mathbf{u}_h(x_j^n, t_n) + (\tau_r - t_n) \frac{\partial \mathbf{u}_h}{\partial t}(x_j^n, t_n) + (x_q - x_j^n) \frac{\partial \mathbf{u}_h}{\partial x}(x_j^n, t_n) \\ &\quad + O(\tau_r - t_n)^2 + O(x_q - x_j^n)^2, \end{aligned}$$

and the predicted solution is given by truncating the Taylor expansion at linear terms, leading to

$$U(x_q, \tau_r) = \mathbf{u}_h(x_j^n, t_n) + (\tau_r - t_n) \frac{\partial \mathbf{u}_h}{\partial t}(x_j^n, t_n) + (x_q - x_j^n) \frac{\partial \mathbf{u}_h}{\partial x}(x_j^n, t_n).$$

Using the conservation law, the time derivative is written as  $\frac{\partial \mathbf{u}}{\partial t} = -\frac{\partial f}{\partial x} = -A \frac{\partial \mathbf{u}}{\partial x}$  so that the predictor is given by

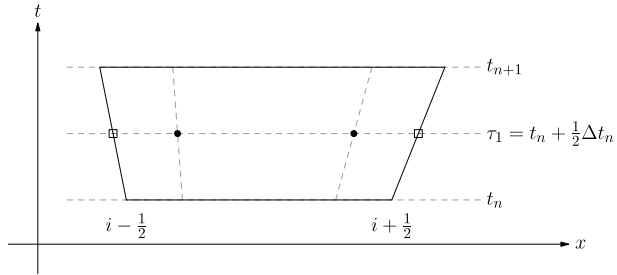
$$U_h(x_q, \tau_r) = \mathbf{u}_h^n(x_j^n) - (\tau_r - t_n) \left[ A(\mathbf{u}_h^n(x_j^n)) - w_q I \right] \frac{\partial \mathbf{u}_h^n}{\partial x}(x_j^n). \tag{7}$$

The above predictor is used for the case of polynomial degree  $k = 1$ . This procedure can be extended to higher orders by including more terms in the Taylor expansion but the algebra becomes complicated. Instead we will adopt the approach of continuous explicit Runge–Kutta (CERK) schemes [25] to approximate the predictor.

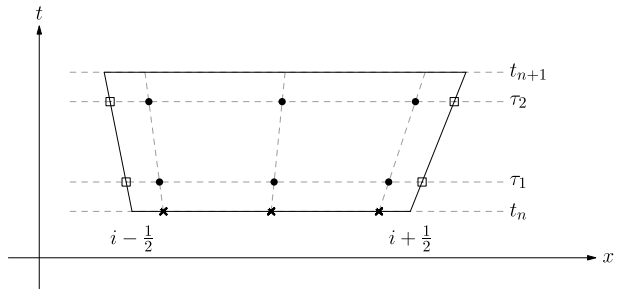
Let us choose a set of  $(k + 1)$  distinct nodes, e.g., Gauss–Legendre or Gauss–Lobatto nodes, which uniquely define the polynomial of degree  $k$ . These nodes are moving with velocity  $w(x, t)$ , so that the time evolution of the solution at node  $x_m$  is governed by

$$\begin{aligned} \frac{dU_m}{dt} &= \frac{\partial}{\partial t} U_h(x_m, t) + w(x_m, t) \frac{\partial}{\partial x} U_h(x_m, t) \\ &= -\frac{\partial}{\partial x} f(U_h(x_m, t)) + w(x_m, t) \frac{\partial}{\partial x} U_h(x_m, t) \\ &= -[A(U_m(t)) - w_m(t)I] \frac{\partial}{\partial x} U_h(x_m, t) =: \mathbf{K}_m(t), \end{aligned}$$

**Fig. 3** Quadrature points for second-order scheme



**Fig. 4** Quadrature points for third-order scheme



wherein we have made use of the PDE to write the time derivative in terms of spatial derivative of the flux. This equation is solved with the initial condition

$$U_m(t_n) = u_h(x_m, t_n).$$

Using a Runge–Kutta scheme of sufficient order (see Appendix B), we will approximate the solution at these nodes as

$$U_m(t) = u_h(x_m, t_n) + \sum_{s=1}^{n_s} b_s((t - t_n)/\Delta t_n)K_{m,s}, \quad t \in [t_n, t_{n+1}), \quad m = 0, 1, \dots, k,$$

where  $K_{m,s} = K_m(t_n + \theta_s \Delta t_n)$ ,  $\theta_s \Delta t_n$  is the stage time and  $b_s$  are certain polynomials related to the CERK scheme and given in Appendix B. Note that we are evolving the nodal values but the computation of  $K_{m,s}$  requires the modal representation of the solution in order to calculate spatial derivative of the solution.

Once the predictor is computed as above, it must be evaluated at the quadrature point  $(x_q, \tau_r)$  as follows. For each time quadrature point  $\tau_r \in (t_n, t_{n+1})$ ,

- i. compute nodal values  $U_m(\tau_r)$ ,  $m = 0, 1, \dots, k$ ;
- ii. for each  $r$ , convert the nodal values to modal coefficients  $u_{m,r}$ ,  $m = 0, 1, \dots, k$ ;
- iii. evaluate predictor  $U_h(x_q, \tau_r) = \sum_{m=0}^k u_{m,r} \varphi_m(x_q, \tau_r)$ .

The conversion from nodal to modal values is accomplished through a Vandermonde matrix of size  $(k + 1) \times (k + 1)$  which is the same for every cell and can be inverted once before the iterations start. The predictor is also computed at the cell boundaries using the above procedure. Figures 3 and 4 show the quadrature points used in the second-, third- and fourth-order scheme. For the second-order scheme, the values at  $\bullet$  and  $\square$  points are obtained from the predictor based on Taylor expansion as given in (7). For third- and

fourth-order schemes, the nodal values  $\times$  are evolved forward in time by the CERK scheme and evaluated at the  $\bullet$  points. The  $\bullet$  point values are converted to modal coefficients using which the solution at the  $\square$  points is computed.

**Remark 2** By looping over each cell in the mesh, the predicted solution is computed in each cell and the cell integral in (6) is evaluated. The trace values  $U_{j+\frac{1}{2}}^-(\tau_r)$  and  $U_{j-\frac{1}{2}}^+(\tau_r)$  at the  $\square$  points needed for quadrature in time are computed and stored. These are later used in a loop over the cell faces where the numerical flux is evaluated. Thus, the algorithm is easily parallelizable on multiple core machines and/or using threads.

### 5 Control of Oscillation by Limiting

High order schemes for hyperbolic equations suffer from spurious numerical oscillations when discontinuities or large gradients are present in the solution which cannot be accurately resolved on the mesh. In the case of scalar problems, this is a manifestation of loss of TVD property and hence limiters are used to satisfy some form of TVD condition. In the case of DG schemes, the limiter is used as a post-processor which is applied on the solution after the time update has been performed. If the limiter detects that the solution is oscillatory, then the solution polynomial is reduced to at most a linear polynomial with a limited slope. In the present scheme, the limiter is applied after the solution is updated from time  $t_n$  to time  $t_{n+1}$ , i.e., the solution  $u_h^{n+1}$  obtained from (6) is post-processed by the limiter. Since the mesh is inherently non-uniform due to it being moved with the flow, we modify the standard TVD limiter to account for this non-uniformity. Also, since we are solving a system of conservation laws, the limiter is applied on the local characteristic variables which gives better results than applying it directly on the conserved variables [26].

The solution in cell  $j$  can be written as<sup>1</sup>

$$u_h(x) = \bar{u}_j + \frac{x - x_j}{\frac{1}{2}h_j} s_j + \text{higher order terms,}$$

where  $\bar{u}_j$  is the cell average value and  $s_j$  is proportional to the derivative of the solution at the cell center. Let  $R_j, L_j$  denote the matrix of right and left eigenvectors evaluated at the cell average value  $\bar{u}_j$  which satisfy  $L_j = R_j^{-1}$ , and the right eigenvectors are given in (2). The local characteristic variables are defined by  $\bar{u}^* = L_j \bar{u}$  and  $s^* = L_j s$ . We first compute the limited slope of the characteristic variables from

$$s_j^{**} = h_j \min\text{mod}\left(\frac{s_j^*}{h_j}, \frac{\bar{u}_j^* - \bar{u}_{j-1}^*}{\frac{1}{2}(h_{j-1} + h_j)}, \frac{\bar{u}_{j+1}^* - \bar{u}_j^*}{\frac{1}{2}(h_j + h_{j+1})}\right),$$

where the minmod function is defined by

$$\min\text{mod}(a, b, c) = \begin{cases} s \min(|a|, |b|, |c|), & \text{if } s = \text{sign}(a) = \text{sign}(b) = \text{sign}(c), \\ 0, & \text{otherwise.} \end{cases}$$

<sup>1</sup> We suppress the time variable for clarity of notation.

If  $s_j^{**} = s_j^*$ , then we retain the solution as it is, and otherwise, the solution is modified to

$$u_h(x) = \bar{u}_j + \frac{x - x_j}{\frac{1}{2}h_j} R_j s_j^{**}.$$

This corresponds to a TVD limiter which is known to lose accuracy at smooth extrema [27] since the minmod function returns zero slope at local extrema. The TVB limiter corresponds to replacing the minmod limiter function with the following function:

$$\overline{\text{minmod}}(a, b, c) = \begin{cases} a, & \text{if } |a| \leq Mh^2, \\ \text{minmod}(a, b, c), & \text{otherwise,} \end{cases} \tag{8}$$

where the parameter  $M$  is an estimate of the second derivative of the solution at smooth extrema [27] and has to be chosen by the user.

### 6 Positivity Property

The solutions of Euler equations are well defined only if the density and pressure are positive quantities. This is not a priori guaranteed by the DG scheme even when the TVD limiter is applied. In the case of Runge–Kutta DG schemes, a positivity limiter has been developed in [28] which preserves accuracy in smooth regions. This scheme is built on a positive first-order finite volume scheme. Consider the first-order version of the ALE-DG scheme which is a finite volume scheme given by

$$h_j^{n+1} \bar{u}_j^{n+1} = h_j^n \bar{u}_j^n - \Delta t_n [\hat{g}_{j+\frac{1}{2}}^n - \hat{g}_{j-\frac{1}{2}}^n]. \tag{9}$$

The only degree of freedom is the cell average value and the solution is piecewise constant. We will analyze the positivity of this scheme for the case of the Rusanov flux which is given in 1. The update equation can be rewritten as

$$\begin{aligned} h_j^{n+1} \bar{u}_j^{n+1} &= \left[ h_j^n - \frac{\Delta t_n}{2} \left( \lambda_{j-\frac{1}{2}}^n + w_{j-\frac{1}{2}}^n + \lambda_{j+\frac{1}{2}}^n - w_{j+\frac{1}{2}}^n \right) \right] \bar{u}_j^n \\ &\quad + \frac{\Delta t_n}{2} \left[ \left( \lambda_{j-\frac{1}{2}}^n - w_{j-\frac{1}{2}}^n \right) \bar{u}_{j-1}^n + f_{j-1}^n \right] \\ &\quad + \frac{\Delta t_n}{2} \left[ \left( \lambda_{j+\frac{1}{2}}^n + w_{j+\frac{1}{2}}^n \right) \bar{u}_{j+1}^n - f_{j+1}^n \right] \\ &= a_j^n \bar{u}_j^n + \frac{\Delta t_n}{2} B_j^n + \frac{\Delta t_n}{2} C_j^n. \end{aligned}$$

From the definition of the Rusanov flux formula, we can easily see that<sup>2</sup>

$$\left( \lambda_{j-\frac{1}{2}} - w_{j-\frac{1}{2}} \right) + v_{j-1} \geq c_{j-1} > 0, \quad \left( \lambda_{j+\frac{1}{2}} + w_{j+\frac{1}{2}} \right) - v_{j+1} \geq c_{j+1} > 0.$$

<sup>2</sup> We drop the superscript  $n$  in some of these expressions.

Consider the first component of  $B_j^n$

$$\begin{aligned} \left(\lambda_{j-\frac{1}{2}} - w_{j-\frac{1}{2}}\right)\rho_{j-1} + \rho_{j-1}v_{j-1} &\geq \left(|v_{j-1} - w_{j-\frac{1}{2}}| + c_{j-1} - w_{j-\frac{1}{2}} + v_{j-1}\right)\rho_{j-1} \\ &\geq c_{j-1}\rho_{j-1} > 0. \end{aligned}$$

Consider the first component of  $C_j^n$

$$\begin{aligned} \left(\lambda_{j+\frac{1}{2}} + w_{j+\frac{1}{2}}\right)\rho_{j+1} - \rho_{j+1}v_{j+1} &\geq (|v_j - w_{j+\frac{1}{2}}| + c_{j+1} + w_{j-1} - v_{j+1})\rho_{j+1} \\ &\geq c_{j+1}\rho_{j+1} > 0. \end{aligned}$$

The pressure corresponding to  $B_j^n$  is

$$p_{j-1} \left( -p_{j-1} + \frac{2\rho_{j-1}(v_{j-1} + \lambda_{j-\frac{1}{2}} - w_{j-\frac{1}{2}})^2}{\gamma - 1} \right) \geq p_{j-1}\rho_{j-1}c_{j-1}^2 \frac{\gamma + 1}{\gamma - 1} \geq 0,$$

and similarly the pressure corresponding to  $C_j^n$  is non-negative. Hence, if the coefficient in term  $a_j^n$  is positive, then the scheme is positive. This requires the CFL condition

$$\Delta t_n \leq \frac{2h_j^n}{\lambda_{j-\frac{1}{2}}^n + w_{j-\frac{1}{2}}^n + \lambda_{j+\frac{1}{2}}^n - w_{j+\frac{1}{2}}^n}.$$

The time step will also be restricted to ensure that the cell size does not change too much in one time step. If we demand that the cell size does not change by more than a fraction  $\beta \in (0, 1)$ , then we need to ensure that the time step satisfies

$$\Delta t_n \leq \frac{\beta h_j^n}{|w_{j+\frac{1}{2}}^n - w_{j-\frac{1}{2}}^n|}.$$

Combining the previous two conditions, we obtain the following condition on the time step:

$$\Delta t_n \leq \Delta t_n^{(1)} := \min_j \left\{ \frac{\left(1 - \frac{1}{2}\beta\right)h_j^n}{\frac{1}{2}\left(\lambda_{j-\frac{1}{2}}^n + \lambda_{j+\frac{1}{2}}^n\right)}, \frac{\beta h_j^n}{|w_{j+\frac{1}{2}}^n - w_{j-\frac{1}{2}}^n|} \right\}. \tag{10}$$

We can now state the following result on the positivity of the first-order finite volume scheme on moving meshes.

**Theorem 1** *The scheme (9) with the Rusanov flux is positivity preserving if the time step condition (10) is satisfied.*

**Remark 3** In this work, we have not attempted to prove the positivity of the scheme for other numerical fluxes. We also do not have a proof of positivity for higher order version of the scheme. In the computations, we use the positivity preserving limiter of [28] which leads to robust schemes which preserve the positivity of the cell average value in all the test cases.

### 7 Preservation of Constant States

An important property of schemes on moving meshes is their ability to preserve constant states for any mesh motion. This is related to the conservation of cell volumes in relation to the mesh motion. In our scheme, if we start with a constant state  $\mathbf{u}_h^n = \mathbf{c}$ , then the predictor is also constant in the space–time interval, i.e.,  $\mathbf{U}_h = \mathbf{c}$ . The space–time terms in (6) are polynomials with degree  $k + 1$  in space and degree one in time and these are exactly integrated by the chosen quadrature rule. The flux terms at cell boundaries in (6) are of degree one in time and these are also exactly integrated. Hence, the scheme (6) can be written as

$$\begin{aligned}
 h_j^{n+1} \mathbf{u}_{j,l}^{n+1} &= h_j^n \mathbf{u}_{j,l}^n + \int_{t_n}^{t_{n+1}} \int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)} \mathbf{g}(\mathbf{c}, w) \frac{\partial}{\partial x} \varphi_l(x, t) dx dt \\
 &+ \int_{t_n}^{t_{n+1}} \left[ \hat{\mathbf{g}}_{j-\frac{1}{2}}(t) \varphi_l \left( x_{j-\frac{1}{2}}^+, t \right) - \hat{\mathbf{g}}_{j+\frac{1}{2}}(t) \varphi_l \left( x_{j+\frac{1}{2}}^-, t \right) \right] dt,
 \end{aligned}$$

where  $\mathbf{u}_{j,0}^n = \mathbf{c}$  and  $\mathbf{u}_{j,l}^n = 0$  for  $l = 1, 2, \dots, k$ . Due to the constant predictor and by consistency of the numerical flux

$$\hat{\mathbf{g}}_{j+\frac{1}{2}}(t) = \mathbf{f}(\mathbf{c}) - w_{j+\frac{1}{2}}^n \mathbf{c}.$$

Moreover, for  $l = 1, 2, \dots, k$ ,

$$\begin{aligned}
 &\int_{t_n}^{t_{n+1}} \int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)} \mathbf{g}(\mathbf{c}, w) \frac{\partial}{\partial x} \varphi_l(x, t) dx dt + \int_{t_n}^{t_{n+1}} \left[ \hat{\mathbf{g}}_{j-\frac{1}{2}}(t) \varphi_l \left( x_{j-\frac{1}{2}}^+, t \right) - \hat{\mathbf{g}}_{j+\frac{1}{2}}(t) \varphi_l \left( x_{j+\frac{1}{2}}^-, t \right) \right] dt \\
 &= \int_{t_n}^{t_{n+1}} \int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)} \frac{\partial}{\partial x} \mathbf{g}(\mathbf{c}, w) \varphi_l(x, t) dx dt + \int_{t_n}^{t_{n+1}} \left[ \hat{\mathbf{g}}_{j-\frac{1}{2}}(t) \varphi_l \left( x_{j-\frac{1}{2}}^+, t \right) - \hat{\mathbf{g}}_{j+\frac{1}{2}}(t) \varphi_l \left( x_{j+\frac{1}{2}}^-, t \right) \right] dt \\
 &\quad - \int_{t_n}^{t_{n+1}} \int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)} \varphi_l(x, t) \frac{\partial}{\partial x} \mathbf{g}(\mathbf{c}, w) dx dt \\
 &= - \int_{t_n}^{t_{n+1}} \frac{\partial}{\partial x} \mathbf{g}(\mathbf{c}, w) \left( \int_{x_{j-\frac{1}{2}}(t)}^{x_{j+\frac{1}{2}}(t)} \varphi_l(x, t) dx \right) dt = 0,
 \end{aligned}$$

where we have used the property that  $w$  is an affine function of  $x$  and  $\varphi_l$  are orthogonal. This implies that  $\mathbf{u}_{j,l}^{n+1} = 0$  for  $l = 1, 2, \dots, k$ . For  $l = 0$ , we get

$$h_j^{n+1} \mathbf{u}_{j,0}^{n+1} = h_j^n \mathbf{c} + \int_{t_n}^{t_{n+1}} \left[ \hat{\mathbf{g}}_{j-\frac{1}{2}}(t) - \hat{\mathbf{g}}_{j+\frac{1}{2}}(t) \right] dt = \left[ h_j^n + \left( w_{j-\frac{1}{2}}^n - w_{j+\frac{1}{2}}^n \right) \Delta t \right] \mathbf{c},$$

and since  $h_j^{n+1} = h_j^n + \left( w_{j-\frac{1}{2}}^n - w_{j+\frac{1}{2}}^n \right) \Delta t$ , we obtain  $u_{j,0} = c$  which implies that  $u_h^{n+1} = c$ .

### 8 Grid Coarsening and Refinement

The size of the cells can change considerably during the time evolution process due to the near Lagrangian movement of the cell boundaries. Near shocks, the cells will be compressed to smaller sizes which will reduce the allowable time step since a CFL condition has to be satisfied. In some regions, e.g., inside expansion fans, the cell size can increase considerably which may lead to loss of accuracy. To avoid too small or too large cells from occurring in the grid, we implement cell merging and refinement into our scheme. If a cell becomes smaller than some specified size  $h_{\min}$ , then it is merged with one of its neighbouring cells and the solution is transferred from the two cells to the new cell by performing an  $L^2$  projection. If a cell size becomes larger than some specified size  $h_{\max}$ , then this cell is refined into two cells by division and the solution is again transferred by the  $L^2$  projection. The use of the  $L^2$  projection for solution transfer ensures the conservation of mass, momentum and energy and preserves the accuracy in smooth regions. We also ensure that the cell sizes do not change drastically between neighbouring cells. To keep a track of refinement of cells, each cell is assigned an initial level equal to 0. The daughter cells created during refinement are assigned a level incremented from the parent cell, while the coarsened cells are assigned a level decremented from the parent cell.

The algorithm for refinement and coarsening is carried out in three sweeps over all the active cells. In the first sweep, we mark the cells for refinement or coarsening based on their size and the level of neighboring cells. Cells are marked for coarsening if the size is less than a pre-specified minimum size. They are marked for refinement if either the size of the cell is larger than the maximum size or if the level of the cell is less than the level of the neighboring cells. If none of the conditions are satisfied, the cells are marked for no change. In the second sweep, a cell is marked for refinement if both the neighboring cells are marked for refinement. A cell is also marked for refinement if the size of the cell is larger than twice the size of either of the neighboring cells, and is also larger than twice the minimum size. The last condition is inserted to prevent a cell being alternately marked for refinement and coarsening in consecutive adaptation cycles. In the third and final sweeps, we again mark cells for refinement if both the neighboring cells are marked for refinement. Further, we ensure that a cell marked for refinement does not have a neighboring cell marked for a coarsening, since this can lead to an inconsistent mesh.

### 9 Numerical Results

The numerical tests are performed with polynomials of degree one, two and three, together with the linear Taylor expansion, two stage CERK and four stage CERK, respectively, for the computation of the predictor. For the quadrature in time, we use the mid-point rule, two- and three-point Gauss–Legendre quadrature, respectively. The time step is chosen using the CFL condition,

$$\Delta t_n = \frac{\text{CFL}}{2k + 1} \Delta t_n^{(1)},$$

**Table 1** Order of accuracy study on static mesh using Rusanov flux

$N$	$k = 1$		$k = 2$		$k = 3$	
	Error	Rate	Error	Rate	Error	Rate
100	4.370E-02	–	3.498E-03	–	3.883E-04	–
200	6.611E-03	2.725	4.766E-04	2.876	1.620E-05	4.583
400	1.332E-03	2.518	6.415E-05	2.885	9.376E-07	4.347
800	3.151E-04	2.372	8.246E-06	2.910	5.763E-08	4.239
1 600	7.846E-05	2.280	1.031E-06	2.932	3.595E-09	4.180

**Table 2** Order of accuracy study on moving mesh using Rusanov flux

$N$	$k = 1$		$k = 2$		$k = 3$	
	Error	Rate	Error	Rate	Error	Rate
100	2.331E-02	–	3.979E-03	–	8.633E-04	–
200	6.139E-03	1.9250	4.058E-04	3.294	1.185E-05	6.186
400	1.406E-03	2.0258	5.250E-05	3.122	7.079E-07	5.126
800	3.375E-04	2.0366	6.626E-06	3.077	4.340E-08	4.760
1 600	8.278E-05	2.0344	8.304E-07	3.057	2.689E-09	4.573

where  $\Delta t_n^{(1)}$  is given by (10), and the factor  $(2k + 1)$  comes from linear stability analysis [27]. In most of the computations, we use  $CFL = 0.9$  unless stated otherwise. We observe that the results using the average or linearized Riemann velocity are quite similar. We use the average velocity for most of the results and show the comparison between the two velocities for some results. The main steps in the algorithm within one time step  $t_n \rightarrow t_{n+1}$  are as follows:

- i. choose the mesh velocity  $w_{j+\frac{1}{2}}$ ;
- ii. choose the time step  $\Delta t_n$ ;
- iii. compute the predictor  $\bar{U}_h$ ;
- iv. update the solution  $u_h^n$  to the next time level  $u_h^{n+1}$ ;
- v. apply the TVD/TVB limiter on  $u_h^{n+1}$ ;
- vi. apply the positivity limiter on  $u_h^{n+1}$  from [28];
- vii. perform grid refinement/coarsening.

In all the solution plots given below, symbols denote the cell average value.

### 9.1 Order of Accuracy

We study the convergence rate of the schemes by applying them to a problem with a known smooth solution. The initial condition is taken as

$$\rho(x, 0) = 1 + \exp(-10x^2), \quad u(x, 0) = 1, \quad p(x, 0) = 1,$$

whose exact solution is  $\rho(x, t) = \rho(x - t, 0)$ ,  $u(x, t) = 1$ ,  $p(x, t) = 1$ . The initial domain is  $[-5, +5]$  and the final time is  $t = 1$  units. The results are presented using Rusanov and



**Table 3** Order of accuracy study on static mesh using HLLC flux

$N$	$k = 1$		$k = 2$		$k = 3$	
	Error	Rate	Error	Rate	Error	Rate
100	4.582E-02	–	3.952E-03	–	3.464E-04	–
200	9.611E-03	2.253	4.048E-04	3.287	2.058E-05	4.073
400	2.052E-03	2.240	4.640E-05	3.206	1.287E-06	4.036
800	4.803E-04	2.192	5.623E-06	3.152	8.061E-08	4.023
1 600	1.184E-04	2.149	6.929E-07	3.119	5.050E-09	4.016

**Table 4** Order of accuracy study on moving mesh using HLLC flux

$N$	$k = 1$		$k = 2$		$k = 3$	
	Error	Rate	Error	Rate	Error	Rate
100	1.590E-02	–	1.626E-03	–	1.962E-04	–
200	4.042E-03	1.977	2.072E-04	2.972	1.269E-05	3.950
400	1.014E-03	1.985	2.605E-05	2.982	7.983E-07	3.971
800	2.538E-04	1.990	3.261E-06	2.988	4.997E-08	3.980
1 600	6.349E-05	1.992	4.077E-07	2.991	3.124E-09	3.985

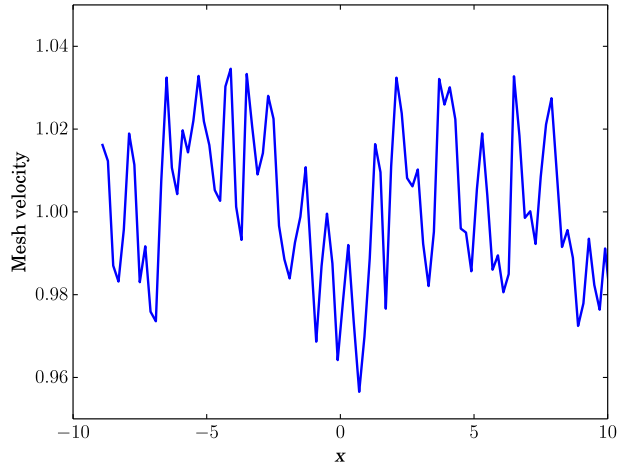
**Table 5** Order of accuracy study on moving mesh using Rusanov flux using higher order limiter [29]

$N$	$k = 1$		$k = 2$	
	Error	Rate	Error	Rate
100	2.053E-02	–	2.277E-03	–
200	4.312E-03	2.251	3.425E-04	2.732
400	1.031E-03	2.064	4.565E-05	2.907
800	2.550E-04	2.015	5.812E-06	2.973
1 600	6.356E-05	2.004	7.315E-07	2.990

HLLC numerical fluxes. The  $L^2$  norm of the error in density is shown in Tables 1 and 3 for the static mesh and in Tables 2 and 4 for the moving mesh. In each case, we see that the error behaves as  $O(h^{k+1})$  which is the optimal rate we can expect for smooth solutions. In Table 5, we show that the ALE DG methods preserve its higher order in the presence of a limiter.

The mesh velocity is constant since the fluid velocity is constant. To study the effect of perturbations in mesh velocity, we add a random perturbation to each mesh velocity,  $w_{j+\frac{1}{2}} \leftarrow (1 + \alpha r_{j+\frac{1}{2}})w_{j+\frac{1}{2}}$  where  $r_{j+\frac{1}{2}}$  is a uniform random variable in  $[-1, +1]$  and  $\alpha = 0.05$  and a sample velocity distribution is shown in Fig. 5. Note that this randomization is performed in each time step with different random variables drawn for each face. For the moving mesh, there is no unique cell size and the convergence rate is computed based on initial mesh spacing which is inversely proportional to the number of cells. From Table 6 which shows results using HLLC flux, we again observe that the error reduces at the optional rate of  $k + 1$  even when the mesh velocity is not very smooth.

**Fig. 5** Example of randomized velocity distribution for smooth test case



**Table 6** Order of accuracy study on moving mesh using HLLC flux with randomly perturbed mesh velocity

<i>N</i>	<i>k</i> = 1		<i>k</i> = 2		<i>k</i> = 3	
	Error	Rate	Error	Rate	Error	Rate
100	1.735E-02	–	1.798E-03	–	2.351E-04	–
200	4.179E-03	2.051	2.848E-04	2.676	1.416E-05	4.069
400	1.054E-03	2.035	4.301E-05	2.703	8.578E-07	4.041
800	2.615E-04	1.943	6.012E-06	2.838	5.476E-08	3.958
1 600	7.279E-05	1.852	8.000E-07	2.909	3.505E-09	3.966

**Table 7** Order of accuracy study on fixed mesh using Roe flux with non-constant velocity smooth test case

<i>N</i>	<i>k</i> = 1		<i>k</i> = 2	
	Error	Rate	Error	Rate
100	8.535E-03	–	1.033E-03	–
200	1.958E-03	2.124	1.221E-04	3.08
400	4.721E-04	2.052	1.581E-05	2.95
800	1.238E-04	1.931	2.14E-06	2.89
1 600	3.563E-05	1.796	2.63E-07	3.02

**9.2 Smooth Test Case with Non-constant Velocity**

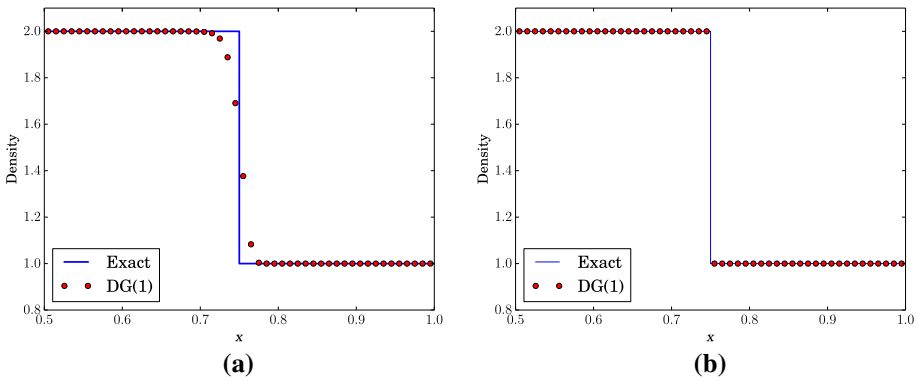
We also test the accuracy of our schemes on a isentropic problem with smooth solutions. In the test case, the initial conditions are given by

$$\rho(x, 0) = 1 + 0.9999995 \sin(\pi x), \quad u(x, 0) = 0, \quad p(x, 0) = \rho^\gamma(x, 0) \tag{11}$$

with  $\gamma = 3$  and periodic boundary conditions. For this kind of special isentropic problem, the Euler equations are equivalent to the two Burgers equations in terms of their two Riemann invariants which can then be used to derive the analytical solution. The errors are then computed with respect to the given analytical solution. In contrast to the previous test

**Table 8** Order of accuracy study on moving mesh using Roe flux with non-constant velocity smooth test case

$N$	$k = 1$		$k = 2$	
	Error	Rate	Error	Rate
100	4.235E-03	–	2.238E-04	–
200	1.058E-03	2.001	3.255E-05	2.87
400	2.586E-04	2.035	4.301E-05	3.133
800	5.804E-05	2.155	5.762E-06	2.901
1 600	1.271E-05	2.192	7.401E-07	2.96



**Fig. 6** Single contact wave using Roe flux and 100 cells: **a** static mesh, **b** moving mesh

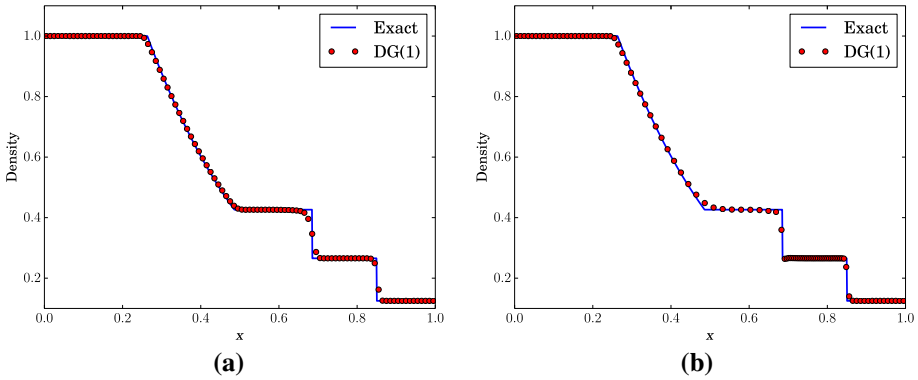
case, the velocity and pressure are not constant which makes this a more challenging test case. We run the simulation with a WENO-type limiter from [28] and the positivity limiter enabled. As we can see from Tables 7 and 8, the rate of convergence is maintained for the moving mesh method with the moving mesh methods exhibiting much lower errors.

### 9.3 Single Contact Wave

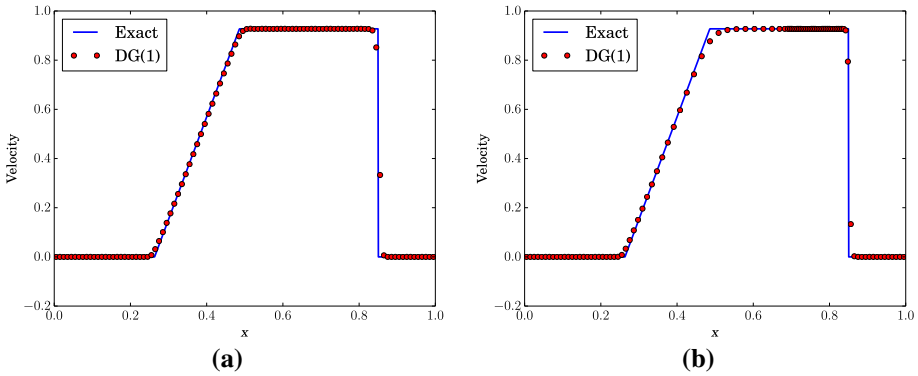
In this example, we choose a Riemann problem which gives rise to a single contact wave in the solution that propagates with a constant speed. The initial condition is given by

$$(\rho, v, p) = \begin{cases} (2.0, 1.0, 1.0), & \text{if } x < 0.5, \\ (1.0, 1.0, 1.0), & \text{if } x > 0.5, \end{cases}$$

and the contact wave moves with a constant speed of 1.0. The solutions on static and moving meshes are shown in Fig. 6 at time  $t = 0.5$  using the Roe flux. The moving mesh is able to exactly resolve the contact wave while the static mesh scheme adds considerable numerical dissipation that smears the discontinuity over many cells. The accurate resolution of contact waves is a key advantage of such moving mesh methods, which are capable of giving very good resolution of the contact discontinuity even on coarse meshes.



**Fig. 7** Sod problem using Roe flux, 100 cells and TVD limiter: **a** static mesh, **b** moving mesh



**Fig. 8** Sod problem using Roe flux, 100 cells and TVD limiter: **a** static mesh, **b** moving mesh

### 9.4 Sod Problem

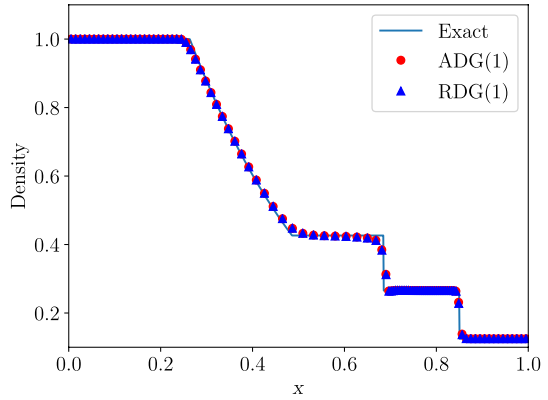
The initial condition for the Sod test case is given by [30]

$$(\rho, v, p) = \begin{cases} (1.0, 0, 1.0), & \text{if } x < 0.5, \\ (0.125, 0, 0.1), & \text{if } x > 0.5, \end{cases}$$

and the solution is computed up to a final time of  $T = 0.2$  with the computational domain being  $[0, 1]$ . Since the fluid velocity is zero at the boundary, the computational domain does not change with time for the chosen final time. The exact solution consists of a rarefaction fan, a contact wave and a shock wave. In Fig. 7, we show the results obtained using the Roe flux with 100 cells and the TVD limiter on static and moving meshes. The contact wave is considerably well resolved on the moving mesh as compared to the static mesh due to reduced numerical dissipation on moving meshes (Figs. 8 and 9).

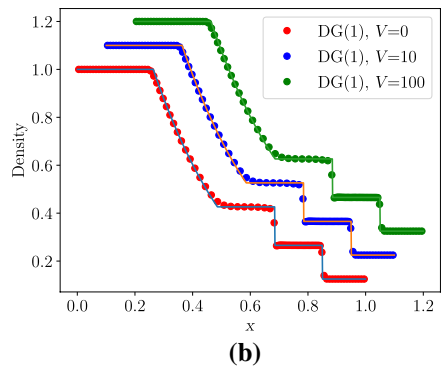
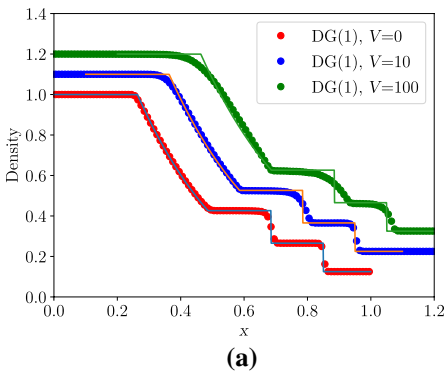
To study the Galilean invariance or the dependence of the solution on the choice of coordinate frame, we add a boost velocity of  $V = 10$  or  $V = 100$  to the coordinate frame, which implies that the initial fluid velocity is  $v(x, 0) = V$  and the other quantities remain as before. Figure 10a shows that the accuracy of the static mesh results degrades

**Fig. 9** Sod problem using Roe flux, 100 cells and TVD limiter. *ADG* average velocity, *RDG* linearized riemann velocity



**Table 9** Number of iterations required to reach time  $t = 0.2$  for Sod test for different boost velocity of the coordinate frame

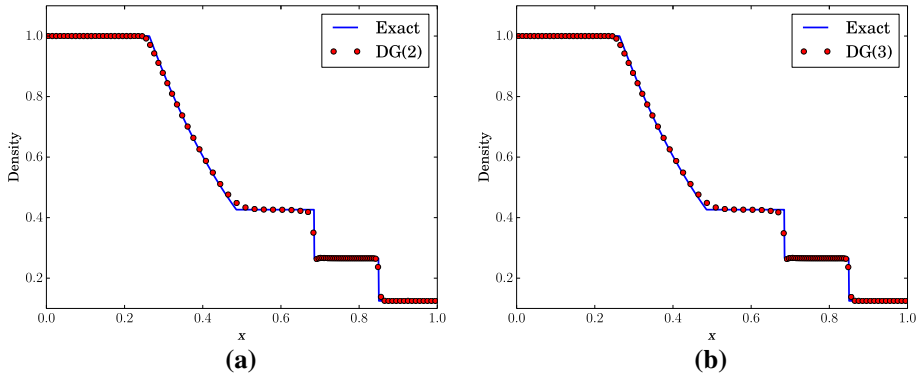
$V$	0	10	100
Static mesh	144	810	6 807
Moving mesh	176	176	176



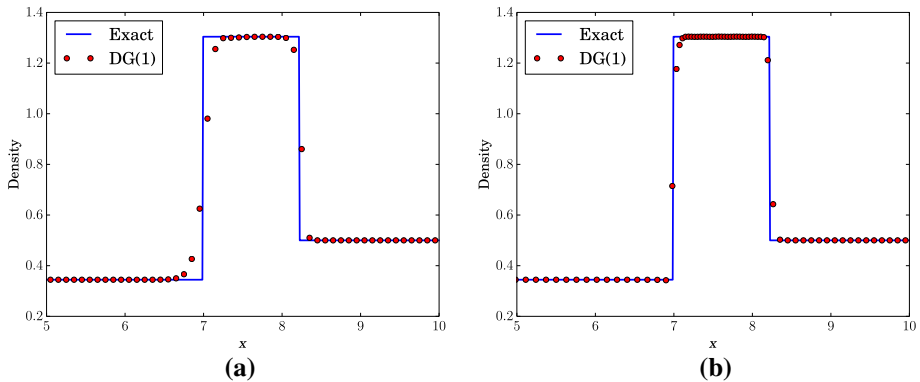
**Fig. 10** Effect of coordinate frame motion on Sod problem using Roe flux, 100 cells and TVD limiter: **a** static mesh, **b** moving mesh

with increase in velocity of the coordinate frame, particularly the contact discontinuity is highly smeared. The results given in Fig. 10b clearly show the independence of the results on the moving mesh with respect to the coordinate frame velocity. The allowed time step from CFL condition decreases with increase in coordinate frame speed for the static mesh case, while in case of the moving mesh, it remains invariant. This means that in case of static mesh, we have to perform more time steps to reach the same final time as shown in Table 9, which increases the computational time. Thus, the moving mesh scheme has the additional advantage of allowing a larger time step compared to the fixed mesh scheme.

Finally, we compute the solutions using quadratic and cubic polynomials and the results are shown in Fig. 11. The solutions look similar to the case of linear polynomials and have the same sharp resolution of discontinuities.



**Fig. 11** Sod problem on moving mesh using Roe flux, 100 cells and TVD limiter: **a** Degree = 2, **b** Degree = 3



**Fig. 12** Lax problem using HLLC flux, 100 cells and TVD limiter: **a** static mesh, **b** moving mesh

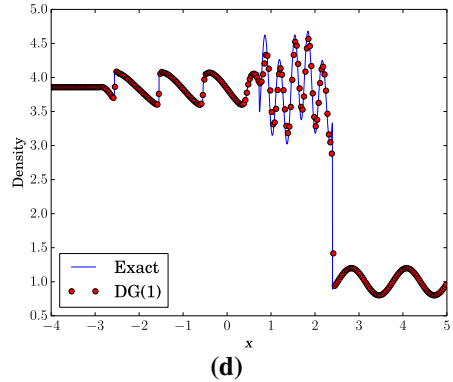
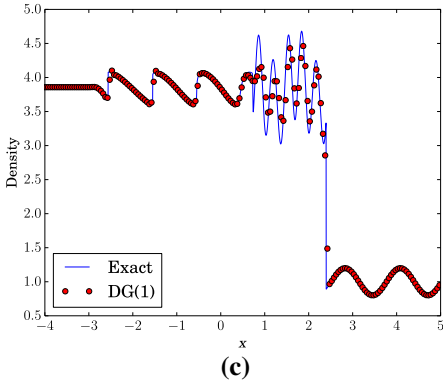
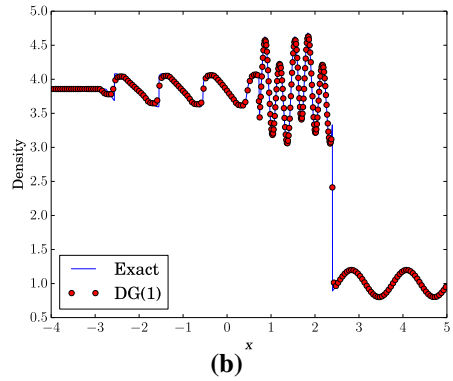
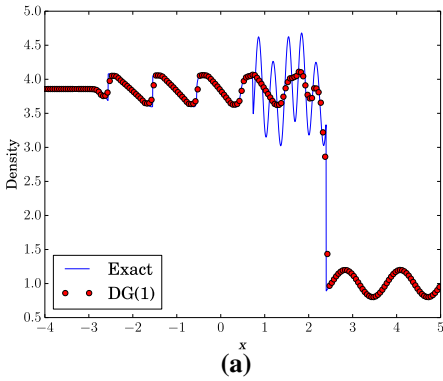
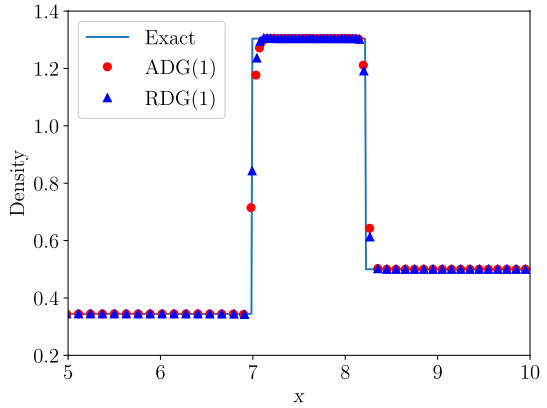
### 9.5 Lax Problem

The initial condition is given by

$$(\rho, v, p) = \begin{cases} (0.445, 0.698, 3.528), & \text{if } x < 0, \\ (0.5, 0, 0.571), & \text{if } x > 0. \end{cases}$$

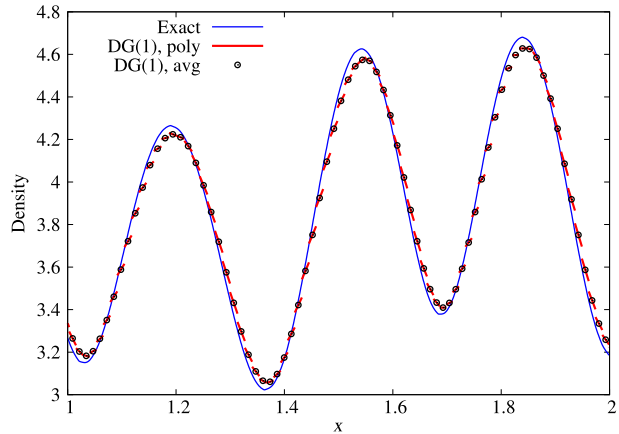
The computational domain is  $[-10, +10]$  and we compute the solution up to a final time of  $T = 1.3$ . This problem has a strong shock and a contact wave that is difficult to resolve accurately. The zoomed view of density is shown at the final time in Fig. 12, and we observe that the moving mesh results are more accurate for the contact wave, which is the first discontinuity in the figure. The second discontinuity is a shock which is equally well resolved in both cases. We can observe that the grid is automatically clustered in the region between the contact and shock wave, but no explicit grid adaptation was used in this simulation (Fig. 13).

**Fig. 13** Lax problem using HLLC flux, 100 cells and TVD limiter. *ADG* average velocity, *RDG* linearized Riemann velocity



**Fig. 14** Shu–Osher problem using Roe flux: **a** static mesh, 200 cells,  $M = 0$ , **b** moving mesh, 200 cells,  $M = 0$ , **c** static mesh, 200 cells,  $M = 100$ , **d** static mesh, 300 cells,  $M = 100$

**Fig. 15** Shu–Osher problem using Roe flux on moving mesh



## 9.6 Shu–Osher Problem

The initial condition is given by [31]

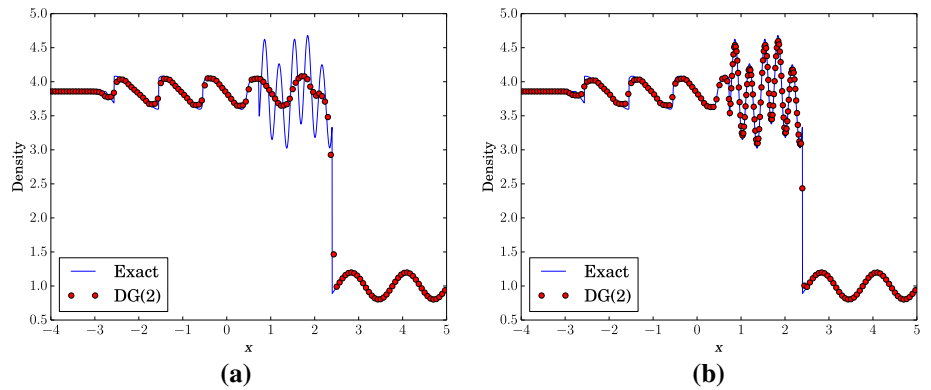
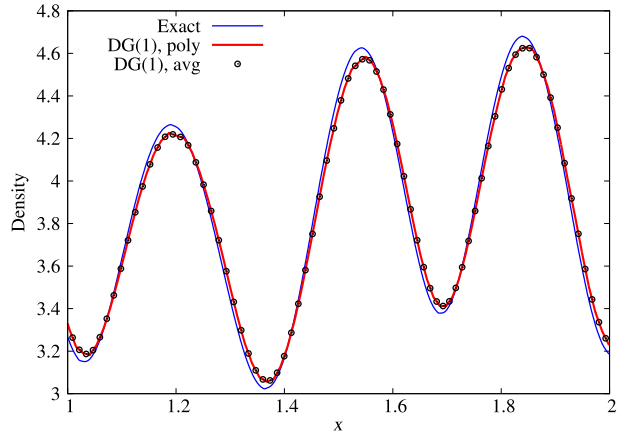
$$(\rho, v, p) = \begin{cases} (3.857143, 2.629369, 10.333333), & \text{if } x < -4, \\ (1 + 0.2 \sin(5x), 0, 1.0), & \text{if } x > -4, \end{cases}$$

which involves a smooth sinusoidal density wave which interacts with a shock. The domain is  $[-5, +5]$  and the solution is computed up to a final time of  $T = 1.8$ . The solutions are shown in Fig. 14a, b on static and moving meshes using 200 cells and the TVD limiter. The moving mesh scheme is considerably more accurate in resolving the sinusoidal wave structure that arises after interaction with the shock. In Fig. 14c, we compute the solution on static mesh with the TVB limiter and the parameter  $M = 100$  in (8). In this case, the solutions on static mesh are more accurate compared to the case of the TVD limiter but still not as good as the moving mesh results. The moving mesh result has more than 200 cells in the interval  $[-5, +5]$  at the final time since cells enter the domain from the left side. Hence in Fig. 14d, we show the static mesh results with 300 cells and using the TVB limiter. The results are further improved for the static mesh case but still not as accurate as the moving mesh case. The choice of parameters in the TVB limiter is very critical but we do not have a rigorous algorithm to choose a good value for this. Hence, it is still advantageous to use the moving mesh scheme which gives improved solutions even with the TVD limiter.

The above results show that the ALE method is very accurate in terms of the cell averages. In Fig. 15, we show a zoomed view of density and pressure, where we also plot the linear polynomial solution. The slope of the solution is not accurately predicted with the Roe scheme and there are spurious contact discontinuities as the pressure and velocity are nearly continuous. This behaviour is observed with all contact preserving fluxes like Roe, HLLC and HLL-CPS but not with the Rusanov flux. Due to the almost Lagrangian character of the scheme, the eigenvalue corresponding to the contact wave,  $\lambda_2 = v - w$ , is nearly zero, which leads to the loss of dissipation in the corresponding characteristic field. If a spurious contact wave is generated during the violent dynamics, then this wave will be preserved by the scheme leading to wrong solutions. We modify the Roe scheme by preventing this eigenvalue from becoming too small or zero, which is similar to the approach



**Fig. 16** Shu–Osher problem using modified Roe flux on moving mesh



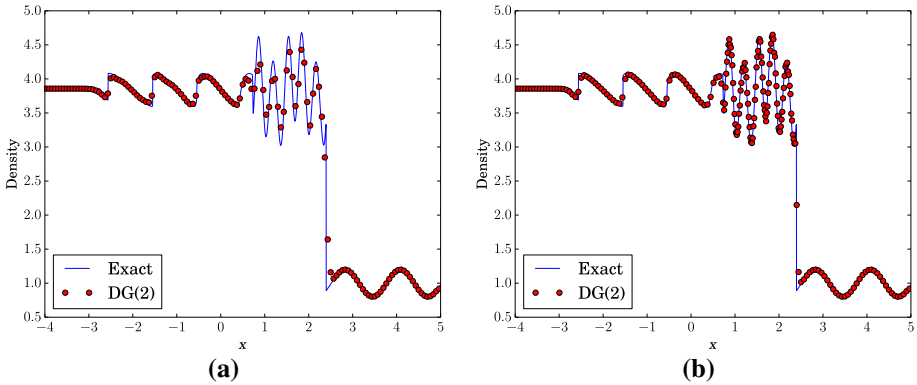
**Fig. 17** Shu–Osher problem using the modified Roe flux, the TVD limiter, quadratic polynomials and 150 cells. **a** static mesh, **b** moving mesh

used for the entropy fix. The eigenvalue  $|\lambda_2|$  used in the dissipative part of the Roe flux is determined from

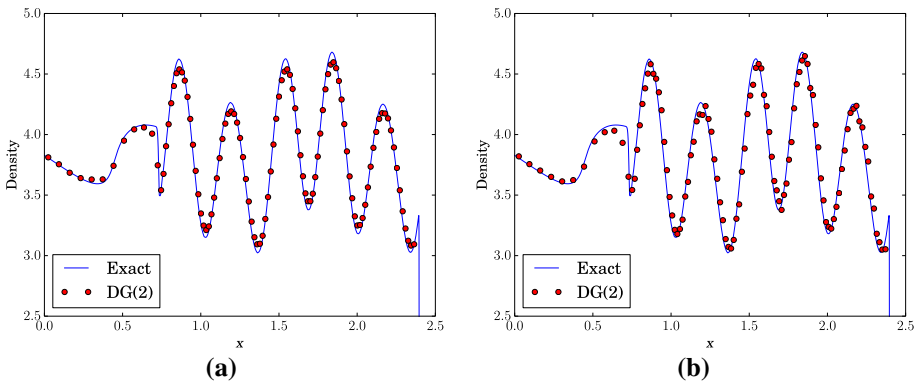
$$|\lambda_2| = \begin{cases} |v - w|, & \text{if } |v - w| > \delta = \alpha c, \\ \frac{1}{2}(\delta + |v - w|^2/\delta), & \text{otherwise.} \end{cases}$$

With this modification and using  $\alpha = 0.1$ , the solution on the moving mesh is shown in Fig. 16 and we do not observe the spurious contact discontinuities which arise with the standard Roe flux, while at the same time, the solution accuracy compares favourably with the previous results that did not use the eigenvalue fix.

We next compute the solutions using quadratic polynomials. Figure 17 shows the results obtained with the TVD limiter which shows the dramatically better accuracy that is achieved on the moving mesh compared to static mesh. In Fig. 18, we perform the same computation with a WENO limiter taken from [29]. The static mesh results are now improved over the case of the TVD limiter but still not as good as the moving mesh results in terms of capturing the extrema. In Fig. 19, we show a zoomed view of the results on the



**Fig. 18** Shu–Osher problem using modified Roe flux, WENO limiter, quadratic polynomials and 150 cells. **a** Static mesh, **b** moving mesh



**Fig. 19** Shu–Osher problem using modified Roe flux, moving mesh, quadratic polynomials and 150 cells. **a** TVD limiter, **b** WENO limiter

moving mesh with TVD and WENO limiters. We see that the TVD limiter is also able to capture all the features and is almost comparable to the WENO limiter.

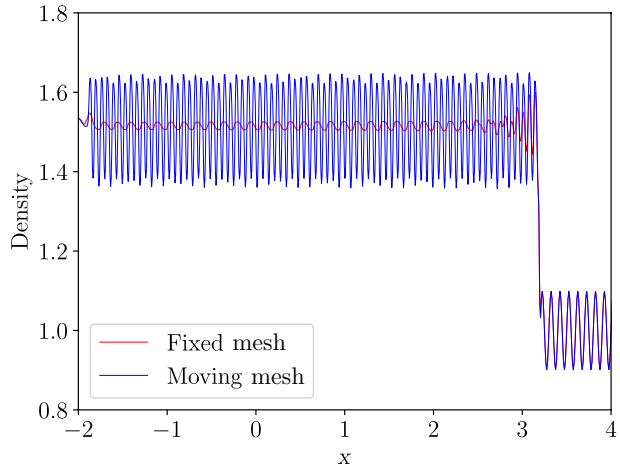
### 9.7 Titarev–Toro Problem

Titarev–Toro problem is an extension of the Shu–Osher problem [32] to test a severely oscillatory wave interacting with a shock wave. It aims to test the ability of higher order methods to capture the extremely high-frequency waves. The initial condition is given by

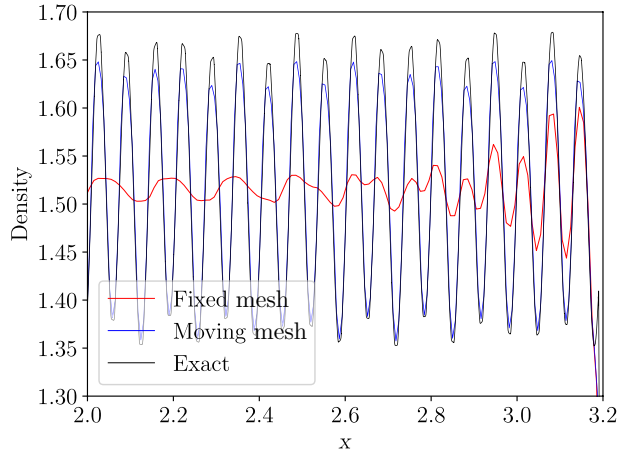
$$(\rho, v, p) = \begin{cases} (1.515\ 695, 0.523\ 346, 1.805), & -5 < x \leq -4.5, \\ (1 + 0.1 \sin(20\pi x), 0, 1), & -4.5 < x \leq 5. \end{cases} \quad (12)$$

The computation is carried out on a mesh of 1 000 cells with the final time  $T = 5$  and the density at this final time is shown in Figs. 20 and 21. The fixed mesh is not able to resolve the high-frequency oscillations due to dissipation in the fluxes and the TVD limiter, but the ALE scheme gives an excellent resolution of these high-frequency oscillations. Note that

**Fig. 20** Titarev problem with HLLC flux, 1 000 cells and TVD limiter



**Fig. 21** Titarev problem with HLLC flux, 1 000 cells and TVD limiter (zoomed version)



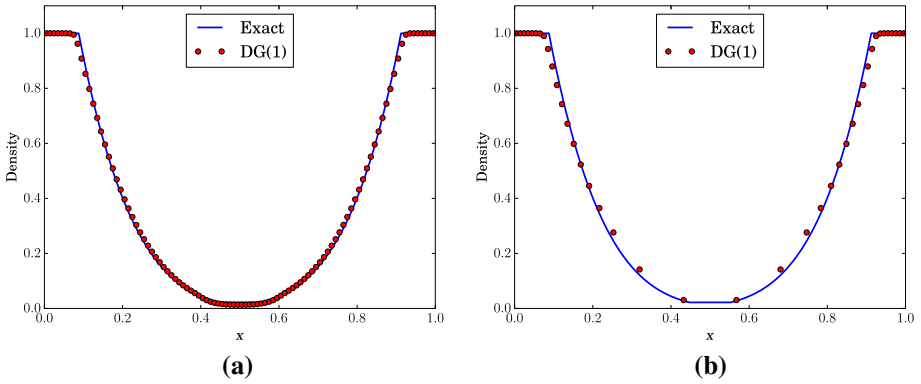
the ALE scheme also uses the same TVD limiter but it is still able to resolve the solution to a very degree of accuracy. This result again demonstrates the superior accuracy that can be achieved using a nearly Lagrangian ALE scheme in problems involving interaction of shocks and smooth flow structures.

**9.8 123 Problem**

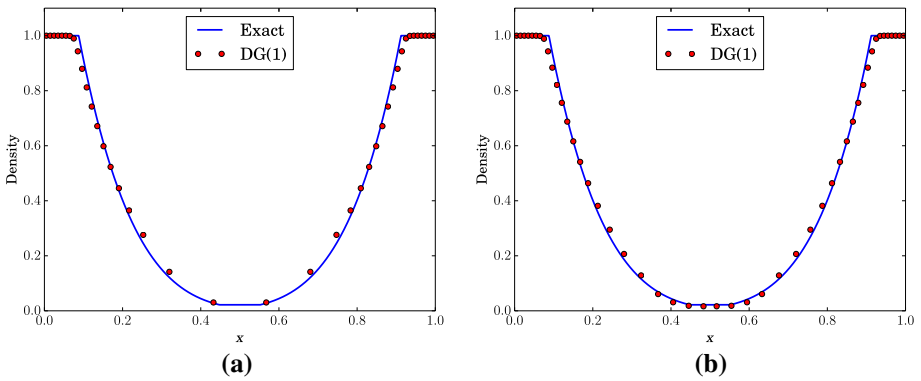
The initial condition is given by [33]

$$(\rho, v, p) = \begin{cases} (1.0, -2.0, 0.4), & x < 0.5, \\ (1.0, +2.0, 0.4), & x > 0.5. \end{cases}$$

The computational domain is [0, 1] and the final time is  $T = 0.15$ . The density using 100 cells is shown in Fig. 22 with static and moving meshes. The mesh motion does not significantly improve the solution compared to the static mesh case since the solution is smooth.

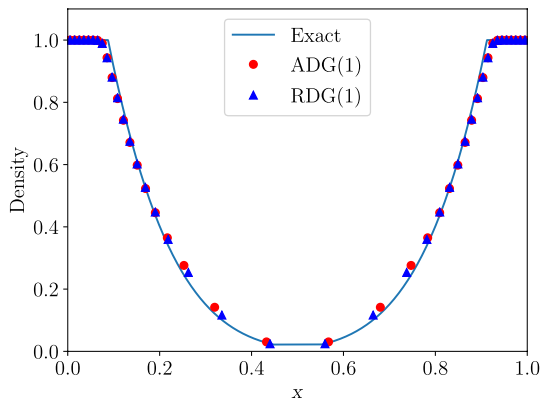


**Fig. 22** 123 problem using HLLC flux and 100 cells: **a** static mesh, **b** moving mesh

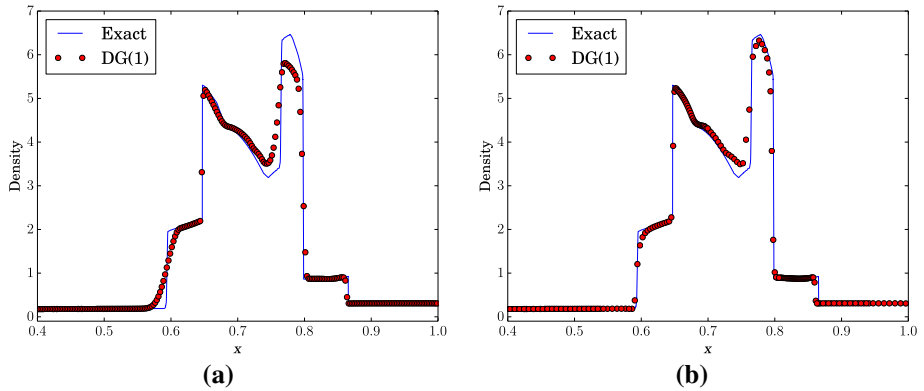


**Fig. 23** 123 problem using HLLC flux and grid refinement: **a** static mesh, **b** moving mesh with mesh adaptation ( $h_{max} = 0.05$ ) leading to 108 cells at final time

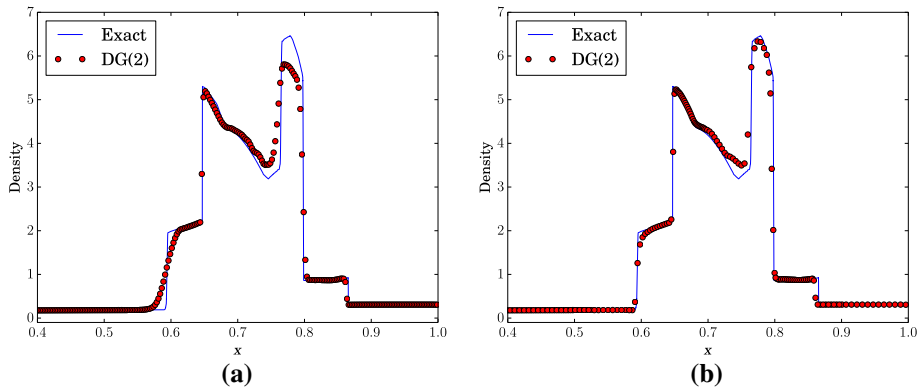
**Fig. 24** 123 problem using HLLC flux, 100 cells and TVD limiter. *ADG* average velocity, *RDG* linearized Riemann velocity



On the contrary, the mesh becomes rather coarse in the expansion region, though the solution is still well resolved. However, severe expansion may lead to very coarse meshes which may be undesirable. To prevent very coarse cells, we switch on the mesh refinement



**Fig. 25** Blast problem using HLLC flux and 400 cells. **a** Static mesh, **b** moving mesh with adaptation ( $h_{\min} = 0.001$ ) leading to 303 cells at final time



**Fig. 26** Blast problem using HLLC flux, quadratic polynomials and 400 cells. **a** Static mesh, **b** moving mesh with adaptation ( $h_{\min} = 0.001$ ) leading to 293 cells at final time

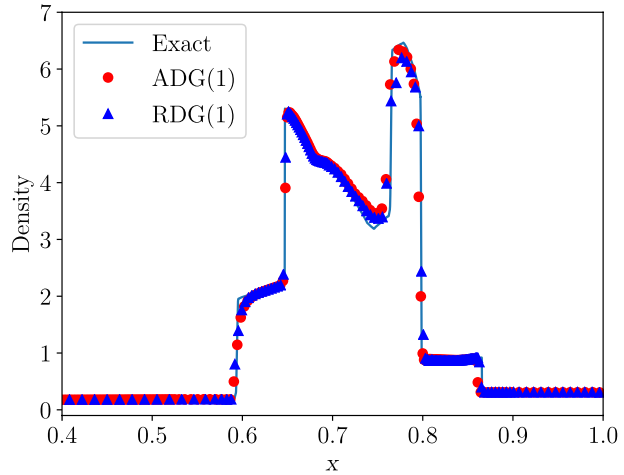
algorithm as described before and use the upper bound on the mesh size as  $h_{\max} = 0.05$ . The resulting solution is shown in Fig. 23 where the number of cells has increased to 108 at the time shown. The central expansion region is now resolved by more uniformly sized cells compared to the case of no grid refinement (Fig. 24).

### 9.9 Blast Problem

The initial condition is given by [34]

$$(\rho, v, p) = \begin{cases} (1.0, 0, 1000.0), & x < 0.1, \\ (1.0, 0, 0.01), & 0.1 < x < 0.9, \\ (1.0, 0, 100.0), & x > 0.9 \end{cases}$$

**Fig. 27** Blast problem using HLLC flux, 100 cells and TVD limiter. *ADG* average velocity, *RDG* linearized Riemann velocity



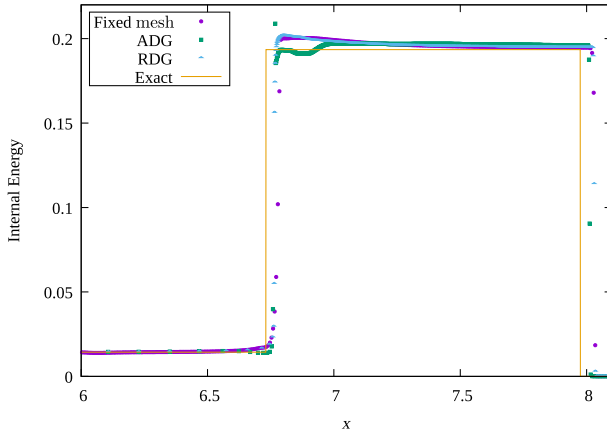
with a domain of  $[0, 1]$  and the final time is  $T = 0.038$ . A reflective boundary condition is used at  $x = 0$  and  $x = 1$ . A mesh of 400 cells is used for this simulation and in case of the moving mesh, we perform grid adaptation with  $h_{\min} = 0.001$  since some cells become very small during the collision of the two shocks. The positivity preserving limiter of [28] is applied together with the TVD limiter and HLLC flux. The static mesh results shown in Fig. 25a indicate too much numerical viscosity in the contact wave around  $x = 0.6$ . This wave is more accurately resolved in the moving mesh scheme as seen in Fig. 25b which is an advantage due to the ALE scheme and is a very good indicator of the scheme accuracy as this is a very challenging feature to compute accurately. We next compute the same problem using quadratic polynomials with all other parameters being as before. The solutions are shown in Fig. 26 and indicate that the Lagrangian moving mesh scheme is more accurate in resolving the contact discontinuity. The higher polynomial degree does not show any major improvement in the solution compared to the linear case, which could be a consequence of the strong shock interactions present in this problem, see Fig. (4.11–4.12) in [29] and Fig. (3.7) in [35] in comparison to current results (Fig. 27).

## 9.10 Le Blanc Shock Tube Test Case

The Le Blanc shock tube test case is an extreme shock tube problem where the initial discontinuity separates a region of high energy and density from one of low energy and density. This is a much more severe test than the Sod problem and hence more challenging for numerical schemes. The computational domain is  $0 \leq x \leq 9$  and is filled with an ideal gas with  $\gamma = 5/3$ . The gas is initially at rest and we perform the simulation up to a time of  $T = 6$  units. The initial discontinuity is at  $x = 3$  and the initial condition is given by

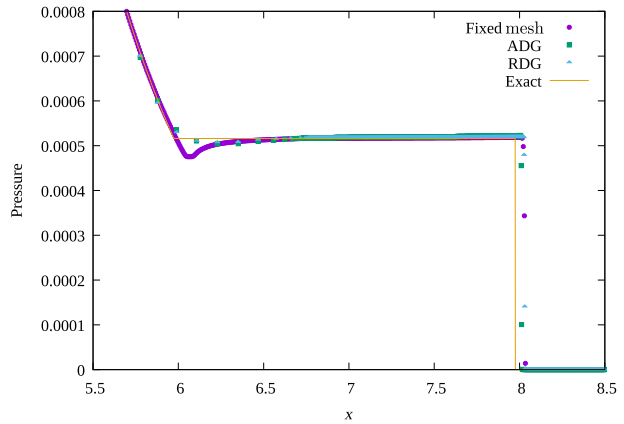
$$(\rho, v, p) = \begin{cases} (1.0, 0, 0.1), & \text{if } x < 3, \\ (0.001, 0, 10^{-7}), & \text{if } x > 3. \end{cases} \quad (13)$$

Note that both the density and pressure have a very large jump in the initial condition. The solution that develops from this initial condition consists of a rarefaction wave moving to the left and a contact discontinuity and a strong shock moving to the right. In Fig. 28, we



**Fig. 28** Internal energy for Le Blanc shock tube with Rusanov flux, 1 400 cells and TVD limiter, *ADG* average velocity, *RDG* linearized Riemann velocity

**Fig. 29** Pressure for Le Blanc shock tube with Rusanov flux, 1 400 cells and TVD limiter, *ADG* average velocity, *RDG* linearized Riemann velocity



show the comparison of the internal energy profile at final time between a fixed mesh solution and moving mesh solutions with two different mesh velocities as described before. Most methods tend to generate a very large spike in the internal energy in the contact region, e.g., compare with Fig. (11) in [36], while the present ALE method here is able to give a better profile. We plot the pressure profile in Fig. 29 which shows that the ALE scheme is able to better represent the region around the contact wave as compared to fixed mesh method.

### 9.11 Two-Dimensional Isentropic Vortex Test Case

The extension to two dimensions involves two aspects that need to be addressed. The first issue is how to handle the grid motion and the second is how to formulate the ALE-DG scheme. The second part is a natural generalization of the DG scheme we have described

**Table 10** Isentropic vortex in 2D: order of accuracy study on two-dimensional static mesh

$N$	$k = 1$		$k = 2$	
	Error	Rate	Error	Rate
$50 \times 50$	2.230E-03	–	1.762E-04	–
$100 \times 100$	5.987E-04	1.945	2.305E-05	2.934
$200 \times 200$	1.498E-04	1.998	2.973E-06	2.955
$400 \times 400$	3.786E-05	1.984	3.762E-07	2.982
$800 \times 800$	9.617E-06	1.977	3.474E-08	2.991

for the 1-D case in this paper, except that we have to construct basis functions on triangles and perform some numerical quadrature. The first part involving grid movement is more complicated and we present only very preliminary results in this section to demonstrate that the idea has merit. We now consider the two-dimensional Euler equations written as

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} + \frac{\partial g(u)}{\partial y} = 0, \tag{14}$$

where

$$u = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad f(u) = \begin{bmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ (E + p)u \end{bmatrix}, \quad g(u) = \begin{bmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ (E + p)v \end{bmatrix}, \tag{15}$$

$$p = (\gamma - 1) \left[ E - \frac{1}{2} \rho (u^2 + v^2) \right]. \tag{16}$$

The test case we consider involves an isentropic vortex that is advecting with the constant velocity and is a smooth solution for which error norms can be calculated. The test is carried out on a square domain  $[-10, 10] \times [-10, 10]$  with periodic boundary conditions. The initial condition is an isentropic vortex (Table 10)

$$T = 1 - \frac{(\gamma - 1)\beta^2}{8\gamma\pi^2} e^{1-r^2}, \tag{17}$$

$$\rho = T^{\frac{1}{\gamma-1}}, \tag{18}$$

$$u = u_\infty - \frac{\beta}{2\pi} y e^{\frac{1-r^2}{2}}, \tag{19}$$

$$v = v_\infty - \frac{\beta}{2\pi} x e^{\frac{1-r^2}{2}}, \tag{20}$$

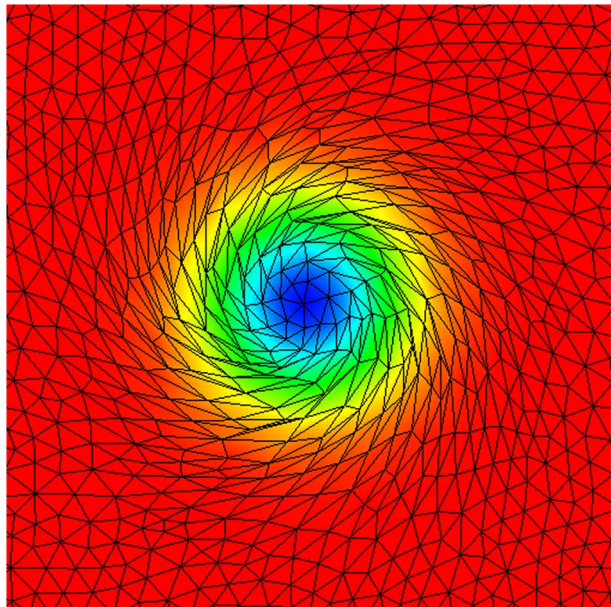
$$p = \rho^\gamma \tag{21}$$

with  $u_\infty = 1, v_\infty = 0, \gamma = 1.4, \beta = 10$ . As the solution evolves in time, the mesh becomes quite deformed because the vortex is continually shearing the mesh, which can lead to

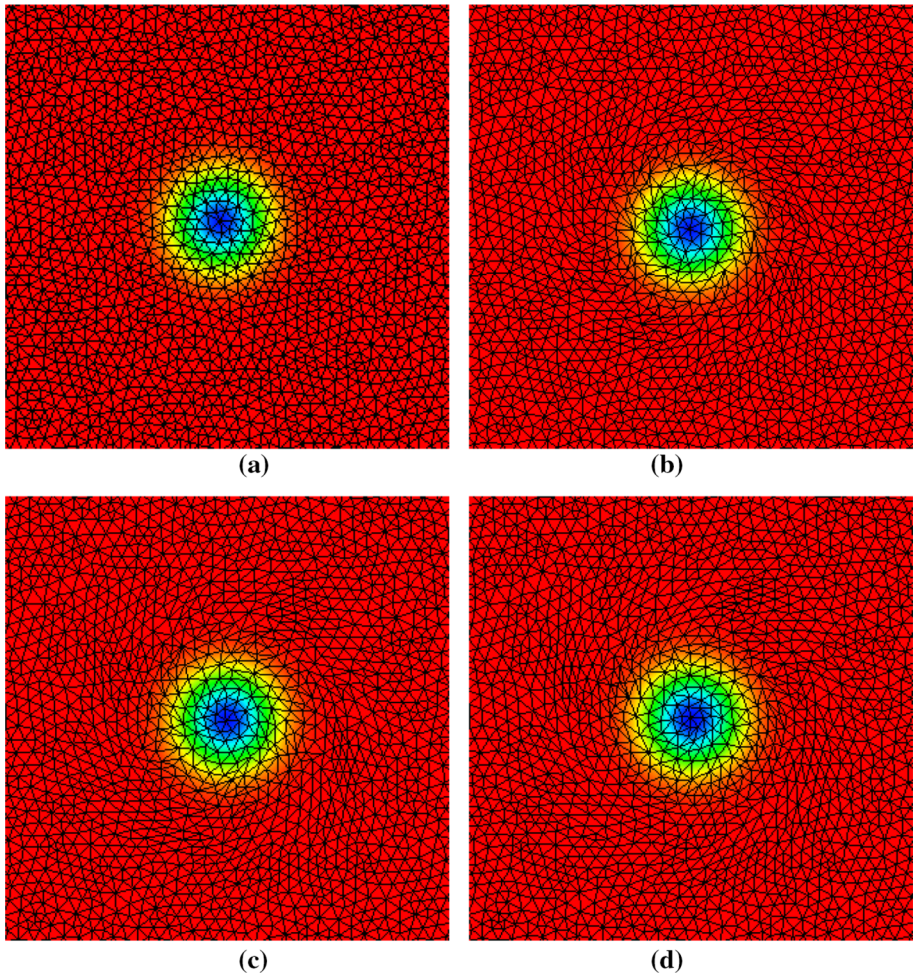


**Table 11** Isentropic vortex in 2D: order of accuracy study on two-dimensional moving mesh

$N$	$k = 1$		$k = 2$	
	Error	Rate	Error	Rate
$50 \times 50$	2.230E-03	–	1.762E-04	–
$100 \times 100$	5.987E-04	1.945	2.305E-05	2.934
$200 \times 200$	1.498E-04	1.998	2.973E-06	2.955
$400 \times 400$	3.786E-05	1.984	3.762E-07	2.982
$800 \times 800$	9.617E-06	1.977	3.474E-08	2.991

**Fig. 30** Isentropic vortex in 2-D: skewed mesh without remeshing  $t = 2.660534$ 

degenerate meshes, as shown in Fig. 30. We avoid the occurrence of badly shaped triangles using a combination of face swapping and mesh velocity smoothing algorithms [37, 38]. The mesh modification is a very local procedure and does not require global remeshing which is a costly process. With these techniques, we are able to maintain a good mesh quality even after the vortex has rotated 4 times around its center as shown in Fig. 31. As the vortex is translating, we plot the solution in a window centered at the vortex center. We can see that the method maintains its high order of accuracy from the convergence rates of the error shown in Table 11; using linear basis functions yields second-order convergence while quadratic basis functions lead to third-order convergence.



**Fig. 31** Isentropic vortex in 2-D: mesh and pressure solution at various times **a**  $t = 0$ , **b**  $t = 6$ , **c**  $t = 12$ , **d**  $t = 20$

## 10 Summary and Conclusions

We have developed an explicit DG scheme on moving meshes using ALE framework and space–time expansion of the solutions within each cell. The near Lagrangian nature of the mesh motion dramatically reduces the numerical dissipation, especially for contact waves. Even moving contact waves can be exactly computed with a numerical flux that is exact for stationary contact waves. The scheme is shown to yield superior results even in the presence of the large boost velocity of the coordinate system indicating its Galilean invariance property. The standard Roe flux does not suffer from entropy violation when applied in the current nearly Lagrangian framework. However, in some problems with strong shocks, spurious contact waves can appear and we propose to fix the dissipation in Roe-type schemes that eliminates this issue. The method yields accurate solutions even in combination with standard TVD limiters, where fixed grid methods

perform poorly. The mesh motion provides automatic grid adaptation near shocks but may lead to very coarse cells inside expansion waves. A grid adaptation strategy is developed to handle the problem of very small or very large cells. The presence of the DG polynomials makes it easy to transfer the solution during grid adaptation without loss of accuracy. The proposed methodology is general enough to be applicable to other systems of conservation laws modelling fluid flows. The basic idea can be extended to multi-dimensions but additional considerations are required to maintain good mesh quality under fluid deformations. The preliminary results shown for the isentropic vortex are very promising for the 2-D case.

**Acknowledgements** This work was started when Jayesh Badwaik was a project assistant at TIFR-CAM, Bangalore. The first two authors gratefully acknowledge the financial support received from the Airbus Foundation Chair on Mathematics of Complex Systems established in TIFR-CAM, Bangalore, for carrying out this work. Christian Klingenberg acknowledges the support of the *Priority Program 1648: Software for Exascale Computing* by the German Science Foundation. On behalf of all authors, the corresponding author states that there is no conflict of interest.

### Appendix A: Numerical Flux

The ALE scheme requires a numerical flux  $\hat{g}(\mathbf{u}_l, \mathbf{u}_r, w)$  which is usually based on some approximate Riemann solver. The numerical flux function is assumed to be consistent in the sense that

$$\hat{g}(\mathbf{u}, \mathbf{u}, w) = \mathbf{g}(\mathbf{u}, w), \quad \forall \mathbf{u} \in \mathbb{R}^3, w \in \mathbb{R}.$$

Since the ALE versions of the numerical fluxes are not so well known, here we list the formulae used in the present work.

#### Rusanov Flux

The Rusanov flux is a variant of the Lax–Friedrich flux and is given by

$$\hat{g}(\mathbf{u}_l, \mathbf{u}_r, w) = \frac{1}{2}[\mathbf{g}(\mathbf{u}_l, w) + \mathbf{g}(\mathbf{u}_r, w)] - \frac{1}{2}\lambda_{lr}(\mathbf{u}_r - \mathbf{u}_l),$$

where  $\lambda_{lr} = \lambda(\mathbf{u}_l, \mathbf{u}_r, w)$ ,

$$\lambda(\mathbf{u}_l, \mathbf{u}_r, w) = \max\{|v_l - w| + c_l, |v_r - w| + c_r\},$$

which is an estimate of the largest wave speed in the Riemann problem. Since the mesh velocity is close to the fluid velocity, the value of  $\lambda$  is close to the local sound speed. Thus, the numerical dissipation is independent of the velocity scale.

#### Roe Flux

The Roe scheme [39] is based on a local linearization of the conservation law and then exactly solving the Riemann problem for the linear approximation. The flux can be written as

$$\hat{g}(\mathbf{u}_l, \mathbf{u}_r, w) = \frac{1}{2}[\mathbf{g}(\mathbf{u}_l, w) + \mathbf{g}(\mathbf{u}_r, w)] - \frac{1}{2}|A_w|(\mathbf{u}_r - \mathbf{u}_l),$$

where the Roe average matrix  $A_w = A_w(\mathbf{u}_l, \mathbf{u}_r)$  satisfies

$$\mathbf{g}(\mathbf{u}_r, w) - \mathbf{g}(\mathbf{u}_l, w) = A_w(\mathbf{u}_r - \mathbf{u}_l),$$

and we define  $|A_w| = R|A - wI|R^{-1}$ . This matrix is evaluated at the Roe average state  $\mathbf{u}(\bar{q})$ ,  $\bar{q} = \frac{1}{2}(\mathbf{q}_l + \mathbf{q}_r)$ , where  $\mathbf{q} = \sqrt{\rho}[1, v, H]^T$  is the parameter vector introduced by Roe.

### HLLC Flux

This is based on a three wave approximate Riemann solver and the particular ALE version we use can also be found in [15]. Define the relative velocity  $q = v - w$ ; then, the numerical flux is given by

$$\hat{\mathbf{g}}(\mathbf{u}_l, \mathbf{u}_r, w) = \begin{cases} \mathbf{g}(\mathbf{u}_l, w), & S_l > 0, \\ \mathbf{g}^*(\mathbf{u}_l^*, w), & S_l \leq 0 < S_M, \\ \mathbf{g}^*(\mathbf{u}_r^*, w), & S_M \leq 0 \leq S_r, \\ \mathbf{g}(\mathbf{u}_r, w), & S_r < 0, \end{cases}$$

where the intermediate states are given by

$$\mathbf{u}_\alpha^* = \frac{1}{S_\alpha - S_M} \begin{bmatrix} (S_\alpha - q_\alpha)\rho_\alpha \\ (S_\alpha - q_\alpha)(\rho v)_\alpha + p^* - p_\alpha \\ (S_\alpha - q_\alpha)E_\alpha - p_\alpha q_\alpha + p^* S_M \end{bmatrix}, \quad \alpha = l, r,$$

and

$$\mathbf{g}^*(\mathbf{u}, w) = S_M \mathbf{u} + \begin{bmatrix} 0 \\ p^* \\ (S_M + w)p^* \end{bmatrix},$$

where

$$p^* = \rho_l(q_l - S_l)(q_l - S_M) + p_l = \rho_r(q_r - S_r)(q_r - S_M) + p_r,$$

which gives  $S_M$  as

$$S_M = \frac{\rho_r q_r (S_r - q_r) - \rho_l q_l (S_l - q_l) + p_l - p_r}{\rho_r (S_r - q_r) - \rho_l (S_l - q_l)}.$$

The signal velocities are defined as

$$S_l = \min\{q_l - c_l, \hat{v} - w - \hat{c}\}, \quad S_r = \max\{q_r + c_r, \hat{v} - w + \hat{c}\},$$

where  $\hat{v}, \hat{c}$  are Roe’s average velocity and speed of sound.

### Appendix B: Continuous Expansion Runge–Kutta (CERK) Schemes

We use a Runge–Kutta scheme to compute the predicted solution used to compute all the integrals in the DG scheme. In this section, we list down the CERK scheme for the following ODE:

$$\frac{du}{dt} = f(u, t).$$

Given the solution  $u^n$  at time  $t_n$ , the CERK scheme gives a polynomial solution in the time interval  $[t_n, t_{n+1})$  of the form

$$u(t_n + \theta h) = u^n + h \sum_{s=1}^{n_s} b_s(\theta)k_s, \quad \theta \in [0, 1],$$

where  $n_s$  is the number of stages and  $h$  denotes the time step.

### Second Order (CERK2)

The number of stages is  $n_s = 2$  and

$$b_1(\theta) = \theta - \theta^2/2, \quad b_2(\theta) = \theta^2/2,$$

and

$$k_1 = f(u^n, t_n),$$

$$k_2 = f(u^n + hk_1, t_n + h).$$

### Third Order (CERK3)

The number of stages is  $n_s = 4$  and

$$k_1 = f(u^n, t_n),$$

$$k_2 = f(u^n + (12/23)hk_1, t_n + 12h/23),$$

$$k_3 = f(u^n + h((-68/375)k_1 + (368/375)k_2), t_n + 4h/5),$$

$$k_4 = f(u^n + h((31/144)k_1 + (529/1152)k_2 + (125/384)k_3), t_n + h),$$

and

$$b_1(\theta) = (41/72)\theta^3 - (65/48)\theta^2 + \theta,$$

$$b_2(\theta) = -(529/576)\theta^3 + (529/384)\theta^2,$$

$$b_3(\theta) = -(125/192)\theta^3 + (125/128)\theta^2,$$

$$b_4(\theta) = \theta^3 - \theta^2.$$

### References

1. Springel, V.: E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. *Mon. Notices R. Astron. Soc.* **401**(2), 791–851 (2010)
2. Johnsen, E., Larsson, J., Bhagatwala, A.V., Cabot, W.H., Moin, P., Olson, B.J., Rawat, P.S., Shankar, S.K., Sjögreen, B., Yee, H.C., Zhong, X., Lele, S.K.: Assessment of high-resolution methods for numerical simulations of compressible turbulence with shock waves. *J. Comput. Phys.* **229**(4), 1213–1237 (2010)

3. Munz, C.D.: On Godunov-type schemes for Lagrangian gas dynamics. *SIAM J. Num. Anal.* **31**(1), 17–42 (1994)
4. Carré, G., Del Pino, S., Després, B., Labourasse, E.: A cell-centered Lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. *J. Comput. Phys.* **228**(14), 5160–5183 (2009)
5. Maire, P.-H.: A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. *J. Comput. Phys.* **228**(7), 2391–2425 (2009)
6. Hirt, C., Amsden, A., Cook, J.: An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *J. Comput. Phys.* **14**(3), 227–253 (1974)
7. Donea, J., Huerta, A., Ponthot, J.-P., Rodríguez-Ferran, A.: *Arbitrary Lagrangian–Eulerian Methods*. Wiley, Oxford (2004)
8. Walter, B.: High order accurate direct arbitrary-Lagrangian–Eulerian ADER-MOOD finite volume schemes for non-conservative hyperbolic systems with stiff source terms. *Commun. Comput. Phys.* **21**(01), 271–312 (2017)
9. Gaburro, E., Castro, M.J., Dumbser, M.: Well-balanced arbitrary-Lagrangian–Eulerian finite volume schemes on moving nonconforming meshes for the Euler equations of gas dynamics with gravity. *Mon. Notices R. Astron. Soc.* **477**(2), 2251–2275 (2018). <https://doi.org/10.1093/mnras/sty542>. 21 June
10. Lomtev, I., Kirby, R., Karniadakis, G.: A discontinuous Galerkin ALE method for compressible viscous flows in moving domains. *J. Comput. Phys.* **155**(1), 128–159 (1999)
11. Venkatasubban, C.S.: A new finite element formulation for ALE (arbitrary Lagrangian Eulerian) compressible fluid mechanics. *Int. J. Eng. Sci.* **33**(12), 1743–1762 (1995)
12. Wang, L., Persson, P.-O.: High-order discontinuous Galerkin simulations on moving domains using ALE formulations and local remeshing and projections. *American Institute of Aeronautics and Astronautics* (2015)
13. Weizhang, H., Russell, R.D.: *Adaptive Moving Mesh Methods*. Applied Mathematical Sciences. Springer, Berlin (2011)
14. He, P., Tang, H.: An adaptive moving mesh method for two-dimensional relativistic hydrodynamics. *Commun. Comput. Phys.* **11**(1), 114–146 (2012)
15. Luo, H., Baum, J.D., Löhner, R.: On the computation of multi-material flows using ALE formulation. *J. Comput. Phys.* **194**(1), 304–328 (2004)
16. Boscheri, W., Dumbser, M., Balsara, D.S.: High-order ADER-WENO ALE schemes on unstructured triangular meshes—application of several node solvers to hydrodynamics and magnetohydrodynamics. *Int. J. Numer. Meth. Fluids* **76**, 737–778 (2014). <https://doi.org/10.1002/flid.3947>
17. Luo, H., Baum, J.D., Löhner, R.: On the computation of multi-material flows using ALE formulation. *J. Comput. Phys.* **194**, 304–328 (2004). <https://doi.org/10.1016/j.jcp.2003.09.026>
18. Liu, W., Cheng, J., Shu, C.-W.: High order conservative Lagrangian schemes with Lax–Wendroff type time discretization for the compressible Euler equations. *J. Comput. Phys.* **228**(23), 8872–8891 (2009)
19. Boscheri, W., Dumbser, M.: Arbitrary-Lagrangian–Eulerian one-step WENO finite volume schemes on unstructured triangular meshes. *Commun. Comput. Phys.* **14**(5), 1174–1206 (2013)
20. Boscheri, W., Dumbser, M.: High order accurate direct arbitrary-Lagrangian–Eulerian ADER-WENO finite volume schemes on moving curvilinear unstructured meshes. *Comput. Fluids* **136**, 48–66 (2016)
21. Boscheri, W., Dumbser, M., Balsara, D.S.: High-order ADER-WENO ALE schemes on unstructured triangular meshes—application of several node solvers to hydrodynamics and magnetohydrodynamics. *Int. J. Num. Methods Fluids* **76**, 737–778 (2014)
22. Klingenberg, C., Schnücke, G., Xia, Y.: Arbitrary Lagrangian–Eulerian discontinuous Galerkin method for conservation laws: analysis and application in one dimension. *Math. Comput.* **86**, 1203–1232 (2017)
23. Harten, A., Hyman, J.M.: Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comput. Phys.* **50**, 253–269 (1983)
24. Gassner, G., Dumbser, M., Hindenlang, F., Munz, C.-D.: Explicit one-step time discretizations for discontinuous Galerkin and finite volume schemes based on local predictors. *J. Comput. Phys.* **230**(11), 4232–4247 (2011)
25. Owren, B., Zennaro, M.: Derivation of efficient, continuous, explicit Runge–Kutta methods. *SIAM J. Sci. Stat. Comput.* **13**(6), 1488–1501 (1992)
26. 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)
27. Cockburn, B., Shu, C.-W.: TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. *Math. Comput.* **52**(186), 411–435 (1989)

28. Zhang, X., Shu, C.-W.: On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. *J. Comput. Phys.* **229**(23), 8918–8934 (2010)
29. Zhong, X., Shu, C.-W.: A simple weighted essentially nonoscillatory limiter for Runge–Kutta discontinuous Galerkin methods. *J. Comput. Phys.* **232**(1), 397–415 (2013)
30. Sod, G.: A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.* **27**, 1–31 (1978)
31. Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.* **77**(2), 439–471 (1988)
32. Titarev, V.A., Toro, E.F.: Finite volume WENO schemes for three-dimensional conservation laws. *J. Comput. Phys.* **201**, 238–260 (2014)
33. Toro, E.F.: *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer, Berlin (1999)
34. Woodward, P., Colella, P.: The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.* **54**(1), 115–173 (1984)
35. Zhu, J., Qiu, J.: A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws. *J. Comput. Phys.* **318**, 110–121 (2016)
36. Loubere, R., Shashkov, M.J.: A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian–Eulerian methods. *J. Comput. Phys.* **209**(1), 105–138 (2005)
37. Olivier, G., Alauzet, F.: A new changing-topology ALE scheme for moving mesh unsteady simulations. In: 49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, p. 474 (2011)
38. Frederic, Alauzet: A changing-topology moving mesh technique for large displacements. *Eng. Comput.* **30**(2), 175–200 (2014)
39. Roe, P.L.: Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.* **43**(2), 357–372 (1981)