INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2015; **77**:249–273
Published online 14 October 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/fld.3965

RESEARCH PAPER PRESENTED AT MULTIMAT 2013
MULTI-MATERIAL HYDRODYNAMICS SIMULATIONS

Monotonicity in high-order curvilinear finite element arbitrary Lagrangian–Eulerian remap

R. W. Anderson[1], V. A. Dobrev[1], Tz. V. Kolev[1] and R. N. Rieben[2,*,†]

[1]*Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA, USA*
[2]*Weapons and Complex Integration, Lawrence Livermore National Laboratory, Livermore, CA, USA*

SUMMARY

The remap phase in arbitrary Lagrangian–Eulerian (ALE) hydrodynamics involves the transfer of field quantities defined on a post-Lagrangian mesh to some new mesh, usually generated by a mesh optimization algorithm. This problem is often posed in terms of transporting (or advecting) some state variable from the old mesh to the new mesh over a fictitious time interval. It is imperative that this remap process be monotonic, that is, not generate any new extrema in the field variables. It is well known that the only linear methods that are guaranteed to be monotonic for such problems are first-order accurate; however, much work has been performed in developing non-linear methods, which blend both high and low (first) order solutions to achieve monotonicity and preserve high-order accuracy when the field is sufficiently smooth. In this paper, we present a set of methods for enforcing monotonicity targeting high-order discontinuous Galerkin methods for advection equations in the context of high-order curvilinear ALE hydrodynamics. Published 2014. This article is a U.S. Government work and is in the public domain in the USA.

Received 11 March 2014; Revised 15 August 2014; Accepted 15 September 2014

KEY WORDS: shock hydrodynamics; multi-material hydrodynamics; monotonicity; ALE methods; finite element methods; high-order methods

1. INTRODUCTION AND MOTIVATION

We are interested in arbitrary Lagrangian–Eulerian (ALE) numerical methods for solving the Euler equations of compressible hydrodynamics for modeling complex, multi-material, high speed flow and shock wave propagation over general 2D and 3D computational domains. The remap phase in ALE methods involves the transfer of state variables defined on a post-Lagrangian mesh to some new mesh, usually generated by a mesh optimization algorithm. This problem can be formulated in terms of transporting (or advecting) the state variables from the old mesh to the new mesh over a fictitious (or pseudo) time interval determined by the motion of the mesh from its original post-Lagrangian position to some new position. It is imperative that this remap process be monotonic, that is, not generate any new extrema in the state variables. In the context of multi-material hydrodynamics, this is most important at contact discontinuities (material interfaces). It is well known that the only linear methods that are guaranteed to be monotonic for such advection problems are first-order accurate [1]; however, much work has been performed in developing non-linear methods,

*Correspondence to: R. N. Rieben, Weapons and Complex Integration, Lawrence Livermore National Laboratory, 7000 East Ave L-095, Livermore, CA 94550, USA.
†E-mail: rieben1@llnl.gov

which blend both high and low (first) order solutions to achieve monotonicity and preserve high-order accuracy when the field is sufficiently smooth [2, 3]. Here, we propose a set of non-linear methods for enforcing monotonicity, which target high-order discontinuous Galerkin (DG) methods for advection equations in the context of high-order curvilinear ALE hydrodynamics. In this paper, we only consider the case of single material flow, but the approaches we describe here can also be applied to multi-material flow, as we discuss in Section 5.

Using the general high-order finite element approach described in [4], we consider an ALE formulation based on three phases:

- *Lagrangian phase*, solving the Euler equations on a moving curvilinear mesh;
- *Mesh optimization phase*, using, for example, harmonic or inverse-harmonic smoothing;
- *Remap phase*, based on conservative and monotonic DG advection remap.

Semi-discrete DG methods for advection equations can be formulated in terms of high-order finite element 'mass' and 'advection' matrices. These formulations result in high-order accuracy for sufficiently smooth fields but produce non-monotonic results (spurious oscillations) for discontinuous fields. In this paper, we present three non-linear approaches for modifying either the mass and advection matrices or the solution itself to enforce monotonicity:

- *Locally scaled diffusion* (*LSD*): In this approach, we locally lump the mass matrices and upwind the advection matrices using an iterative process and a local 'monotonicity measure'.
- *Generalized high-order flux-corrected transport* (*FCT*): This is a high-order generalization of the FCT methods in [3], which is modified to account for non-lumped mass matrices.
- *Optimization-based remap* (*OBR*): In this approach, we apply an optimization procedure to enforce monotonicity in the remapped fields, based on the work of Bochev *et al.* [5].

The paper is organized as follows. In Section 2, we define a set of high-order semi-discrete DG methods for remapping the state variables of ALE hydrodynamics (density, internal energy and velocity). In Section 3, we define monotonicity in the context of these semi-discrete equations and develop a linear, first-order variation of the general equations using mass lumping and discrete 'upwinding' that is guaranteed to be monotonic. We then describe three non-linear approaches for generating monotone solutions, which are designed to achieve high-order accuracy whenever the solution is sufficiently smooth and automatically transition to first-order monotonic methods in regions where the solution is not sufficiently smooth. In Section 4, we present a sequence of numerical results comparing each of the methods to verify and quantify their performance on a series of benchmark test problems. Finally, in Section 5, we summarize our results and draw some conclusions.

2. ADVECTION-BASED REMAP USING A SEMI-DISCRETE DG FORMULATION

This section provides a very brief overview of the high-order DG advection-based ALE remap scheme to which we apply the monotonicity algorithms that are of central interest to the current paper. A more detailed description of the derivation and the properties of our DG advection approach will be presented in a future publication.

The goal of ALE remap is to transfer a field, for example, the mass density ρ, defined on an initial spatial domain, $\widetilde{\Omega} \subset \mathbb{R}^d$, to a new domain, $\Omega \subset \mathbb{R}^d$, where d denotes the spatial dimension. We assume that every point (particle) $\widetilde{x} \in \widetilde{\Omega}$ is associated with a unique point (particle) $x \in \Omega$, and we are given a continuous transition function $F(\widetilde{x}, \tau) : \widetilde{\Omega} \times [0, 1] \mapsto \mathbb{R}^d$, such that

$$F(\widetilde{x}, 0) = \widetilde{x} \quad \text{and} \quad F(\widetilde{x}, 1) = x, \quad \forall \widetilde{x} \in \widetilde{\Omega}.$$

We refer to the parameter $\tau \in [0, 1]$ as 'pseudo-time'.

We think of $\Omega(\tau) = \left\{ F(\widetilde{x}, \tau) : \widetilde{x} \in \widetilde{\Omega} \right\}$ as an intermediate domain in pseudo-time, which 'interpolates' between the initial and new configurations, because $\Omega(0) = \widetilde{\Omega}$ and $\Omega(1) = \Omega$. One simple way to define these intermediate domains is to use linear interpolation in pseudo-time

$$F(\widetilde{x}, \tau) = \widetilde{x} + \tau(x - \widetilde{x}). \tag{1}$$

DOI: 10.1002/fld

In typical ALE hydrodynamics calculations, we are only given an initial and final mesh to consider, and so linear interpolation is the best we can do. For small displacements, this choice is a good option; however, for large displacements (specifically, large rotations), it is possible to obtain inverted elements for $\tau \in (0,1)$ even when both the initial and final elements are valid. One possible solution for that case could be to use a mapping F that is a higher-order polynomial in pseudo-time, for example, constructed by interpolating in pseudo-time a sequence of valid meshes obtained during the mesh relaxation process.

Based on the transition function F, we define the mesh pseudo-velocity

$$u \equiv \frac{\partial F}{\partial \tau}. \tag{2}$$

In the simple linear transition case (1), this formula reduces to

$$u(\widetilde{x}) = x - \widetilde{x},$$

which is simply the mesh displacement field commonly used in advection-based remap methods (e.g., [6]).

Similar to the case of Lagrangian flow [4], we can track a 'particle' $\widetilde{x} \in \widetilde{\Omega}$ in pseudo-time through $x(\tau) = F(\widetilde{x}, \tau)$ and introduce the concept of a 'pseudo-material' derivative along these particle trajectories

$$\frac{\mathrm{d}\rho}{\mathrm{d}\tau}(x(\tau),\tau) = \frac{\mathrm{d}}{\mathrm{d}\tau}[\rho(F(\widetilde{x},\tau),\tau)] = \frac{\partial \rho}{\partial \tau} + u \cdot \nabla \rho.^{1}$$

Because the goal of ALE remap is to extend $\rho(\cdot,0)$ to $\rho(\cdot,1)$ without changing it with respect to an Eulerian frame, that is, such that $\frac{\partial \rho}{\partial \tau} = 0$,[1] one commonly used approach for remap is to define ρ by solving the pseudo-time advection equation

$$\frac{\mathrm{d}\rho}{\mathrm{d}\tau} = u \cdot \nabla \rho. \tag{3}$$

2.1. A semi-discrete DG formulation of the advection equation

Motivated by conservation of mass (the integral of ρ), we derive a weak formulation of the advection (3) by evaluating the change of the integral of ρ multiplied with a function ψ, which follows the pseudo-time deformation (i.e., its 'pseudo-material' derivative is zero) over an arbitrary subregion $U(\tau) \subset \Omega(\tau)$. Using the Reynolds transport theorem (see, e.g., (2.4) in [4]), the fact that $\frac{\mathrm{d}\psi}{\mathrm{d}\tau} = 0$, applying (3), and integrating by parts, we obtain

$$\begin{aligned} \frac{d}{d\tau}\int_{U(\tau)} \rho\psi &= \int_{U(\tau)} \frac{\mathrm{d}}{\mathrm{d}\tau}\,\psi + \rho\frac{\mathrm{d}\psi}{\mathrm{d}\tau} + \rho\psi\nabla\cdot u = \int_{U(\tau)} (u\cdot\nabla\rho)\psi + \rho\psi\nabla\cdot u \\ &= \int_{U(\tau)} \nabla\cdot(\rho u)\psi = -\int_{U(\tau)} \rho u\cdot\nabla\psi + \int_{\partial U(\tau)} \rho u\cdot n\psi. \end{aligned} \tag{4}$$

A semi-discrete discretization of (4) is based on a computational mesh $\mathcal{T}(\tau)$ on each deformed domain $\Omega(\tau)$, and a finite element space for ρ and ψ defined on that mesh. In this paper, our main focus is on the case of discontinuous ρ, which is the natural choice for thermodynamic quantities in many ALE codes. Specifically, we approximate the density function using the expansion

$$\boldsymbol{\rho}(x,\tau) = \sum_i \boldsymbol{\rho}_i(\tau)\boldsymbol{\psi}_i(x,\tau) = \boldsymbol{\rho}(\tau)^T\boldsymbol{\psi}(x,\tau),^{2} \tag{5}$$

where $\{\psi_i\}_{i=1}^{N_v}$ is a (moving) basis of a finite element space $\mathcal{V}(\tau) \subset L_2(\Omega)$ defined on $\mathcal{T}(\tau)$, $\rho(\tau)$ is an unknown pseudo-time-dependent vector of size $N_{\mathcal{V}} = \dim\mathcal{V}(\tau)$ and ψ is a column vector of all the basis

[1][Correction added on 27 November 2014, after first online publication: dρ/dτ was changed to ∂ρ/∂τ]
[2][Correction added on 27 November 2014, after first online publication: ρ and ψ were changed to bold in equation (5)]

DOI: 10.1002/fld

functions.[3] The finite element functions in $\mathcal{V}(\tau)$ are discontinuous across the interior faces $f \in \mathcal{F}_i(\tau)$ of the mesh,[3] and therefore, to fix notation, we introduce the jump $[\![\cdot]\!]$ and average $\{\cdot\}$ operators:

$$[\![\phi]\!] = \phi^- - \phi^+, \quad \{\phi\} = \frac{1}{2}(\phi^- + \phi^+), \quad \text{where} \quad \phi^{\pm}(x) = \lim_{s \to 0+} \phi\big(x \pm s n_f(x)\big), \quad x \in f,$$

and n_f is a normal vector field on each interior face f of the mesh, which varies continuously on f. While the expansion (5) and the scheme we develop later in the text are valid for all choices of $\mathcal{T}(\tau)$ and $\mathcal{V}(\tau)$,[3] our interest is in the case of high-order finite elements on high-order (curvilinear) meshes, which arise in the high-order treatment of the Lagrangian phase of ALE [4].

Applying (4) on each element T of the mesh $\mathcal{T}(\tau)$, summing up the results and modifying the flux in the integrals associated with internal faces, we obtain

$$\frac{d}{d\tau}\int_{\Omega(\tau)} \rho\psi = -\sum_{T\in\mathcal{T}(\tau)} \int_T \rho u \cdot \nabla\psi + \sum_{f\in\mathcal{F}_i(\tau)} \int_f \{\rho u \cdot n_f\}_* [\![\psi]\!] \qquad \forall \psi \in \mathcal{V}(\tau)\,. \tag{6}$$

The numerical flux $\{\rho u \cdot n_f\}_*$ accounts for the fact that ρ is discontinuous on the internal mesh faces and its particular definition may affect the quality of the scheme substantially. In our computations, we use the Godunov (upwind) flux

$$\{\rho u \cdot n_f\}_* = \rho_u(u \cdot n_f) = (u \cdot n_f)\{\rho\} - \frac{1}{2}|u \cdot n_f|[\![\rho]\!], \tag{7}$$

which simply defines ρ_u as one of the traces of ρ at each point $x \in f$, based on the sign of $(u \cdot n_f)(x)$. Note that the right-hand side of (6) vanishes for $\psi \equiv 1$, which implies mass conservation.

The scheme (6) can be recast in the matrix–vector form

$$\frac{d}{d\tau}(\mathrm{M}\rho) = \mathrm{A}\rho, \tag{8}$$

where $\mathbf{M}$ and $\mathbf{A}$ are the mass and 'advection' matrices

$$\mathbf{M}(\tau) = \int_{\Omega(\tau)} \boldsymbol{\psi}\boldsymbol{\psi}^{\mathrm{T}}, \qquad \mathbf{A}(\tau) = -\sum_{T\in\mathcal{T}(\tau)} \int_T (u \cdot \nabla\boldsymbol{\psi})\boldsymbol{\psi}^{\mathrm{T}} + \sum_{f\in\mathcal{F}_i(\tau)} \int_f (u \cdot n_f)[\![\boldsymbol{\psi}]\!]\boldsymbol{\psi}_u^{\mathrm{T}}. \tag{9}$$

In these settings, mass conservation follows from the fact that $\mathbf{A}$ has zero column sums.

We can equivalently rewrite (8) as a system of ordinary differential equations for the vector of finite element degrees of freedom (DOF),

$$\mathrm{M}\frac{d\rho}{d\tau} = \mathrm{K}\rho, \tag{10}$$

by setting $\mathbf{K} = \mathbf{A} - \frac{d\mathbf{M}}{d\tau}$. Using arguments similar to (4), one can derive a formula for the pseudo-time derivative of the mass matrix, from which we obtain the following expression for the advection matrix $\mathbf{K}$:

$$\mathbf{K}(\tau) = \sum_{T\in\mathcal{T}(\tau)} \int_T \boldsymbol{\psi}(u \cdot \nabla\boldsymbol{\psi}^{\mathrm{T}}) - \sum_{f\in\mathcal{F}_i(\tau)} \int_f (u \cdot n_f)\boldsymbol{\psi}_d[\![\boldsymbol{\psi}^{\mathrm{T}}]\!], \tag{11}$$

where $(u \cdot n_f)\psi_d$ denotes the downwind flux:

$$(u \cdot n_f)\psi_d = (u \cdot n_f)\{\psi\} + \frac{1}{2}|(u \cdot n_f)|[\![\psi]\!]$$

Note that $\mathbf{K}$ has zero row sums, which implies that (10) will preserve constant functions. More generally, one can show that our scheme will preserve any initial ρ that is representable

[3] [Correction added on 27 November 2014, after first online publication: V was changed to $\mathcal{V}$]

DOI: 10.1002/fld

in all intermediate finite element spaces $\mathcal{V}(\tau)$.[4] For example, a linear field in physical coordinates belongs to the Q_2 space on any Q_2 quadrilateral mesh and therefore will be preserved.

In this paper, we will study the convergence properties of the method described here only empirically (in Section 4) and in the context of the monotonicity schemes introduced later, which are the focus of this article. As a general remark, we observe a convergence rate of $O(h^{k+1})$ for the method using k-th order DG spaces P_k/Q_k. Naturally, this rate is observed only for 'smooth' problems where not only the quantity being remapped is 'smooth' but also the transition map F (and, therefore, the velocity u) as well.

As mentioned earlier, (8) and (10) are equivalent; however, this is no longer true after discretization in pseudo-time. It is straightforward to verify that applying a Runge–Kutta (RK) method to (8) leads to a conservative scheme (because $\mathbf{A}^{\mathrm{T}}\mathbf{1}=\mathbf{0}$), which, in general, does not preserve constant fields (i.e., oscillations are introduced); on the other hand, discretizing (10) leads to a constant preserving scheme (because $\mathbf{K1}=\mathbf{0}$), which is not conservative. Therefore, special time discretization schemes need to be applied in order to ensure that both properties are valid on fully discrete level; for example, in [7], the authors consider the case of continuous piecewise linear finite elements. In this article, we will focus on discretizations of (10), which are only conservative up to the pseudo-time discretization error. Ultimately, the monotonicity approaches presented here should be applicable to a more general class of time-integration schemes, including fully conservative ones.

2.2. *Example: Bernstein basis and DG advection scheme in one dimension*

The discretization method described in the previous section is independent of the choice of basis for the discrete spaces $\mathcal{V}(\tau)$.[4] However, in the following sections, this choice becomes important because of the algebraic nature of our approach: working with the discrete vectors and matrices directly. Our choice is to use the Bernstein basis, which in one spatial dimension for degree p is defined as

$$\psi_k(x) = \binom{p}{k} x^k (1-x)^{p-k}, \qquad k = 0, \ldots, p, \qquad x \in [0,1].$$

This definition is readily extendable to multiple dimensions in the case of quadrilaterals and hexahedra via tensor products, and it can also be extended to other types of elements. Two simple but important properties of this basis are that it is non-negative, $\psi_k(x) \geqslant 0$, and a partition of unity, $\sum_k \psi_k(x) \equiv 1$. As an example in 1D, consider an element of size h and discrete space of degree $p=2$: the mass matrix $\mathbf{M}_i$ and the advection matrix $\mathbf{K}_i$ (assuming constant mesh velocity $u>0$) for element i are given by

$$\mathbf{M}_i = \frac{h}{30}\begin{pmatrix} 6 & 3 & 1 \\ 3 & 4 & 3 \\ 1 & 3 & 6 \end{pmatrix} \quad \mathbf{K}_i = \frac{u}{6}\begin{pmatrix} -3 & 2 & 1 & 0 \\ -2 & 0 & 2 & 0 \\ -1 & -2 & -3 & 6 \end{pmatrix}.$$

The last column in $\mathbf{K}_i$ corresponds to the first degree of freedom in the next element to the right. The evolution (10) for element i then becomes

$$\frac{d}{d\tau}\begin{pmatrix} \rho_i^{(0)} \\ \rho_i^{(1)} \\ \rho_i^{(2)} \end{pmatrix} = \frac{u}{h}\begin{pmatrix} -2 & 2 & -3 & 3 \\ -1 & 0 & 10 & -9 \\ 0 & -2 & -7 & 9 \end{pmatrix}\begin{pmatrix} \rho_i^{(0)} \\ \rho_i^{(1)} \\ \rho_i^{(2)} \\ \rho_{i+1}^{(0)} \end{pmatrix}.$$

[4][Correction added on 27 November 2014, after first online publication: V was changed to $\mathcal{V}$]

DOI: 10.1002/fld

2.3. Extension to multi-field remap for ALE hydrodynamics

In many applications, we need to simultaneously remap more than one physical field. For example, in ALE hydrodynamics, we need to remap the material density ρ, the velocity v and the internal energy e. The semi-discrete DG advection approach from the previous section naturally extends to these settings: we introduce the scalar finite element spaces $\mathcal{V}_\rho(\tau)$, $\mathcal{V}_v(\tau)$ and $\mathcal{V}_e(\tau)$ with moving bases $\{\psi_i\}$, $\{w_j\}$ and $\{\phi_k\}$, respectively, and discretize

$$\rho(x,\tau) = \sum_i \rho_i(\tau)\psi_i(x), \qquad v^{x,y,z}(x,\tau) = \sum_j v_j^{x,y,z}(\tau) w_j(x), \qquad e(x,\tau) = \sum_k e_k(\tau)\phi_k(x).$$

As before, the spaces $\mathcal{V}_\rho(\tau)$ and $\mathcal{V}_e(\tau)$ are typically discontinuous (although our framework can also handle the continuous case),[3] but we assume that $\mathcal{V}_v(\tau)$ is continuous,[5] so each component of the velocity v^x, v^y and v^z will be remapped independently as a continuous field. For simplicity, we will skip the subscript and focus on the remap of one velocity component denoted with v. We also only consider single material flow, because the handling of mixed material zones during remap will be addressed in a future publication.

Without going into details, starting from the advection equations for the density, momentum and internal energy

$$\frac{d\rho}{d\tau} = u\cdot\nabla\rho, \qquad \frac{d(\rho v)}{d\tau} = u\cdot\nabla(\rho v), \qquad \frac{d(\rho e)}{d\tau} = u\cdot\nabla(\rho e),$$

we apply the previously described DG weak formulation approach with trial and test functions in $\mathcal{V}_\rho(\tau)$, $\mathcal{V}_v(\tau)$ and $\mathcal{V}_e(\tau)$ to derive the semi-discrete advection equations

$$\mathrm{M}_\rho \frac{d\rho}{d\tau} = \mathrm{K}_\rho \rho, \tag{12}$$

$$\mathrm{M}_\mathrm{v} \frac{dv}{d\tau} = \mathrm{K}_\mathrm{v} v, \tag{13}$$

$$\mathrm{Me} \frac{de}{d\tau} = \mathrm{K}_\mathrm{e} e. \tag{14}$$

where $\mathbf{M}_\rho$ and $\mathbf{K}_\rho$ are defined by (9) and (11), while the velocity and energy mass and advection matrices are given by

$$\mathbf{M}_\mathbf{v} = \int_\Omega \rho\, \boldsymbol{w}\boldsymbol{w}^\mathrm{T}, \qquad \mathbf{K}_\mathbf{v} = \sum_{T\in\mathcal{T}(\tau)} \int_T \rho\, \boldsymbol{w}(u\cdot\nabla \boldsymbol{w}^\mathrm{T}),$$
$$\mathbf{M}_\mathbf{e} = \int_\Omega \rho\, \boldsymbol{\phi}\boldsymbol{\phi}^\mathrm{T}, \qquad \mathbf{K}_\mathbf{e} = \sum_{T\in\mathcal{T}(\tau)} \int_T \rho\, \boldsymbol{\phi}(u\cdot\nabla \boldsymbol{\phi}^\mathrm{T}) - \sum_{f\in\mathcal{F}_i(\tau)} \int_f \rho_u (u\cdot n_f)\phi_d [\![\boldsymbol{\phi}^\mathrm{T}]\!].$$

Note that the previous matrices are weighted with the density and that $\mathbf{K}_\mathbf{v}$ and $\mathbf{K}_\mathbf{e}$ have zero row sums, so constant velocities and energies will be preserved by (13) and (14)

In the following section, we will discuss methods for ensuring monotonicity applicable to any of the equations (12)–(14). Applying these methods individually to the equations, however, will generally lead to loss of conservation in the velocity (13) and energy (14) schemes because of their dependence on the density remap (12). To address this issue, special 'synchronized' versions of the monotonicity schemes have to be developed, which are outside the scope of this article; see, for example, [8].

[5][Correction added on 27 November 2014, after first online publication: V was changed to $\mathcal{V}$]

DOI: 10.1002/fld

3. NON-LINEAR APPROACHES FOR HIGH-ORDER MONOTONICITY

Besides conservation and accuracy (e.g., preservation of constants), another critical requirement on the remap process is that it does not create artificial oscillations or lead to unphysical undershoots or overshoots. This requirement is formalized by different concepts of 'monotonicity', and in this paper, we follow the work of Kuzmin and Turek [9] and Kuzmin *et al.* [3] to extend these concepts to our high-order DG advection algorithm.

Consider the semi-discrete advection (10) for the density

$$\frac{d\rho}{d\tau} = \mathrm{M}^{-1}\mathrm{K}\rho.$$

Because $\mathbf{K}$ has zero row sums, we can write the individual entries in the following flux form:

$$\frac{d\rho_i}{d\tau} = \sum_{j\neq i} \left(\mathrm{M}^{-1}\mathrm{K}\right)_{ij}\left(\rho_j - \rho_i\right). \tag{15}$$

Let us consider the following assumption for the off-diagonal entries of $\mathbf{M}^{-1}\mathbf{K}$:

$$\left(\mathbf{M}^{-1}\mathbf{K}\right)_{ij} \geqslant 0, \qquad i \neq j, \tag{16}$$

which holds in the piecewise-constant (Q_0) case. For higher-order spaces, this property can be ensured only after suitable modifications to $\mathbf{M}$ and $\mathbf{K}$, for example, using mass lumping and 'discrete upwinding', as discussed in the subsections later. Assuming that (16) holds, we can conclude that

$$\text{"maxima do not increase"} : \rho_i = \max_j \rho_j \;\Rightarrow\; \rho_j - \rho_i \leqslant 0 \;\Rightarrow\; \frac{d\rho_i}{d\tau} \leqslant 0,$$

and

$$\text{"minima do not decrease"} : \rho_i = \min_j \rho_j \;\Rightarrow\; \rho_j - \rho_i \geqslant 0 \;\Rightarrow\; \frac{d\rho_i}{d\tau} \geqslant 0.$$

Because the discrete advection operator $\mathbf{M}^{-1}\mathbf{K}$ is sparse, the fact that *local* maxima/minima cannot increase/decrease implies that ρ will stay between given initial bounds (e.g., ρ will remain positive). This is a form of monotonicity known as the local extremum diminishing (LED) property. The work of Kuzmin and Turek [9] and Kuzmin *et al.* [3] show that LED is the multi-dimensional, unstructured generalization of the total variation diminishing property [10].

Much of the work in developing monotone/LED advection methods has focused on low-order finite elements, although there are some exceptions [11]. The extension of the notion of monotonicity to high-order finite element functions poses a number of challenges. First, the mass matrix cannot be lumped without damaging the high-order approximation, and its inverse in general contains positive and negative entries. This means that considering the signs of the entries of $\mathbf{K}$ is not enough in the high-order case. Second, high-order functions easily oscillate by nature, so suppressing the creation of local extrema is even more challenging. Finally, the high-order polynomial basis can be critical, because having a positive vector of DOF ρ does not necessarily translate to having a positive function ρ. In our experiments in Section 4, we use the Bernstein polynomial basis, which has positive basis functions, guaranteeing that ρ will be positive if ρ is.

For discretization in pseudo-time of the semi-discrete equations (15), we will consider explicit multistage RK methods, among which the class of strong stability preserving (SSP) methods [12] is of particular interest because they define high-order monotone schemes by combining multiple monotone forward Euler steps. Generally, we will focus on a single time step evolving $\rho = \rho(\tau)$ to $\rho^{new} \approx \rho(\tau + \Delta\tau)$. Let us consider the forward Euler method:

$$\rho_i^{new} = \rho_i + \Delta\tau \sum_{j\neq i} \left(\mathrm{M}^{-1}\mathrm{K}\right)_{ij}\left(\rho_j - \rho_i\right).$$

DOI: 10.1002/fld

This example highlights another motivation for the requirement that the off-diagonal entries in $\mathbf{M}^{-1}\mathbf{K}$ are non-negative: with this assumption, we can guarantee that there is a small enough time step in forward Euler, such that the every entry of ρ^{new} is a convex combination of the neighbor entries from the old vector ρ.

To formulate the LED property on a fully discrete level, we first introduce a local neighborhood N_i for each degree of freedom i. Given N_i, we impose the following monotonicity requirement at each i:

$$\rho_i^{min} \leqslant \rho_i^{new} \leqslant \rho_i^{max}, \qquad \text{where} \qquad \rho_i^{min} = \min_{j \in N_i} \rho_j \ \text{ and } \ \rho_i^{max} = \max_{j \in N_i} \rho_j. \tag{17}$$

Our default option is to define N_i based on the sparsity pattern of the i-th row of $\mathbf{K}$, motivated by the sum in (15) for a diagonal $\mathbf{M}$. Because of the face integrals in our DG scheme, this means that N_i consists of all DOF in elements that share a face with the element containing i. More restrictive choices, which, for example, select only upwind neighbors, are also possible.

In the remainder of this section, we present three algebraic approaches that take the original high-order DG solution and modify it, in order to ensure monotonicity. By using the Godunov theorem, the modification cannot lead to a linear scheme and maintain high-order approximation at the same time. Thus, all of the approaches use the high-order solution in different, non-linear, ways.

3.1. LSD

Our goal is to modify the high-order linear scheme (10) in order to remove oscillatory behavior while preserving the high-order in smooth regions. We utilize the following two-step approach: (i) we modify (10) to obtain a low-order monotone linear scheme; (ii) we define a blending algorithm that combines the high-order scheme with the low-order monotone scheme. The blending process will depend on the input data favoring the monotone scheme where oscillations arise and the high-order scheme in smooth regions.

One way to derive a monotone scheme from (10) is to apply 'discrete upwinding' to the matrix $\mathbf{K}$ and lump the mass matrix $\mathbf{M}$ (see, e.g., [9]):

$$\mathrm{M}^* \frac{d\rho}{d\tau} = \mathrm{K}^* \rho. \tag{18}$$

The diagonal lumped mass matrix $\mathbf{M}^*$ and 'discretely upwinded' advection matrix $\mathbf{K}^*$ can be written as

$$\mathrm{M}^* = \mathrm{M} + \mathrm{L}, \qquad \mathrm{K}^* = \mathrm{K} + \mathrm{D}, \tag{19}$$

where the matrices $\mathbf{L}$ and $\mathbf{D}$ are algebraically constructed symmetric matrices with off-diagonal entries

$$\mathbf{L}_{ij} = -\mathbf{M}_{ij}, \qquad \mathbf{D}_{ij} = \max\{0, -\mathbf{K}_{ij}, -\mathbf{K}_{ji}\}, \tag{20}$$

and diagonal entries, which zero out the row sums

$$\mathbf{L}_{ii} = -\sum_{j \neq i} \mathbf{L}_{ij}, \qquad \mathbf{D}_{ii} = -\sum_{j \neq i} \mathbf{D}_{ij}. \tag{21}$$

The matrices $\mathbf{L}$ and $-\mathbf{D}$ are algebraic diffusion matrices in that they have properties typical for discretizations of the diffusion operator, $(-\Delta)$: they are symmetric and sparse and have non-positive off-diagonal entries and zero row sums. The use of the positive Bernstein basis is essential here because it guarantees that $\mathbf{L}_{ij} = -\mathbf{M}_{ij} \leqslant 0$. The function of the matrix $\mathbf{D}$ is to render $\mathbf{K}^*$ LED by eliminating non-positive off-diagonal entries of the advection matrix $\mathbf{K}$ in a conservative manner. Indeed, mass conservation for (18) follows from the identities $\mathbf{L}^{\mathrm{T}}\mathbf{1} = \mathbf{D}^{\mathrm{T}}\mathbf{1} = \mathbf{0}$, which combined with $\frac{d\mathrm{M}}{d\tau} = \mathrm{A} - \mathrm{K}$, and $\mathbf{A}^{\mathrm{T}}\mathbf{1} = \mathbf{0}$ imply

$$\frac{d}{d\tau}\left(1^{\mathrm{T}}\mathrm{M}\rho\right) = 1^{\mathrm{T}}\frac{d\mathrm{M}}{d\tau}\rho + 1^{\mathrm{T}}\mathrm{M}^*\frac{d\rho}{d\tau} = 1^{\mathrm{T}}(\mathrm{A} - \mathrm{K})\rho + 1^{\mathrm{T}}\mathrm{K}^*\rho = 0.$$

DOI: 10.1002/fld

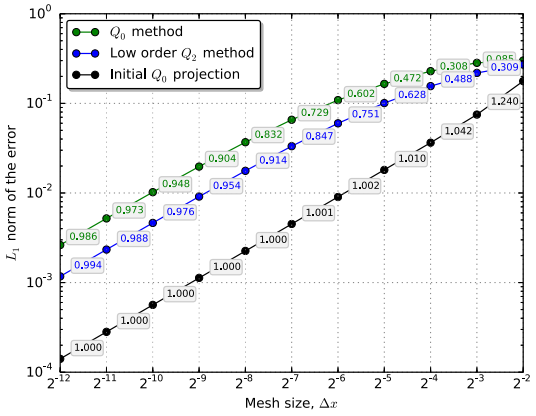


Figure 1. Convergence rates of low-order DG remap schemes in 1D: comparison of the Q_0 method and the low-order Q_2 method (22); both methods use the same initial Q_0 projection.

It is interesting to note that, because of the upwinding (7) of the face fluxes, the off-diagonal (element-wise) blocks in $\mathbf{K}$ have non-negative entries in the Bernstein basis. As a consequence, in this case, the discrete upwinding of $\mathbf{K}$ is always local to each element.

Let us consider the 1D example from Section 2.2. We point out that if we apply discrete upwinding directly to $\mathbf{M}^{-1}\mathbf{K}$, the result will have dependencies that are not consistent with the upwind direction which, in this example, means that there will be positive entries below the diagonal. On the other hand, using mass lumping of $\mathbf{M}$ and discrete upwinding of $\mathbf{K}$ leads to the low-order scheme

$$\frac{d}{d\tau}\begin{pmatrix}\rho_i^{(0)}\\ \rho_i^{(1)}\\ \rho_i^{(2)}\end{pmatrix} = \frac{u}{h}\begin{pmatrix}-3 & 2 & 1 & 0\\ 0 & -2 & 2 & 0\\ 0 & 0 & -3 & 3\end{pmatrix}\begin{pmatrix}\rho_i^{(0)}\\ \rho_i^{(1)}\\ \rho_i^{(2)}\\ \rho_{i+1}^{(0)}\end{pmatrix} \tag{22}$$

where only upwind dependencies are present — the matrix is upper triangular.

While this form of simple upwinding is an effective way to ensure monotonicity, it is far too diffusive (first-order accurate at best; see Figure 1) to be useful in practice. Instead, we propose here a simple algorithm that localizes the upwinding process only to the regions where the LED property is violated.

First, we need a good 'monotonicity measure' to inform our choice as to where to apply the discrete upwinding. This measure, associated with the DOF and denoted as $\mu(\rho)$, is such that $\mu_i(\rho) \in [0, 1]$ should vanish ($\mu = 0$) where the monotonicity bounds (17) are not violated, and only be active ($\mu = 1$) where (17) does not hold. A straightforward way to define $\mu(\rho)$ is to perform a high-order, linear advection step based on (10), and then based on the high-order solution ρ, compute

$$\mu_i(\rho) = \begin{cases}\beta\left(\rho_i - \rho_i^{\max}\right), & \text{if } \rho_i > \rho_i^{\max}\\ \beta\left(\rho_i^{\min} - \rho_i\right), & \text{if } \rho_i > \rho_i^{\min}\\ 0 & \text{otherwise.}\end{cases} \tag{23}$$

The function $\beta : \mathrm{R}^+ \to [0, 1]$ can be defined in many ways;[6] the simplest case is to return the value one for any argument. One can also use smoothly varying functions of the argument designed to transition between the 'off' state ($\mu = 0$), for small violations of monotonicity, and the 'on' state ($\mu = 1$), for larger violations.

[6][Correction added on 27 November 2014, after first online publication: 7 was remove after R^+]

DOI: 10.1002/fld

Given the 'monotonicity measure' μ, we define local lumping and upwinding matrices $\mathbf{L}_\mu$ and $\mathbf{D}_\mu$, which are symmetric and have zero row sums and off-diagonal entries

$$\left(\mathbf{L}_\mu\right)_{ij} = -\max\left\{\mu_i, \mu_j\right\}\mathbf{M}_{ij}, \quad \left(\mathbf{D}_\mu\right)_{ij} = \max\left\{\mu_i, \mu_j\right\}\max\left\{0, -\mathbf{K}_{ij}, -\mathbf{K}_{ji}\right\}, \qquad i \neq j.$$

Note that if $\mu_i = 1$, the off-diagonal entries in the i-th row of $(\mathbf{M}+\mathbf{L}_\mu)^{-1}(\mathbf{K}+\mathbf{D}_\mu)$ are non-negative, so the LED property holds there. In other words, if μ has marked the DOF where LED is violated, the previous local modification will fix them to be monotonic.

This motivates the following iterative process, which we refer to as LSD:

0. Input: $\rho \approx \rho(\tau)$; output: $\rho^{new} \approx \rho(\tau + \Delta\tau)$.
1. Compute the monotonicity bounds ρ^{min} and ρ^{max} from (17).
2. Set $\mu = 0$.
3. Compute ρ^{new} from ρ while applying partial upwinding $\mathbf{K}(\tau) \to \mathbf{K}(\tau) + \mathbf{D}_\mu(\tau)$ and lumping $\mathbf{M}(\tau) \to \mathbf{M}(\tau) + \mathbf{L}_\mu(\tau)$ at every stage of the time-stepping algorithm.
4. Compute monotonicity error: $\mu^e = \mu^e(\rho^{new})$ where μ^e is defined by (23) using $\beta(t) = t$.
5. Check for convergence: if $\|\mu^e\|_\infty < \epsilon$, then exit the LSD iteration.
6. Compute the new monotonicity measure: $\mu^{new} = \mu(\rho^{new})$.
7. Set $\mu_i = \max\left\{\mu_i, \mu_i^{new}\right\}$, and go to step 3.

Clearly, the previous algorithm will do nothing if the high-order solution is already monotonic, that is, $\mu(\rho^{new}) = 0$, so in particular, it will preserve linear fields.

In general, there is no theoretical guarantee that the LSD algorithm will converge, and in some very rare cases, we have, in fact, observed stagnation. This condition is easy to detect: if the entries of μ remain unchanged during the update in step 7, that is, if $\mu_i^{new} \leqslant \mu_i$, $\forall i$. In practice, we also limit the number of LSD iteration to 20; however, in our experience, the algorithm generally converges much faster, in three to four iterations. The numerical tests presented in Section 4 use the sharp cut-off function $\beta(t) = \min\{1, 10^{12}t\}$ in (23) for step 6, and the stopping tolerance $\epsilon = 10^{-10}$.

3.2. Generalized high-order FCT

A method that guarantees monotonicity without an outer iteration can be constructed using the ideas of FCT. The basic philosophy of an FCT approach is to begin with a method that is guaranteed to be monotonic (by compromising accuracy) and then seek local non-linear corrections to the flux, which improve accuracy while maintaining monotonicity (in contrast to LSD, which starts from the high-order solution and makes non-linear corrections to render it monotonic). This is accomplished by adding back in the difference between the diffusive approximation and the high-order approximation (known in FCT literature as 'anti-diffusion') in a controlled manner. The key to maintaining conservation will be adding the anti-diffusion in an anti-symmetric flux form. To guarantee monotonicity, we examine the consequences of anti-diffusive flux contributions and explicitly scale back or set to zero those terms, which could generate bounds violations in the worst case of all positive or all negative fluxes acting in concert with no canceling by fluxes of the opposite sign. According to (19), the equation

$$\mathrm{M}^* \frac{d\rho}{d\tau} = \mathrm{K}^*\rho + (\mathrm{M}^* - \mathrm{M})\frac{d\rho}{d\tau} - \mathrm{D}\rho \tag{24}$$

can be seen to be equivalent to the original high-order equation (10). It can also be interpreted as the low-order equation plus two correction terms. The correction terms account for the discrepancy between the low-order and high-order equations in the lumped mass matrix and in the addition of explicit diffusion.

Following Kuzmin and Turek [13], we now show that both of these terms can be put into an anti-symmetric flux form. By doing so, we afford the ability to modify the fluxes (as long as the modifications are symmetric) while retaining conservation. Because $\mathbf{D}$ has zero row sums, we can write

$$(\mathrm{D}\rho)_i = \sum_{i \neq i} \mathrm{D}_{ij}\left(\rho_j - \rho_i\right). \tag{25}$$

DOI: 10.1002/fld

Because $\mathbf{L}=\mathbf{M}^{*}-\mathbf{M}$ has zero row sums (see (20) and (21)), we can also write (26)

$$\left(\mathrm{M}^{*}\frac{d\rho}{d\tau}-\mathrm{M}\frac{d\rho}{d\tau}\right)_{i}=\sum_{j\neq i}\mathrm{L}_{ij}\left(\frac{d\rho_{j}}{d\tau}-\frac{d\rho_{i}}{d\tau}\right)=\sum_{j\neq i}\mathrm{M}_{ij}\left(\frac{d\rho_{j}}{d\tau}-\frac{d\rho_{i}}{d\tau}\right) \tag{26}$$

We now have the semi-discrete flux form required for the anti-diffusion:

$$f_{ij}=\left(\mathrm{M}_{ij}\frac{d}{d\tau}+\mathrm{D}_{ij}\right)\left(\rho_{i}-\rho_{j}\right). \tag{27}$$

Because it is the anti-symmetry $f_{ij}=-f_{ji}$ that ensures conservation, any symmetric scaling of the fluxes

$$f_{ij}^{*}=\alpha_{ij}f_{ij},\quad \text{where}\quad \alpha_{ij}=\alpha_{ji}, \tag{28}$$

is also conservative. Our modified equation is then

$$\mathrm{M}^{*}\frac{d\rho}{d\tau}=\mathrm{K}^{*}\rho+f^{*} \tag{29}$$

where

$$f^{*}=\sum_{j\neq i}\alpha_{ij}f_{ij} \tag{30}$$

are the limited anti-diffusive corrections. When $\alpha_{ij}\equiv 1$, the previous equations are just a different form of (10), while for $\alpha_{ij}\equiv 0$, they reduce to the low-order approximation (18). The fact that intermediate values for α_{ij} could be picked to interpolate from the latter case to the former case is the main idea in FCT.

To formulate this on fully discrete level, we consider a forward Euler discretization for the high-order and low-order equations:

$$\mathbf{M}\Delta\rho^{H}=\Delta\tau\mathbf{K}\rho, \tag{31}$$

$$\mathbf{M}^{*}\Delta\rho^{L}=\Delta\tau\mathbf{K}^{*}\rho. \tag{32}$$

Note that (31) can be expanded into a form analogous to (24):

$$\mathbf{M}^{*}\Delta\rho^{H}=\mathbf{M}^{*}\Delta\rho^{L}+(\mathbf{M}^{*}-\mathbf{M})\Delta\rho^{H}-\Delta\tau\mathbf{D}\rho. \tag{33}$$

This is precisely the high-order update, written as a low-order update plus the anti-diffusion correction terms. Putting the anti-diffusion terms into flux form, we have

$$\left(\mathrm{M}^{*}\Delta\rho^{H}\right)_{i}=\left(\mathrm{M}^{*}\Delta\rho^{L}\right)_{i}+\sum_{j\neq i}\mathrm{M}_{ij}\left(\Delta\rho_{i}^{H}-\Delta\rho_{j}^{H}\right)+\Delta\tau\sum_{j\neq i}\mathrm{D}_{ij}\left(\rho_{i}-\rho_{j}\right) \tag{34}$$

or

$$\left(\mathrm{M}^{*}\Delta\rho^{H}\right)_{i}=\left(\mathrm{M}^{*}\Delta\rho^{L}\right)_{i}+\sum_{j\neq i}f_{ij}. \tag{35}$$

DOI: 10.1002/fld

If we insert the flux limiting factors $\alpha_{ij} = \alpha_{ji}$, we have an update, which is now an approximation to the high-order update:

$$(\mathrm{M}^* \Delta\rho)_i = \left(\mathrm{M}^* \Delta \rho^L\right)_i + \sum_{j \neq i} \alpha_{ij} f_{ij}, \tag{36}$$

with a way to control the contributions from the anti-diffusive fluxes.

The procedure for choosing the α_{ij} follows analogously to traditional FCT. First, we choose allowable bounds on the solution, examine the consequences of the anti-diffusive corrections to see if they could lead to bounds violations, and then use a simple 'worst case' analysis to choose α_{ij} that guarantees monotonicity with a single step procedure.

The solution bounds are defined by the extremal values within the neighborhood N_i defined by the non-zero entries in the advection matrix $\mathbf{K}$, as discussed previously in Section 3. We can then form the allowable changes because of anti-diffusive corrections:

$$\Delta\rho_i^{A+,\max} = \rho_i^{\max} - \rho_i^L, \qquad \Delta\rho_i^{A-,\max} = \rho_i^L - \rho_i^{\min}, \tag{37}$$

where ρ_i^L is the solution updated by the low-order approximation. Finding a set of α_{ij} that prevents bounds violations but is otherwise as close to unity as possible is a globally coupled problem. However, we can obtain an admissible set of α_{ij} that guarantees monotonicity by assuming a 'worst case' situation where all positive or negative anti-diffusive fluxes contribute together to increase or decrease a solution value, while all fluxes of the opposite sign have been canceled. Under this scenario, we compute:

$$\Delta\rho^{A\pm} = \frac{1}{\mathrm{M}^*_{ii}} \sum_{j \neq i} \mathrm{f}_{ij}^{\pm}. \tag{38}$$

The positive or negative fluxes associated with a solution value must be scaled back if $\Delta\rho^{A\pm}$ exceeds $\Delta\rho^{A\pm,\max}$:

$$\alpha_i^{\pm} = \min\left(1, \frac{\Delta\rho^{A\pm,\max}}{\Delta\rho^{A\pm}}\right). \tag{39}$$

Each flux will then potentially have two constraints associated with it, one from each solution value it influences. We can guarantee no violation of bounds if we choose the most restrictive constraint for each flux:

$$\alpha_{ij} = \begin{cases} \min\left(\alpha_i^+, \alpha_j^-\right) & \mathrm{f}_{ij} > 0 \\ \min\left(\alpha_i^-, \alpha_j^+\right) & \mathrm{f}_{ij} < 0. \end{cases} \tag{40}$$

Because $\mathbf{f}_{ij} = -\mathbf{f}_{ji}, \alpha_{ij} = \alpha_{ji}$.[7]

Using the new monotone update $\Delta\rho$ in place of the low-order update $\Delta\rho^L$, we can now repeat the FCT process. The new high-order anti-diffusive fluxes are given by $\mathbf{f}_{ij} \leftarrow (1 - \alpha_{ij})\mathbf{f}_{ij}$, that is, the remainder of the fluxes that was not used as a correction to define $\Delta\rho$ from $\Delta\rho^L$. In other words, instead of (35), the starting point now is

$$\mathbf{M}^* \Delta\rho^H = \mathbf{M}^* \Delta\rho + \sum_{j \neq i} (1 - \alpha_{ij}) \mathbf{f}_{ij}.$$

This process can be repeated multiple times and is sometimes referred to as *iterated* FCT, as originally developed in [14].

In order to use the FCT algorithm in a high-order multistage RK method, we replace the ODE right-hand side $\mathbf{M}^{-1}\mathbf{K}\rho = \Delta\rho^H/\Delta\tau$ with the term $\Delta\rho/\Delta\tau$ at every stage of the RK method. In addition,

[7] [Correction added on 27 November, after first online publication: *i* has been corrected to *j* in equation 40]

DOI: 10.1002/fld

at every stage, new bounds ρ^{min} and ρ^{max} are computed. The value of $\Delta\tau$ is important for the FCT algorithm because smaller $\Delta\tau$ will generally produce a slope vector $\Delta\rho/\Delta\tau$ that is closer to $\Delta\rho^H/\Delta\tau$. In order to ensure monotonicity with SSP methods, we use the same $\Delta\tau$ in the FCT algorithm as the full RK pseudo-time step.

3.3. OBR

Our third method for ensuring monotonicity consists of two main steps: (i) given a vector $\rho \approx \rho(\tau)$, compute a high-order approximation $\rho^H \approx \rho(\tau+\Delta\tau)$ to problem (10), using, for example, an explicit multistage RK method; (ii) define the new density vector $\rho^{new} \approx \rho(\tau+\Delta\tau)$ by solving an optimization problem that ensures the following properties for ρ^{new}: it is close to ρ^H, has the same mass as ρ^H and satisfies the monotonicity bounds (17). Specifically, in step 2, we use the mass-variable mass-target (MVMT) version of the OBR algorithm; see [5, 15, 16]. The MVMT-OBR problem reads

$$\|\rho^{new}-\rho^H\|\ell_2 \mapsto \min, \quad \mathbf{m}^T\rho^{new} = \mathbf{m}^T\rho^H, \quad \rho_i^{\min} \leqslant \rho_i^{new} \leqslant \rho_i^{\max} \quad \forall i, \tag{41}$$

where $\mathbf{m}=\mathbf{M}(\tau+\Delta\tau)\mathbf{1}$ is the vector of the row sums of the mass matrix at pseudo-time $\tau+\Delta\tau$; $\|\cdot\|\ell_2$ denotes the Euclidean vector norm, $\|\mathrm{x}\|_{\ell_2}^2 = \mathbf{x}^T\mathbf{x} = \sum_i \mathbf{x}_i^2$; and $\rho^{\min}, \rho^{\max}$ are the vectors of local lower and upper bounds in (17). This problem can be solved efficiently [16], leading to a fast one step method that is both conservative and LED monotonic.

The OBR approach is very general and has been applied successfully in low-order remap settings [17] and is similar in nature to the 'repair' approach in [18]. However, the fact that it is not based on a semi-discrete formulation has the potential drawback that physical intuition might be difficult to express in terms of formal constraints. For example, the solution of (41) may change ρ^H globally, even though the violation of the bounds is local. This is due to the use of the ℓ_2 norm to measure the error between ρ^{new} and ρ^H, and while other, more localized choices of norms are possible, for example, ℓ_1 norm, they lead to other issues such as non-uniqueness and increased difficulty in the solution of the optimization problem.

3.4. Pseudo-time step control

The number of time steps in pseudo-time will depend on the size of the mesh displacement relative to the size of the mesh elements. In effect, this is very similar to a CFL constraint on $\Delta\tau$. In practice, we control $\Delta\tau$ using the constraint

$$\Delta\tau \leqslant C \min_i \frac{\mathrm{M}^*_{ii}}{\mathrm{K}^*_{ii}},$$

where $\mathbf{M}^*$ and $\mathbf{K}^*$ are the fully lumped mass matrix and the fully upwinded advection matrix, respectively (see (19)–(21)), and $C \leqslant 1$ is a CFL constant. The motivation for this choice is the fact that *only* under this constraint the low-order method (18) based on $\mathbf{M}^*$ and $\mathbf{K}^*$ will be monotone in the LED sense. Generally, this choice is independent of the solution; however, when the matrices $\mathbf{M}$ and $\mathbf{K}$ (and, consequently, $\mathbf{M}^*$ and $\mathbf{K}^*$) depend on one of the fields being remapped, then $\Delta\tau$ may depend on the solution. This is the case when remapping density and specific internal energy, where density serves as a weight in $\mathbf{M}_e$ and $\mathbf{K}_e$, see Section 2.3.

4. NUMERICAL RESULTS

In this section, we consider a set of numerical experiments, which are designed to verify our high-order DG advection method as well as each of the monotonicity methods presented. We consider remap examples where the mesh motion is prescribed by some predetermined analytic motion as well as our target application of ALE hydrodynamics, where the mesh motion is a complicated function of the hydrodynamic motion. In each case, we discuss the differences and similarities of each approach and highlight their unique features.

DOI: 10.1002/fld

4.1. Remap of analytic functions with prescribed mesh motion

We first compare different remapping strategies using (10) with a 2D Q_2 mesh and a discontinuous Q_2 Bernstein polynomial basis on a series of test functions representing the initial field on a unit square. The motion of the mesh is prescribed by a displacement function, which rotates the mesh through 30° while keeping the boundary fixed. The intent is to verify and examine the behavior of the remap algorithms in a relatively simple setting. Subsequent sections test the algorithm on more demanding problems.

The first test function is a simple 2D linear field $\rho = x + y$. We consider it a design principle that any admissible scheme must preserve linear functions. Indeed, all of the monotone schemes (LSD, FCT and OBR), as well as the unmodified DG scheme, exhibit this property. However, it is worth noting that the 'discrete upwinding' procedure by itself does not. These results are shown in Figure 2.

The next test problem is a smooth function, $\rho = 1 + \sin(\pi x)\sin(\pi y)$, with a maximum at the center. A comparison of the unmodified DG scheme and the three monotone schemes is shown in Figure 3. They are essentially indistinguishable by eye, as we might expect for a smooth test function. However, an examination of the convergence behavior shows that the methods do not produce identical results. The convergence of the four methods in the L_1 and L_∞ norms for this test function is shown in Figure 4. These calculations were performed using an 8×8, 16×16 and 32×32 Q_2 element sequence of meshes for each method. The convergence can be seen to be greater than second order for each of the methods. In the subsequent section, we examine the more challenging case of an extremal point translating through the mesh, whereas in this test, it remains at a fixed location.

Next, we examine the behavior of three models of a transition between the values of 0 and 1, of increasing sharpness. These can be thought of as simple models for a shock profile or a contact discontinuity, which may be smeared by an artificial viscosity or by monotone projection of a material interface not aligned with a mesh.

The smoothest version of this test is a profile $\rho = \pi/2 + \arctan(w(x - 1/2))$. The parameter w controls the width of the profile and is set to $w = 20$ for these tests. With the 20×20 element mesh used in the previous test problems, the results are again very close for all of the schemes (Figure 5). The convergence behavior is shown in Figure 6.

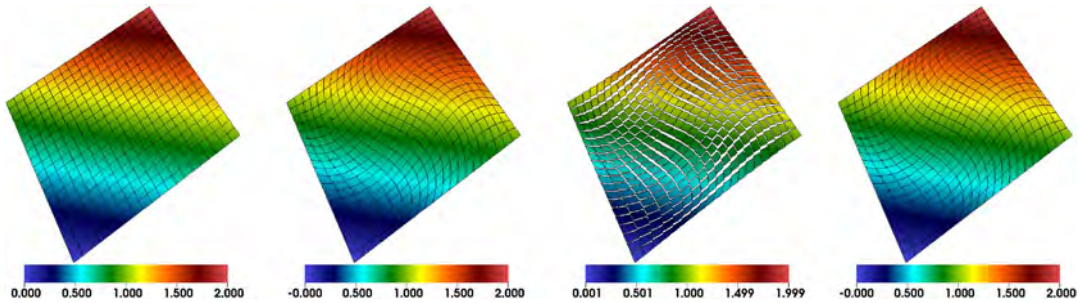
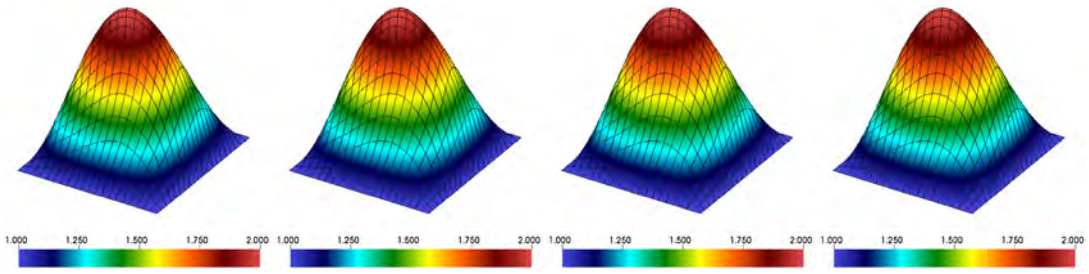


Figure 2. Results for remap of linear density field (*left to right*): initial mesh and density, high-order result, fully upwinded result, locally scaled diffusion result (representative of all monotone schemes).

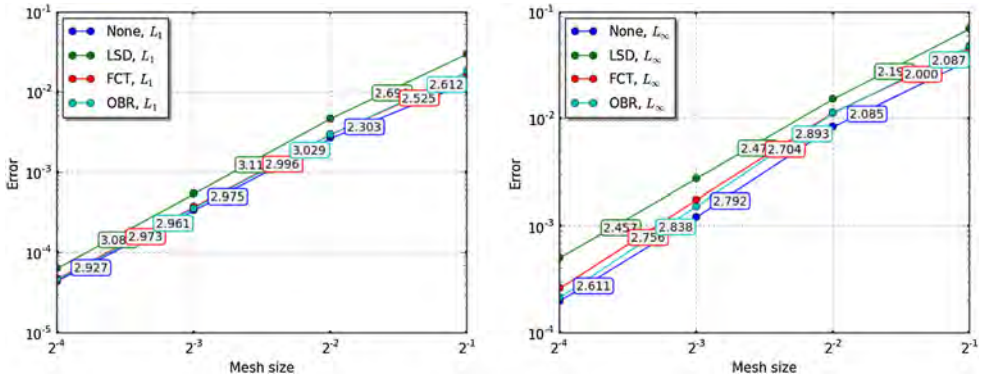


(a) Unmodified DG (b) LSD (c) FCT (d) OBR

Figure 3. Results for remap of smooth field $\rho = 1 + \sin(\pi x)\sin(\pi y)$.

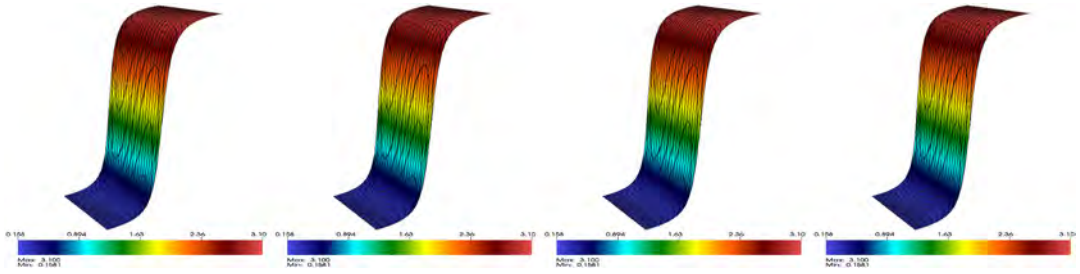
Published 2014. This article is a U.S. Government work and is in the public domain in the USA.

Int. J. Numer. Meth. Fluids 2015; **77**:249–273
DOI: 10.1002/fld



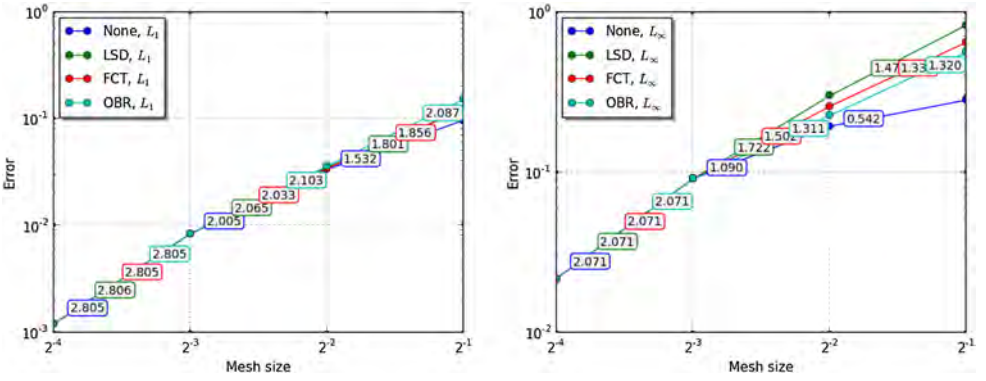
(a) L_1 (b) L_∞

Figure 4. Convergence behavior for remap strategies of smooth field $\rho = 1 + \sin(\pi x)\sin(\pi y)$.



(a) Unmodified DG (b) LSD (c) FCT (d) OBR

Figure 5. Results for remap of arctan field.



(a) L_1 (b) L_∞

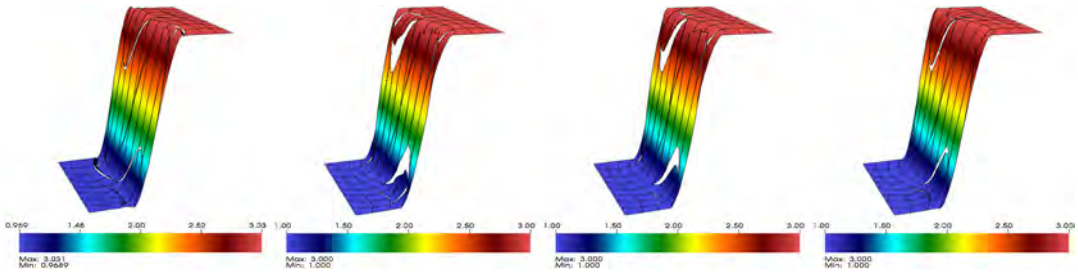
Figure 6. Convergence behavior of remap strategies of arctan field.

A profile that is only C^1 continuous can be formed from piecewise quadratic profiles. This profile also has the property that it has zero initial projection error, so that any error in the solution reflects only the remapping error. The results on an 8×8 grid, and convergence behavior are shown in Figures 7 and 8. A coarser grid was used than in other test problems to make differences in the solution more apparent.

A test problem that exercises the monotonicity strategies is the remap of a step function, using the same prescribed rotational motion. The results are shown in Figure 9. The unmodified DG method exhibits undershoots and overshoots as expected, but all three of the monotone schemes preserve the minimum and maximum values in the initial condition. While there are some qualitative differences in the solution, none of the methods stands out as clearly superior by inspection. The convergence behavior of the methods is shown in Figure 10. All methods have similar

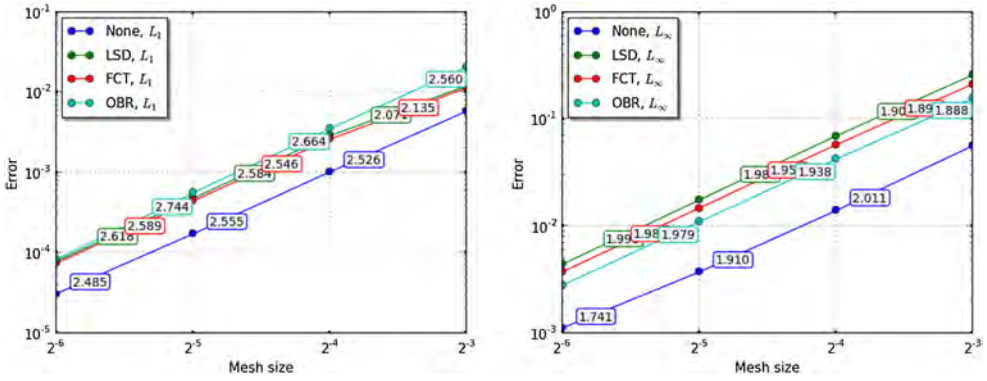
Published 2014. This article is a U.S. Government work and is in the public domain in the USA.

DOI: 10.1002/fld



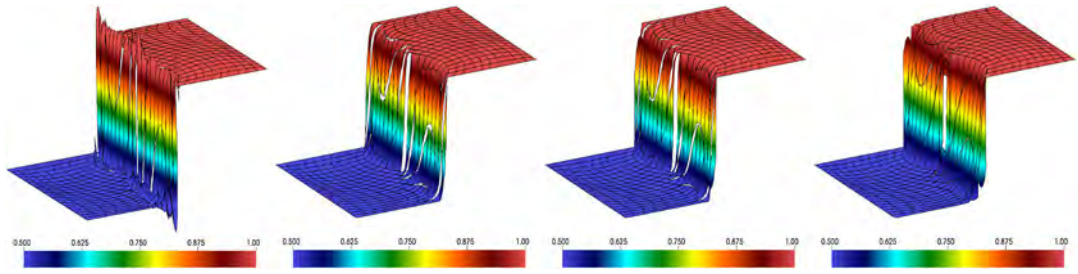
(a) Unmodified DG (b) LSD (c) FCT (d) OBR

Figure 7. Results for remap of a piecewise quadratic field.



(a) L_1 (b) L_∞

Figure 8. Convergence behavior of remap strategies of piecewise quadratic field.



(a) Unmodified DG (b) LSD (c) FCT (d) OBR

Figure 9. Results for remap of a step function.

convergence rates in L_1, while the OBR method seems to have slightly larger error magnitudes at any given resolution.

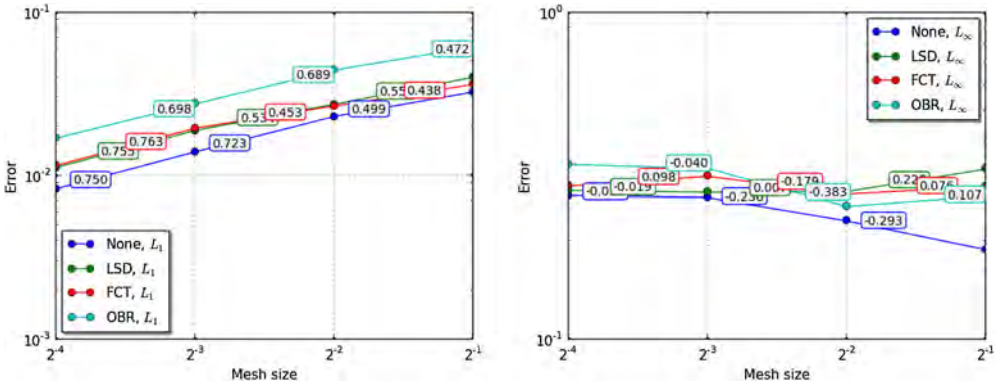
4.2. Rotation of a smooth profile

The goal of this test is to investigate and compare the performance of the three monotonicity approaches applied to a smooth field subject to a relatively large amount of advection. Specifically, we use the following smooth function (Figure 11) as initial data:

$$\rho(x,y) = \exp\left\{-r_t^2\left[\left(\frac{x-p_x}{r_x}\right)^2 + \left(\frac{y-p_y}{r_y}\right)^2 - 1\right]^2\right\} + 1,^8$$

[8]Correction added on 27 November 2014, after first online publication: terms x in px and rx were made subscripts on the equation.

DOI: 10.1002/fld



(a) L_1 (b) L_∞

Figure 10. Convergence behavior of remap strategies on step function.

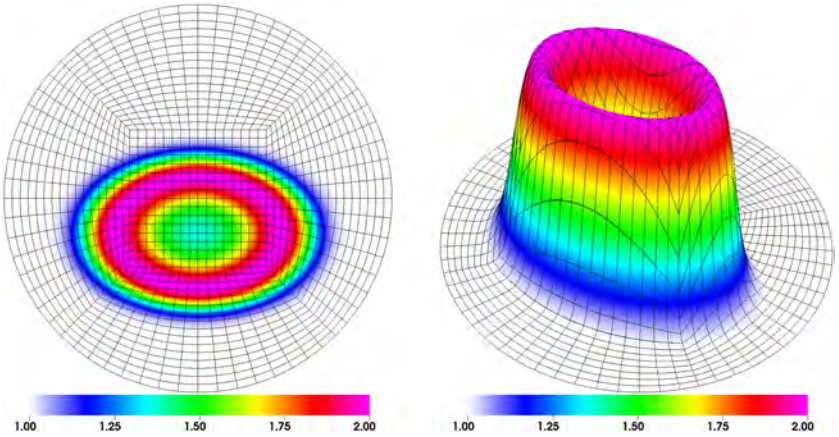


Figure 11. Initial smooth field: view from the top (left) and a 3D view (right).

where we set

$$p_x = 0, \qquad p_y = -0.5, \qquad r_x = 1.2, \qquad r_y = 0.8, \quad \text{and} \quad r_t = 1.$$

This specific field was chosen such that it has a (global) maximum along a curve (ellipse) and not just a point. This feature will test and emphasize the behavior of the three methods near maximum.

We advect the field using the following procedure: on the coarse mesh ($h = 1/2$ in the central mesh block), we perform 180 steps consisting of

a. remapping the field from the original mesh to the mesh rotated by 2° in counterclockwise direction;
b. moving the rotated mesh back to the original location without modifying the discrete representation of the field.

With each level of (dyadic) uniform mesh refinement, we double the number of steps and halve the angle of rotation of the mesh in step a.

The domain is the disk centered at the origin with radius $r = \sqrt{8}$, and we use a sequence of quadratic (Q_2) meshes conforming to the curved boundary. The spatial discretization uses Q_2 discontinuous finite elements. For pseudo-time integration, in this case, we use the SSP RK method with three stages [12], in order to match the third-order spatial discretization.

In Figure 12, we present a convergence study of the original DG scheme and the three monotonicity algorithms: LSD, FCT and OBR. The results show that the original DG method is close to third-order accurate in both the L_1 and L_∞ norms. On the other hand, all three monotone schemes exhibit reduced convergence rates of about 2–2.5 in the L_1 norm and 1.5 in the L_∞ norm. The

Published 2014. This article is a U.S. Government work and is in the public domain in the USA.

DOI: 10.1002/fld

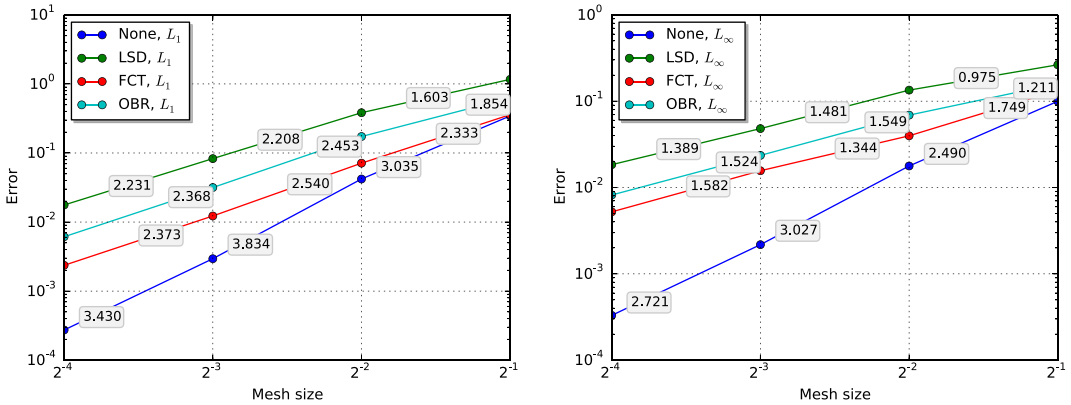


Figure 12. Convergence plots for the smooth field rotation test; L_1 (left) and L_∞ (right) norms. The number indicates the convergence rates.

reason for this reduction, as illustrated in Figure 13, is the large error near the ellipse where the maximum of the smooth field is located. For all three monotonicity algorithms, the error is clearly localized around the ellipse and focuses near it under refinement. Overall, on this test, FCT performs best of the three, followed by OBR and then LSD.

Closer inspection of the error for OBR reveals an interesting observation; see Figure 14: there is a small shift in the error and the field itself in the region away from the maximum where the field should stay very close to the value of 1 at all times. One possible explanation of this behavior is the fact that OBR uses a non-local mechanism to distribute the mass that is being removed (added) because of clipping of overshoots (undershoots).

4.3. Solid body rotation on unstructured mesh

This test problem is based on the solid body rotation benchmark as described in [2, 9] where three shapes are projected onto a mesh as an initial condition, then subject to a 360° rotation of the mesh. In this case, we consider an unstructured disk mesh with the so-called 'butterfly' topology, with the shapes initialized at the points

$$\begin{aligned}\vec{r}_1 &= 1.5\{\cos(\pi/6), \sin(\pi/6)\},\\ \vec{r}_2 &= 1.5\{\cos(3\pi/2), \sin(3\pi/2)\},\\ \vec{r}_3 &= 1.5\{\cos(5\pi/6), \sin(5\pi/6)\}.\end{aligned}$$

The initial conditions for the density field are defined according to the function[9]

$$\rho(x,y) = cone\left(x,y,\vec{r}_1\right) + hump\left(x,y,\vec{r}_2\right) + cyl\left(x,y,\vec{r}_3\right) + 1, \tag{42}$$

where

$$cone(x,y,\vec{r}) = \begin{cases} 0, & r(x,y,\vec{r}) > 1 \\ 1 - r(x,y,\vec{r}), & otherwise \end{cases}$$

$$hump(x,y,\vec{r}) = \begin{cases} 0, & r(x,y,\vec{r}) > 1 \\ \frac{1}{4}\left(1 + \cos\left(\pi \min\left(r\left(x,y,\vec{r}\right),1\right)\right)\right), & otherwise \end{cases}$$

$$cyl(x,y,\vec{r}) = \begin{cases} 0, & r(x,y,\vec{r}) > 1 \\ 0, & \left(y > 0.7\,\vec{r}_y\right) \,\&\, y < (1.3\,\vec{r}) \,\&\, \left(x > 1.4\vec{r}_x\right) \\ 1, & otherwise \end{cases}$$

[9][Correction added on 27 November 2014, after first online publication: *humpr* has been corrected to *hump* in (equation 42)]

DOI: 10.1002/fld

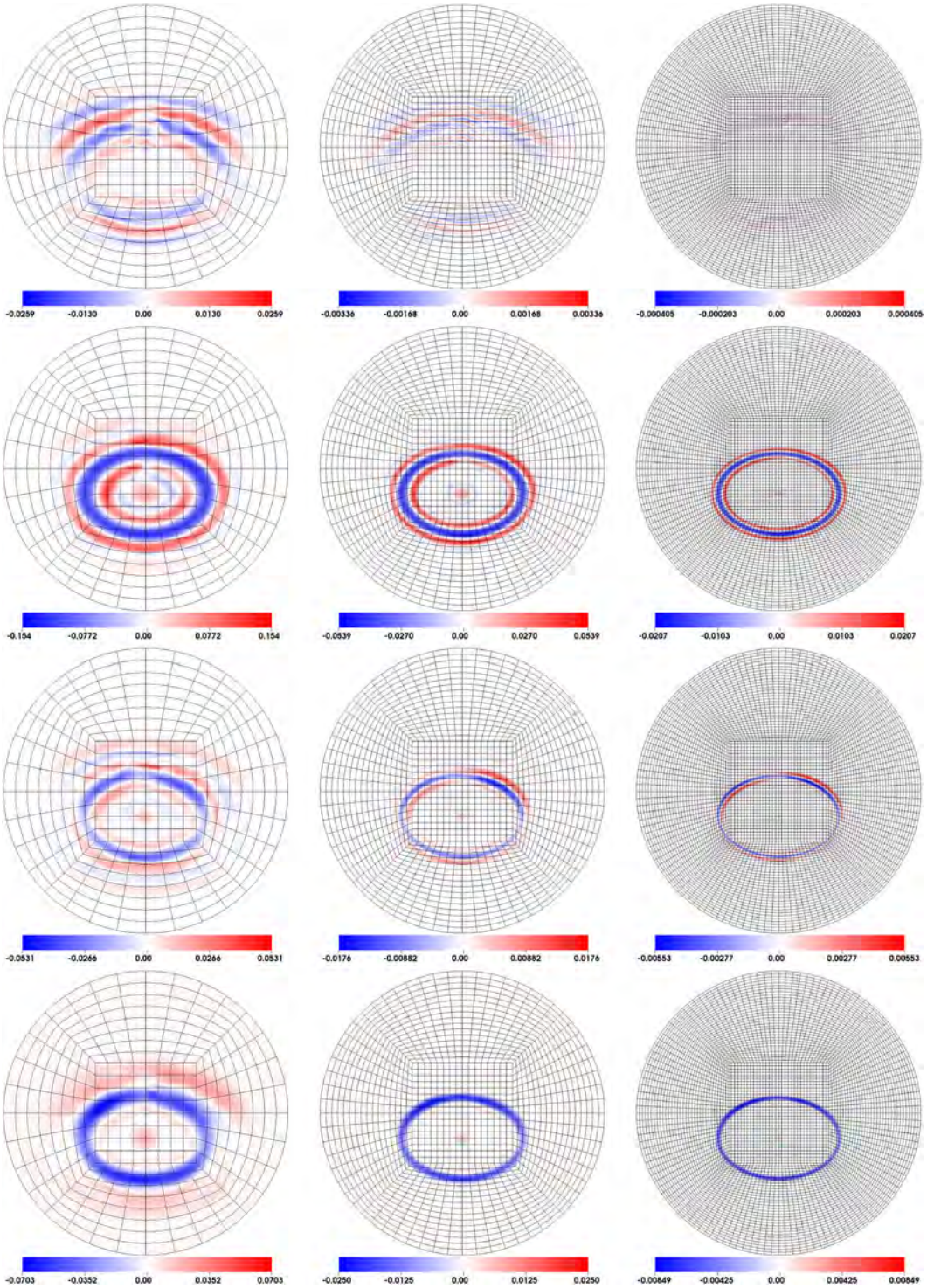


Figure 13. Plots of the error after one full 360° rotation; left to right: three levels of uniform mesh refinement; top to bottom: monotonicity algorithm: none, LSD, FCT and OBR, respectively. Each plot uses the exact range of values (symmetric around zero) spanned by the error function.

and the distance function is defined as

$$r\left(x, y, \vec{r}\right) = \frac{1}{0.7|\vec{r}|}\sqrt{\left(x - \vec{r}_x\right)^2 + \left(y - \vec{r}_y\right)^2}$$

DOI: 10.1002/fld

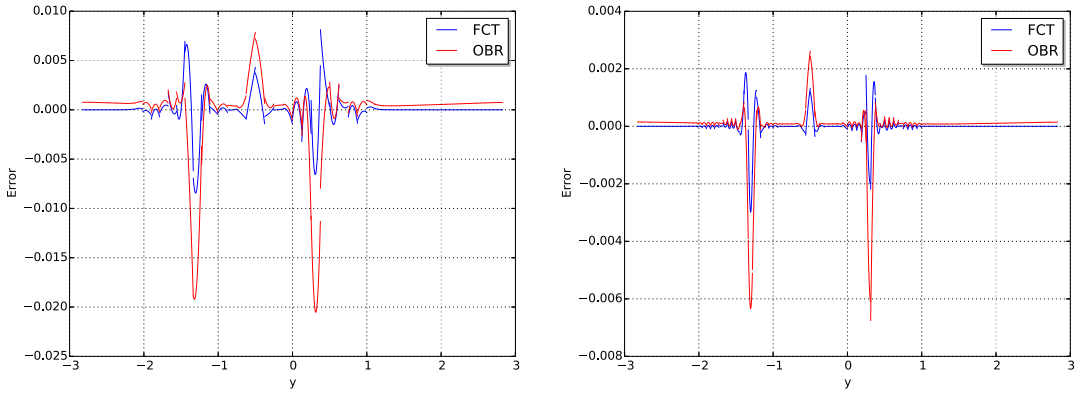


Figure 14. Comparison of the errors for the FCT and OBR monotonicity algorithms, cross section at $x = 10^{-3}$ for mesh sizes $h = 1/8$ (left) and $h = 1/16$ (right). Notice that the error for OBR is shifted away from zero in the regions away from the maximum and minimum.

We consider two cases for the initial density function defined via (42), each with the same total number of DOF:

1. A fine mesh using Q_2 discontinuous elements (nine DOF/element);
2. A coarse mesh using Q_4 discontinuous elements (25 DOF/element).

In each case, we use the positive Bernstein polynomial basis for both the initial projection (which ensures a monotone initial condition) and the DG remap basis. The initial conditions for each case are shown in Figure 15. We advect the density field for a full 360° rotation by prescribing the mesh motion to be a rotation about the origin of 1/4 degree for 1440 steps in a manner similar to the previous test problem of Section 4.2. For each step, the semi-discrete advection equation of (10) is solved in pseudo-time using an explicit RK2 time-integration method with two pseudo-time steps per remap step. For the FCT method, we use two iterations for the Q_2 case and four iterations for the Q_4 case.

The results for the Q_2 and Q_4 cases are shown in Figures 16 and 17, respectively. For each case, the unmodified high-order results exhibit spurious oscillations (monotonicity violations) at the boundaries of the slotted-cylinder shape (as expected) but do an excellent job of preserving the cone and hump shapes. Each of the monotone methods preserves the initial bounds of the function. The LSD method leads to more 'peak clipping' for the cone and hump functions, while FCT and OBR do a better job at preserving these features, consistent with the results of Section 4.2. For the Q_4 case, LSD gives the sharpest version of the slotted cylinder while both FCT and OBR generate some (monotone) fluctuations at the edges of the slot. While the quality of each monotone method is similar via the plots of Figures 16 and 17, there is a distinct difference in the computational cost of each method as shown in Table I. The LSD method is the most expensive (about three times the cost of OBR and FCT) in terms of run time because of the requirement of multiple (iterative) solves per remap step. The single pass nature of the OBR method makes it the best in terms of run-time performance, although the FCT method is very competitive, with only a slightly greater increase in relative run time for each case.

4.4. ALE shock triple-point problem

Finally, we explore our new methods in the context of our target application, ALE hydrodynamics, by considering a single material version of the so-called shock triple-point benchmark (e.g., as described in [19]). We modify the problem for the single material case by making each of the three regions an ideal gas with the gamma-law coefficient $\gamma = 1.5$.

We use Q_4 continuous finite elements for the kinematic variables (position, velocity, etc.) and Q_3 discontinuous finite elements for the internal energy combined with an explicit energy calculation

DOI: 10.1002/fld

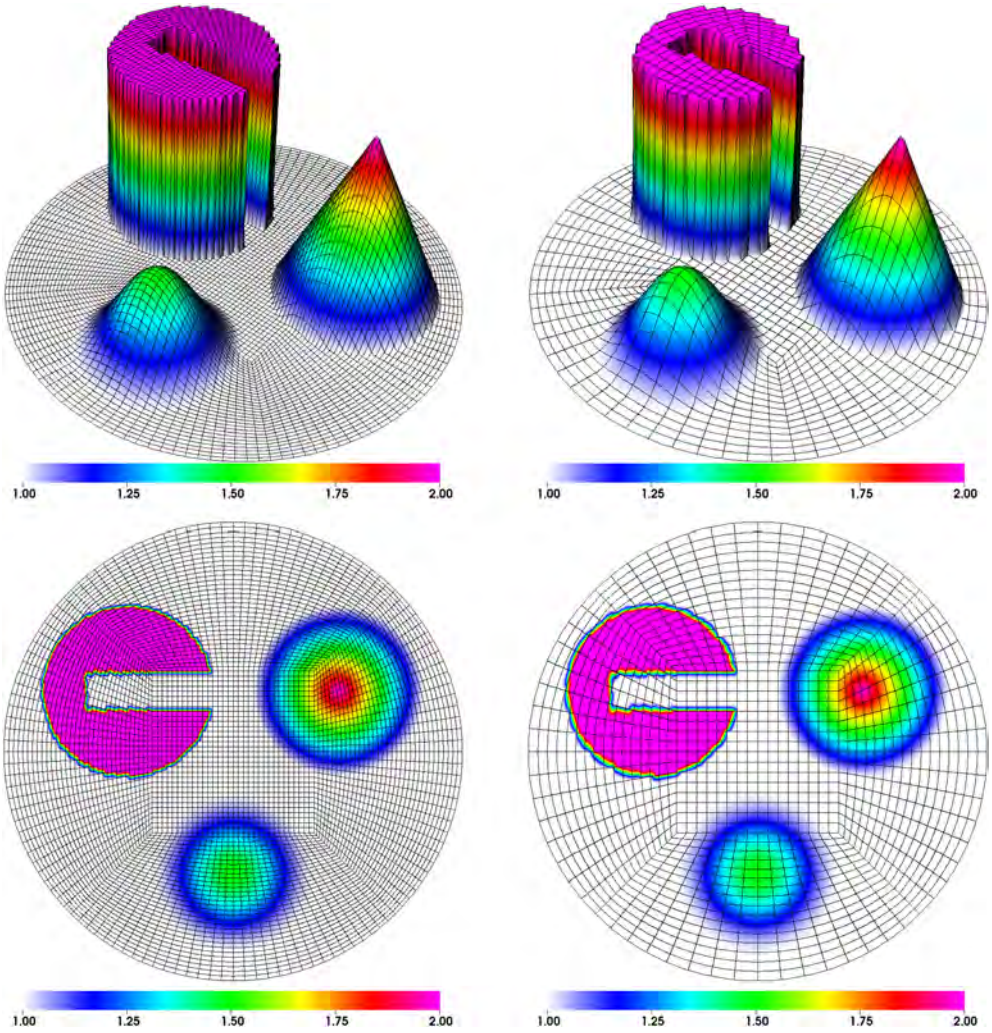


Figure 15. 3D view (*top*) and 2D view (*bottom*) of initial mesh and density using Q_2 elements on a fine mesh (*left*) and Q_4 elements on a coarse mesh (*right*).

conserving RK2 dynamic time-stepping algorithm for the Lagrangian phase of the ALE calculation. Every 20 Lagrange cycles, we apply two iterations of a curvilinear mesh relaxation algorithm followed by a pseudo-time advection remap where we use Q_4 continuous elements for the velocity (momentum), Q_4 discontinuous elements for the density and Q_3 discontinuous elements for the internal energy in solving (13), (12) and (14), respectively, in pseudo-time using an explicit RK2 time-stepping algorithm. For both the density and internal energy remaps, we apply our three different monotone methods (and also consider the unmodified version of these). We do not apply any monotonicity treatment for the velocity remap under the assumption that the field is made sufficiently smooth via artificial viscosity during the Lagrange phase.

In Figure 18, we show the plots of the ALE mesh and density fields at the final time of $t=5.0$ for the unmodified, LSD, FCT and OBR cases. We also show zoomed-in views of the vortical flow region around the triple point. The unmodified results exhibit spurious oscillations/ringing along the material interface, while the monotone methods do not. Each of the monotone methods yields essentially the same shock structures, but there are noticeable differences along the material interface. The LSD result yields the smoothest interface, while FCT and especially OBR exhibit some fluctuations along the interface. The relative run time and cycle counts of each result are given in Table II. Each of the monotone methods results

DOI: 10.1002/fld

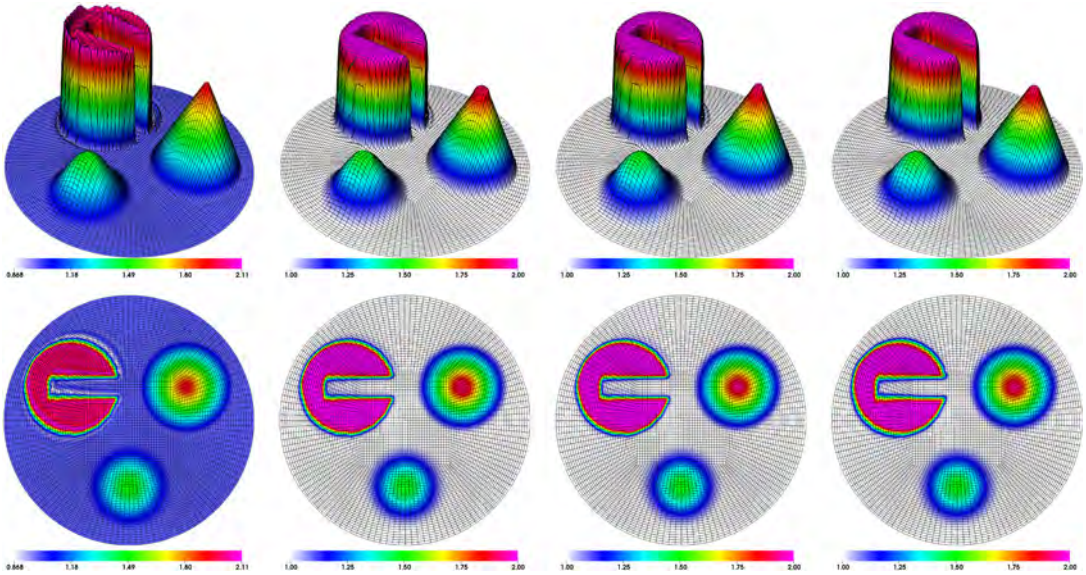


Figure 16. 3D view (*top*) and 2D view (*bottom*) of results for solid body rotation test on unstructured mesh using Q_2 elements (*left to right*): high-order result, LSD result, FCT result and OBR result.

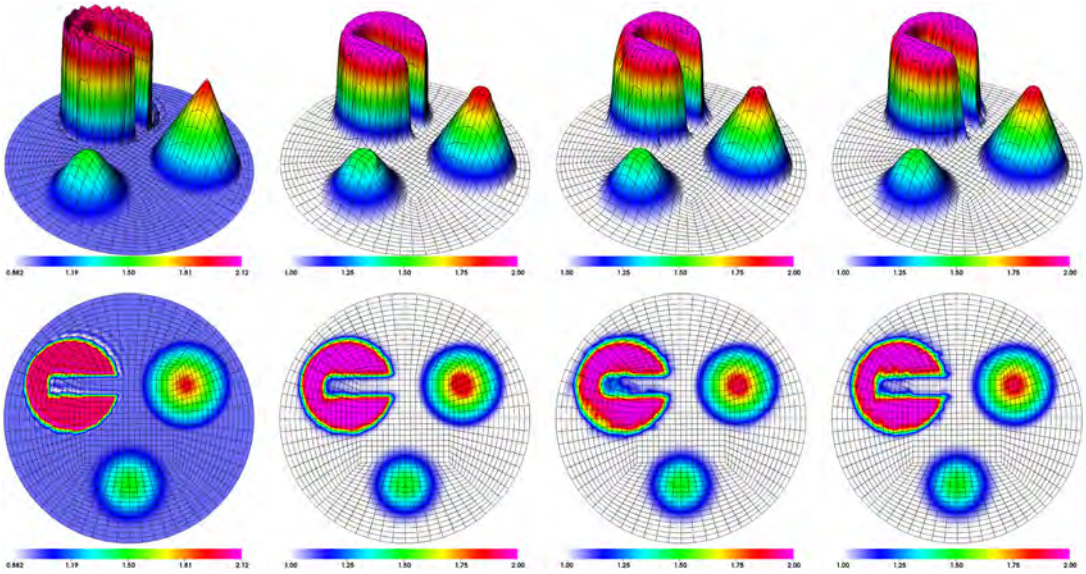


Figure 17. 3D view (*top*) and 2D view (*bottom*) of results for solid body rotation test on unstructured mesh using Q_4 elements (*left to right*): high-order result, LSD result, FCT result and OBR result.

Table I. Relative run times of each monotone method for the solid body rotation test problem.

Method	Relative run time
Unmodified Q_2	1.0
OBR Q_2	1.02
FCT Q_2	1.12
LSD Q_2	2.95
Unmodified Q_4	2.52
OBR Q_4	2.59
FCT Q_4	2.90
LSD Q_4	8.91

DOI: 10.1002/fld

in fewer overall cycles to complete the calculation, resulting in an overall speedup of the calculation for FCT and OBR. The three times cost of the LSD remap is significantly amortized via the 20 Lagrange steps per remap and the overall reduced cycle count but still results in the largest computational cost of all the methods.

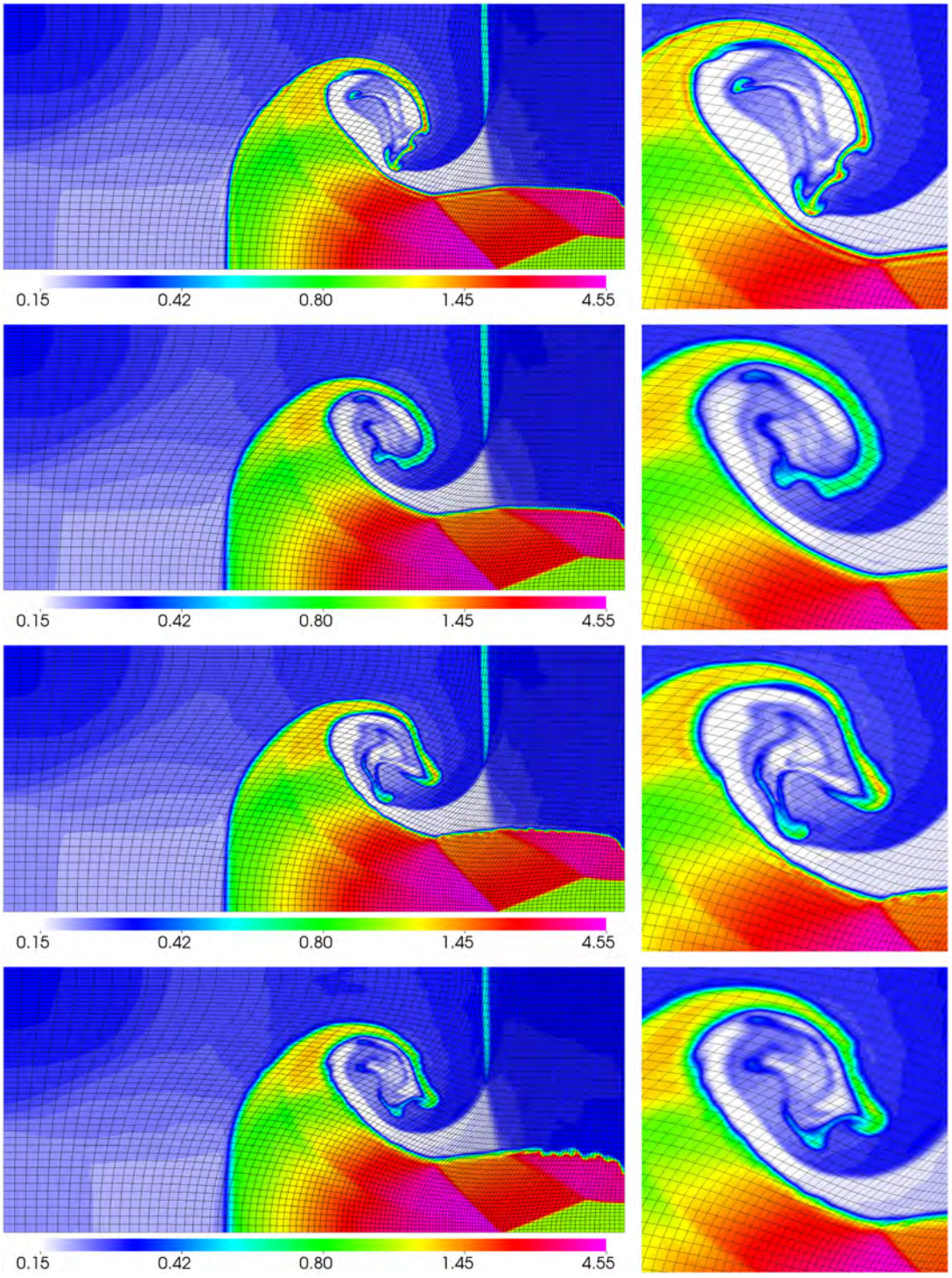


Figure 18. Results for ALE shock triple-point problem using Q_4 elements (*top to bottom*): high-order result, LSD result, FCT result and OBR result.

DOI: 10.1002/fld

Table II. Run times and cycle count of each monotone method, relative to unmodified Q_4 method, for the shock triple-point ALE hydrodynamics test problem.

Method	No. of cycles	Relative run time
Unmodified Q_4	42,586	1.0
OBR Q_4	33,650	0.79
FCT Q_4	39,459	0.95
LSD Q_4	27,538	1.35

5. CONCLUSIONS AND FUTURE WORK

We have presented three approaches for ensuring a monotone solution for advection equations using high-order DG finite element methods. Both the LSD and generalized high-order FCT approaches constitute new methods, and the application of OBR to high-order DG advection is, to our knowledge, new. Our particular interest in this problem is for the remap phase used in the context of high-order curvilinear ALE hydrodynamics where the goal is to transfer a given field quantity defined on a post-Lagrangian mesh to some new mesh. Based on the results of Section 4, each of the non-linear methods is capable of achieving some level of high-order accuracy for sufficiently smooth fields (although in some cases, this is not as good as the unmodified DG method) while producing a monotone solution for discontinuous initial conditions. In terms of run-time performance, the OBR and generalized high-order FCT methods excel while the LSD method is substantially more expensive because of its multi-pass nature. Based on these results, we favor the high-order FCT approach for both its run-time performance (which is slightly worse than OBR) and its local nature. While this paper has focused exclusively on single material ALE hydrodynamics, our ultimate interest is multi-material flow. In future work, we intend to apply these same monotonicity approaches to the multi-material case by using high-order finite element-based volume fraction functions.

ACKNOWLEDGEMENTS

The authors would like to thank Denis Ridzal and Pavel Bochev for helpful discussions about the OBR approach and for providing the software for solving the optimization problem (41).
This work performed under the auspices of the US Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, LLNL-JRNL-651254.

REFERENCES

1. Godunov SK. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Matematicheskii Sbornik* 1959; **47**:271–306.
2. LeVeque RJ. High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal on Numerical Analysis* 1996; **33**(2):627–665.
3. Kuzmin D, Lohner R, Turek S. *Flux-corrected Transport* first edition. Principles[10], Algorithms and Applications. Springer Verlag, 2005.
4. Dobrev VA, Kolev TV, Rieben RN. High order curvilinear finite element methods for Lagrangian hydrodynamics. *SIAM Journal on Scientific Computing* 2012; **5**(34):B606–B641.
5. Bochev P, Ridzal D, Peterson K. Optimization-based remap and transport: a divide and conquer strategy for feature-preserving discretizations. *Journal of Computational Physics* 2013; **257**:1113–1139.
6. Rieben RN, White DA, Wallin BK, Solberg JM. An arbitrary Lagrangian–Eulerian discretization of MHD on 3D unstructured grids. *Journal of Computational Physics* 2007; **226**:534–570.
7. Scovazzi G, Lopez Ortega A. *Flux-corrected Transport* second edition, chapter Algebraic flux correction and geometric conservation in arbitrary Lagrangian–Eulerian computations. Principles[10], Algorithms and Applications. Springer Verlag, 2012.
8. Liska R, Shashkov M, Vachal P, Wendroff B. Optimization-based synchronized flux-corrected conservative interpolation (remapping) of mass and momentum for arbitrary Lagrangian–Eulerian methods. *Journal of Computational Physics* 2010; **229**:1467–1497.
9. Kuzmin D, Turek S. High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. *Journal of Computational Physics* 2004; **198**(1):131–158.

[10][Correction added on 27 November, after first online publication: *Principals* has been corrected to *Principles* in references 3 and 7]

DOI: 10.1002/fld

10. LeVeque RJ. *Numerical Methods for Conservation Laws*. Birkhauser-Verlag: Basel, 1990.
11. Kuzmin D. Slope limiting for discontinuous Galerkin approximations with a possibly non-orthogonal Taylor basis. *International Journal for Numerical Methods in Fluids* 2013; **71**(9):1178–1190.
12. Gottlieb S, Shu C, Tadmor E. Strong stability preserving high order time discretization methods. *SIAM Review* 2001; **43**:89–112.
13. Kuzmin D, Turek S. Flux correction tools for finite elements. *Journal of Computational Physics* 2002; **175**(2):525–558.
14. Schar C, Smolarkiewicz P. A synchronous and iterative flux-correction formalism for coupled transport equations. *Journal of Computational Physics* 1996; **128**:101–120.
15. Bochev P, Ridzal D, Scovazzi G, Shashkov M. Constrained-optimization based data transfer. In Kuzmin D, Löhner R, Turek S (eds), *Flux-corrected Transport*. Springer Netherlands: Scientific Computation, 2012; 345–398.
16. Bochev P, Ridzal D, Shashkov M. Fast optimization-based conservative remap of scalar fields through aggregate mass transfer. *Journal of Computational Physics* 2013; **246**(0):37–57.
17. Bochev P, Ridzal D, Scovazzi G, Shashkov M. Formulation, analysis and numerical study of an optimization-based conservative interpolation (remap) of scalar fields for arbitrary Lagrangian–Eulerian methods. *Journal of Computational Physics* 2011; **230**:5199–5225.
18. Shashkov M, Wendroff B. The repair paradigm and application to conservation laws. *Journal of Computational Physics* 2004; **198**:265–277.
19. Galera S, Maire P-H, Breil J. A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction. *Journal of Computational Physics* 2010; **229**(16):5755–5787.

DOI: 10.1002/fld