INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 0000; **00**:1–23
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld

Monotonicity in High-Order Curvilinear Finite Element ALE 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*
[2] *Weapons and Complex Integration, Lawrence Livermore National Laboratory*

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, i.e. not generate any new extrema in the field variables. It is well known that the only linear methods which are guaranteed to be monotonic for such problems are first order accurate; however, much work has been done 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 new methods for enforcing monotonicity targeting high-order Discontinuous Galerkin (DG) methods for advection equations in the context of high-order curvilinear ALE hydrodynamics. Copyright © 0000 John Wiley & Sons, Ltd.

Received . . .

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, i.e. 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 which are guaranteed to be monotonic for such advection problems are first order accurate [1]; however, much work has been done in developing non-linear methods which blend both

[†] This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, LLNL-JRNL-651254

[*] Correspondence to: Weapons and Complex Integration, Lawrence Livermore National Laboratory, 7000 East Ave L-095, Livermore, CA 94550, rieben1@llnl.gov

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 new 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.

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 e.g. 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 both the mass and advection matrices from the remap phase 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."
- **Flux Corrected Transport (FCT)**: This is a high-order generalization of the FCT methods of [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 [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 DISCONTINUOUS GALERKIN FORMULATION

This section provides a very brief overview of the high-order Discontinuous Galerkin (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, e.g. 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) $\tilde{x} \in \widetilde{\Omega}$ is associated with a unique point (particle) $x \in \Omega$, and we are given a continuous transition function $F(\tilde{x}, \tau) : \widetilde{\Omega} \times [0, 1] \mapsto \mathbb{R}^d$, such that

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

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

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

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

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. However, we can imagine more general scenarios where F is e.g. constructed through a higher-order interpolation in pseudo-time based on a sequence of meshes from a mesh relaxation process.

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

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

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

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

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" $\tilde{x} \in \widetilde{\Omega}$ in pseudo-time through $x(\tau) = F(\tilde{x},\tau)$, and introduce the concept of a "pseudo-material" derivative along these particle trajectories:

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

Since the goal of ALE remap is to extend $\rho(\cdot,0)$ to $\rho(\cdot,1)$ without changing it with respect to an Eulerian frame, i.e. such that $\frac{\partial \rho}{\partial \tau} = 0$, 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 equation (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 get

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

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

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

where $\{\psi_i\}_{i=1}^{N_{\mathcal{V}}}$ is a (moving) basis of a finite element space $\mathcal{V}(\tau) \subset L_2(\Omega)$ defined on $\mathcal{T}(\tau)$, $\boldsymbol{\rho}(\tau)$ is an unknown pseudo-time dependent vector of size $N_{\mathcal{V}} = \dim \mathcal{V}(\tau)$ and $\boldsymbol{\psi}$ is a column vector of all the basis functions. The finite element functions in $\mathcal{V}(\tau)$ are discontinuous across the interior faces $f \in \mathcal{F}_i(\tau)$ of the mesh, and therefore to fix notation we introduce the jump $[\![\cdot]\!]$ and average $\{\cdot\}$ operators:

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

and n_f is a fixed normal vector on each interior face f of the mesh. While the expansion (5) and the scheme we develop below is valid for any choices of $\mathcal{T}(\tau)$ and $\mathcal{V}(\tau)$, 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

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

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,

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

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 get 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 in all intermediate finite element spaces $\mathcal{V}(\tau)$, for example a linear field on general second-order (Q_2) quadrilateral meshes.

2.2. 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

$$\rho(x,\tau) = \sum_i \boldsymbol{\rho}_i(\tau)\psi_i(x)\,, \quad v^{x,y,z}(x,\tau) = \sum_j \boldsymbol{v}_j^{x,y,z}(\tau)w_j(x)\,, \quad e(x,\tau) = \sum_k \boldsymbol{e}_k(\tau)\phi_k(x)\,.$$

As before, the spaces $\mathcal{V}_\rho(\tau)$ and $\mathcal{V}_e(\tau)$ are typically discontinuous (though our framework can also handle the continuous case), but we assume that $\mathcal{V}_v(\tau)$ is continuous, 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, since 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

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

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

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

$$\mathbf{M}_{\mathbf{v}}\frac{d\boldsymbol{v}}{d\tau} = \mathbf{K}_{\mathbf{v}}\boldsymbol{v}\,, \tag{13}$$

$$\mathbf{M}_{\mathbf{e}}\frac{d\boldsymbol{e}}{d\tau} = \mathbf{K}_{\mathbf{e}}\boldsymbol{e}\,. \tag{14}$$

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

$$\mathbf{M}_{\mathbf{v}} = \int_\Omega \rho\,\boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}\,, \qquad \mathbf{K}_{\mathbf{v}} = \sum_{T\in\mathcal{T}(\tau)}\int_T \rho\,\boldsymbol{w}(u\cdot\nabla\boldsymbol{w}^{\mathrm{T}})\,,$$

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

Note that the above 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). If the density space is large enough, $\mathcal{V}_e(\tau) \subseteq \mathcal{V}_\rho(\tau)$ and $\mathcal{V}_v(\tau) \subseteq \mathcal{V}_\rho(\tau)$, then we have internal energy and momentum conservation on semi-discrete level. If the density space is even larger, $\mathcal{V}_v(\tau)^2 \subseteq \mathcal{V}_\rho(\tau)$, the above scheme also preserves the kinetic energy.

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 [7, 3] to extend these concepts to our high-order DG advection algorithm.

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

$$\frac{d\boldsymbol{\rho}}{d\tau} = \mathbf{M}^{-1}\mathbf{K}\boldsymbol{\rho}\,.$$

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

$$\frac{d\boldsymbol{\rho}_i}{d\tau} = \sum_{j\neq i}(\mathbf{M}^{-1}\mathbf{K})_{ij}\left(\boldsymbol{\rho}_j - \boldsymbol{\rho}_i\right)\,. \tag{15}$$

Lets assume for now that the matrix $\mathbf{M}$ is diagonal with positive entries so the entries of $\mathbf{M}^{-1}\mathbf{K}$ and $\mathbf{K}$ have the same sparsity and signs. This is the case for example if ρ is piecewise-constant, or if the mass matrix has been lumped, as discussed in the subsections below.

With this assumption, consider the signs of the off-diagonal entries $\mathbf{K}_{ij}$, $j \neq i$. If

$$\mathbf{K}_{ij} \geq 0,$$

then

$$\text{“maxima do not increase”}\ :\ \boldsymbol{\rho}_i = \max_j \boldsymbol{\rho}_j \ \ \Rightarrow\ \ \boldsymbol{\rho}_j - \boldsymbol{\rho}_i \leq 0 \ \ \Rightarrow\ \ \frac{d\boldsymbol{\rho}_i}{d\tau} \leq 0\,,$$

and

$$\text{“minima do not decrease”}\ :\ \boldsymbol{\rho}_i = \min_j \boldsymbol{\rho}_j \ \ \Rightarrow\ \ \boldsymbol{\rho}_j - \boldsymbol{\rho}_i \geq 0 \ \ \Rightarrow\ \ \frac{d\boldsymbol{\rho}_i}{d\tau} \geq 0\,.$$

Since the discrete advection operator $\mathbf{K}$ is sparse, the fact that *local* maxima/minima cannot increase/decrease implies that ρ will stay between given initial bounds (for example ρ will remain positive). This is a form of monotonicity know as the Local Extremum Diminishing (LED) property. The work of [7, 3] show that LED is the multi-dimensional, unstructured generalization of the Total Variation Diminishing (TVD) property [8].

Much of the work in developing monotone / LED advection methods has focused on low order finite elements, although there are some exceptions [9]. 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, since having a positive vector of degrees of freedom $\boldsymbol{\rho}$ 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 $\boldsymbol{\rho}$ is.

For discretization in pseudo-time of the semi-discrete equations (15), we will consider explicit multistage Runge-Kutta methods, among which the class of strong stability preserving (SSP) methods [10] 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 $\boldsymbol{\rho} = \boldsymbol{\rho}(\tau)$ to $\boldsymbol{\rho}^{new} \approx \boldsymbol{\rho}(\tau + \Delta\tau)$. Let us consider the forward Euler method:

$$\boldsymbol{\rho}_i^{new} = \boldsymbol{\rho}_i + \Delta\tau \sum_{j\neq i}(\mathbf{M}^{-1}\mathbf{K})_{ij}\left(\boldsymbol{\rho}_j - \boldsymbol{\rho}_i\right)\,.$$

This example highlights another motivation for the requirement that the off-diagonal entries in $\mathbf{K}$ are non-negative: with this assumption (and the simplifying condition that $\mathbf{M}$ is diagonal) we can guarantee that there is a small enough time-step in forward Euler, such that the value of $\boldsymbol{\rho}^{new}$ at each point is a convex combination of the values of $\boldsymbol{\rho}$ in its neighbors.

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:

$$\boldsymbol{\rho}_i^{min} \le \boldsymbol{\rho}_i^{new} \le \boldsymbol{\rho}_i^{max}\,, \qquad \text{where} \quad \boldsymbol{\rho}_i^{min} = \min_{j \in N_i} \boldsymbol{\rho}_j \quad \text{and} \quad \boldsymbol{\rho}_i^{max} = \max_{j \in N_i} \boldsymbol{\rho}_j\,. \tag{16}$$

Our default option is to define N_i based on the sparsity pattern of the ith row of $\mathbf{K}$, motivated by the sum in (15) for a diagonal $\mathbf{M}$. Due to the face integrals in our DG scheme, this means that N_i consists of all degrees of freedom in elements that share a face with the element containing i. More restrictive choices, which e.g. 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 the Godunov theorem, the modification can not 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. Locally Scaled Diffusion

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: (1) we modify (10) to obtain a low-order monotone linear scheme; (2) 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. [7]:

$$\mathbf{M}^* \frac{d\boldsymbol{\rho}}{d\tau} = \mathbf{K}^* \boldsymbol{\rho}\,. \tag{17}$$

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

$$\mathbf{M}^* = \mathbf{M} + \mathbf{L}\,, \qquad \mathbf{K}^* = \mathbf{K} + \mathbf{D}\,, \tag{18}$$

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

$$\mathbf{L}_{ij} = -\mathbf{M}_{ij}\,, \qquad \mathbf{D}_{ij} = \max\{0, -\mathbf{K}_{ij}, -\mathbf{K}_{ji}\}\,, \tag{19}$$

and diagonal entries which zero-out the row-sums

$$\mathbf{L}_{ii} = -\sum_{j \ne i} \mathbf{L}_{ij}\,, \qquad \mathbf{D}_{ii} = -\sum_{j \ne i} \mathbf{D}_{ij}\,. \tag{20}$$

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, sparse, and have non-positive off-diagonal entries and zero row-sums. The use of the positive Bernstein basis is essential here since it guarantees that $\mathbf{L}_{ij} = -\mathbf{M}_{ij} \le 0$. The function of the matrix $\mathbf{D}$ is to render $\mathbf{K}^*$ local extremum diminishing by eliminating non-positive off-diagonal entries of the advection matrix $\mathbf{K}$ in a conservative manner. Indeed, mass conservation for (17) follows from the identities $\mathbf{L}^{\mathrm{T}}\mathbf{1} = \mathbf{D}^{\mathrm{T}}\mathbf{1} = \mathbf{0}$ which combined with $\frac{d\mathbf{M}}{d\tau} = \mathbf{A} - \mathbf{K}$, and $\mathbf{A}^{\mathrm{T}}\mathbf{1} = \mathbf{0}$ imply

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

While this form of simple upwinding is an effective way to ensure monotonicity, it is far too diffusive (first order accurate at best) 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 degrees of freedom and denoted as

$\mu(\boldsymbol{\rho})$, is such that $\mu_i(\boldsymbol{\rho}) \in [0,1]$, should vanish ($\mu = 0$) where the monotonicity bounds (16) are not violated, and only be active ($\mu = 1$) where (16) does not hold. A straightforward way to define $\mu(\boldsymbol{\rho})$ is to perform a high order, linear advection step based on (10), and then based on the high-order solution $\boldsymbol{\rho}$ compute

$$\mu_i(\boldsymbol{\rho}) = \begin{cases} \beta(\boldsymbol{\rho}_i - \boldsymbol{\rho}_i^{max}), & \text{if } \boldsymbol{\rho}_i > \boldsymbol{\rho}_i^{max} \\ \beta(\boldsymbol{\rho}_i^{min} - \boldsymbol{\rho}_i), & \text{if } \boldsymbol{\rho}_i < \boldsymbol{\rho}_i^{min} \\ 0 & \text{otherwise.} \end{cases} \tag{21}$$

The function $\beta : \mathbb{R}^+ \mapsto [0,1]$ can be defined in many ways; 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.

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

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

Note that if $\mu_i = 1$, the off-diagonal entries in the ith 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 degrees of freedom where LED is violated, the above local modification will fix them to be monotonic. This motivates the following iterative process, which we refer to as Locally Scaled Diffusion (LSD):

0. Input: $\boldsymbol{\rho} \approx \boldsymbol{\rho}(\tau)$; output: $\boldsymbol{\rho}^{new} \approx \boldsymbol{\rho}(\tau + \Delta\tau)$.
1. Compute the monotonicity bounds $\boldsymbol{\rho}^{min}$ and $\boldsymbol{\rho}^{max}$ from (16).
2. Set $\mu = 0$.
3. Compute $\boldsymbol{\rho}^{new}$ from $\boldsymbol{\rho}$ 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: $\mu^e = \mu^e(\boldsymbol{\rho}^{new})$ where μ^e is defined by (21) using $\beta(t) = t$.
5. Check for convergence: if $\|\mu^e\|_\infty < \epsilon$ then exit the LSD iteration.
6. Compute the new monotonicity measure: $\mu^{new} = \mu(\boldsymbol{\rho}^{new})$.
7. Set $\mu_i = \max\{\mu_i, \mu_i^{new}\}$ and go to step 3.

Clearly the above algorithm will do nothing if the high-order solution is already monotonic, i.e. $\mu(\boldsymbol{\rho}^{new}) = 0$, 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, i.e. if $\mu_i^{new} \leq \mu_i$, $\forall i$. In practice, we also limit the number of LSD iteration to 20, however, in our experience, the algorithm generally converges much faster, in 3–4 iterations. The numerical tests presented in Section 4 use the sharp cutoff function $\beta(t) = \min\{1, 10^{12}t\}$ in (21) for step 6, and the stopping tolerance $\epsilon = 10^{-10}$.

3.2. Flux Corrected Transport

A method which guarantees monotonicity without an outer iteration can be constructed using the ideas of flux corrected transport (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 nonlinear corrections to the flux which improve accuracy while maintaining monotonicity (in contrast to LSD, which starts from the high-order solution and makes nonlinear 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 (18), the equation

$$\mathbf{M}^* \frac{d\boldsymbol{\rho}}{d\tau} = \mathbf{K}^* \boldsymbol{\rho} + (\mathbf{M}^* - \mathbf{M}) \frac{d\boldsymbol{\rho}}{d\tau} - \mathbf{D}\boldsymbol{\rho} \tag{22}$$

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- and high-order equations in the lumped mass matrix and in the addition of explicit diffusion.

Following Kuzmin [11], 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.

Since $\mathbf{D}$ has zero row-sums, we can write:

$$(\mathbf{D}\boldsymbol{\rho})_i = \sum_{j \neq i} \mathbf{D}_{ij} (\boldsymbol{\rho}_j - \boldsymbol{\rho}_i). \tag{23}$$

Since $\mathbf{L} = \mathbf{M}^* - \mathbf{M}$ has zero row-sums, see (19) and (20), we can also write

$$\left(\mathbf{M}^* \frac{d\boldsymbol{\rho}}{d\tau} - \mathbf{M} \frac{d\boldsymbol{\rho}}{d\tau} \right)_i = \sum_{j \neq i} \mathbf{L}_{ij} \left(\frac{d\boldsymbol{\rho}_j}{d\tau} - \frac{d\boldsymbol{\rho}_i}{d\tau} \right) = \sum_{j \neq i} \mathbf{M}_{ij} \left(\frac{d\boldsymbol{\rho}_i}{d\tau} - \frac{d\boldsymbol{\rho}_j}{d\tau} \right). \tag{24}$$

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

$$\boldsymbol{f}_{ij} = \left(\mathbf{M}_{ij} \frac{d}{d\tau} + \mathbf{D}_{ij} \right) (\boldsymbol{\rho}_i - \boldsymbol{\rho}_j). \tag{25}$$

Since it is the anti-symmetry $\boldsymbol{f}_{ij} = -\boldsymbol{f}_{ji}$ that ensures conservation, any symmetric scaling of the fluxes

$$\boldsymbol{f}^*_{ij} = \alpha_{ij} \boldsymbol{f}_{ij} \,, \qquad \text{where} \qquad \alpha_{ij} = \alpha_{ji} \tag{26}$$

is also conservative. Our modified equation is then

$$\mathbf{M}^* \frac{d\boldsymbol{\rho}}{d\tau} = \mathbf{K}^* \boldsymbol{\rho} + \boldsymbol{f}^* \tag{27}$$

where

$$\boldsymbol{f}^* = \sum_{j \neq i} \alpha_{ij} \boldsymbol{f}_{ij} \tag{28}$$

are the limited anti-diffusive corrections. When $\alpha_{ij} \equiv 1$, the above equations are just a different form of (10), while for $\alpha_{ij} \equiv 0$ they reduce to the low-order approximation (17). The fact that intermediate values for α_{ij} could be picked to interpolate from the latter 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:

$$\mathbf{M} \Delta \boldsymbol{\rho}^H = \Delta\tau \mathbf{K} \boldsymbol{\rho} \,, \tag{29}$$

$$\mathbf{M}^* \Delta \boldsymbol{\rho}^L = \Delta\tau \mathbf{K}^* \boldsymbol{\rho} \,. \tag{30}$$

Note that (29) can be expanded into a form analogous to (22):

$$\mathbf{M}^* \Delta \boldsymbol{\rho}^H = \mathbf{M}^* \Delta \boldsymbol{\rho}^L + (\mathbf{M}^* - \mathbf{M}) \Delta \boldsymbol{\rho}^H - \Delta\tau \mathbf{D} \boldsymbol{\rho}. \tag{31}$$

DOI: 10.1002/fld

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

$$\mathbf{M}^*\Delta\boldsymbol{\rho}^H = \mathbf{M}^*\Delta\boldsymbol{\rho}^L + \sum_{j\neq i}\mathbf{M}_{ij}(\Delta\boldsymbol{\rho}_i^H - \Delta\boldsymbol{\rho}_j^H) + \Delta\tau\sum_{j\neq i}\mathbf{D}_{ij}\left(\boldsymbol{\rho}_i - \boldsymbol{\rho}_j\right). \tag{32}$$

or

$$\mathbf{M}^*\Delta\boldsymbol{\rho}^H = \mathbf{M}^*\Delta\boldsymbol{\rho}^L + \sum_{j\neq i}\mathbf{f}_{ij}. \tag{33}$$

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:

$$\mathbf{M}^*\Delta\boldsymbol{\rho} = \mathbf{M}^*\Delta\boldsymbol{\rho}^L + \sum_{j\neq i}\alpha_{ij}\mathbf{f}_{ij}, \tag{34}$$

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 due to antidiffusive corrections:

$$\Delta\boldsymbol{\rho}_i^{A+,\max} = \boldsymbol{\rho}_i^{\max} - \boldsymbol{\rho}_i^L,\ \ \Delta\boldsymbol{\rho}_i^{A-,\max} = \boldsymbol{\rho}_i^L - \boldsymbol{\rho}_i^{\min}\,, \tag{35}$$

where $\boldsymbol{\rho}_i^L$ is the solution updated by the low-order approximation. Finding a set of α_{ij} which prevents bounds violations, but are otherwise as close to unity as possible is a globally coupled problem. However, we can obtain an admissible set of α_{ij} which 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:

$$\Delta\boldsymbol{\rho}^{A\pm} = \sum_{j\neq i}\mathbf{f}_{ij}^{\pm}\,. \tag{36}$$

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

$$\alpha_i^{\pm} = \min\left(1, \frac{\Delta\boldsymbol{\rho}^{A\pm,\max}}{\Delta\boldsymbol{\rho}^{A\pm}}\right)\,. \tag{37}$$

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:

$$\alpha_{ij} = \begin{cases}\min\left(\alpha_i^+, \alpha_j^-\right) & \mathbf{f}_{ij} > 0\\ \min\left(\alpha_i^-, \alpha_j^+\right) & \mathbf{f}_{ij} < 0.\end{cases} \tag{38}$$

Since $\mathbf{f}_{ij} = -\mathbf{f}_{ji}$, $\alpha_{ij} = \alpha_{ji}$.

Using the new monotone update $\Delta\boldsymbol{\rho}$ in place of the low-order update $\Delta\boldsymbol{\rho}^L$, we can now repeat the FCT process. The new high-order anti-diffusive fluxes are given by $\mathbf{f}_{ij} \leftarrow (1-\alpha_{ij})\mathbf{f}_{ij}$, i.e. the remainder of the fluxes that was not used as a correction to define $\Delta\boldsymbol{\rho}$ from $\Delta\boldsymbol{\rho}^L$. In other words, instead of (33), the starting point now is

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

This process can be repeated multiple times and is sometimes referred to as *iterated* FCT.

In order to use the FCT algorithm in a high-order multistage Runge-Kutta (RK) method, we replace the ODE right-hand side $\mathbf{M}^{-1}\mathbf{K}\boldsymbol{\rho} = \Delta\boldsymbol{\rho}^H/\Delta\tau$ with the term $\Delta\boldsymbol{\rho}/\Delta\tau$ at every stage of the RK method. In addition, at every stage, new bounds $\boldsymbol{\rho}^{min}$ and $\boldsymbol{\rho}^{max}$ are computed. The value of $\Delta\tau$ is important for the FCT algorithm because smaller $\Delta\tau$ will generally produce a slope vector $\Delta\boldsymbol{\rho}/\Delta\tau$ that is closer to $\Delta\boldsymbol{\rho}^H/\Delta\tau$. In order to ensure monotonicity with strong stability preserving (SSP) methods, we use the same $\Delta\tau$ in the FCT algorithm as the full RK pseudo-time step.

3.3. Optimization Based Remap

Our third method for ensuring monotonicity consists of two main steps: (1) given a vector $\boldsymbol{\rho} \approx \boldsymbol{\rho}(\tau)$ compute a high-order approximation $\boldsymbol{\rho}^H \approx \boldsymbol{\rho}(\tau + \Delta\tau)$ to problem (10) using, for example, an explicit multistage Runge-Kutta method; (2) define the new density vector $\boldsymbol{\rho}^{new} \approx \boldsymbol{\rho}(\tau + \Delta\tau)$ by solving an optimization problem that ensures the following properties for $\boldsymbol{\rho}^{new}$: it is close to $\boldsymbol{\rho}^H$, has the same mass as $\boldsymbol{\rho}^H$, and satisfies the monotonicity bounds (16). Specifically, in step 2, we use the mass-variable mass-target (MVMT) version of the optimization-based remap (OBR) algorithm, see [12, 13, 5]. The MVMT-OBR problem reads

$$\|\boldsymbol{\rho}^{new} - \boldsymbol{\rho}^H\|_{\ell_2} \mapsto \min\,, \qquad \mathbf{m}^{\mathrm{T}}\boldsymbol{\rho}^{new} = \mathbf{m}^{\mathrm{T}}\boldsymbol{\rho}^H\,, \qquad \boldsymbol{\rho}_i^{min} \le \boldsymbol{\rho}_i^{new} \le \boldsymbol{\rho}_i^{max} \quad \forall i\,, \tag{39}$$

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, $\|\mathbf{x}\|^2_{\ell_2} = \mathbf{x}^{\mathrm{T}}\mathbf{x} = \sum_i \mathbf{x}_i^2$; and $\boldsymbol{\rho}^{min}$, $\boldsymbol{\rho}^{max}$ are the vectors of local lower and upper bounds in (16). This problem can be solved efficiently [13], leading to a fast one step method that is both conservative and LED monotonic.

The OBR approach is very general, but 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 (39) may change $\boldsymbol{\rho}^H$ 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 $\boldsymbol{\rho}^{new}$ and $\boldsymbol{\rho}^H$, and while other, more localized, choices of norms are possible, e.g. ℓ_1-norm, they lead to other issues such as non-uniqueness and increased difficulty in the solution of the optimization problem.

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.

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 degrees 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, 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 1.

DOI: 10.1002/fld

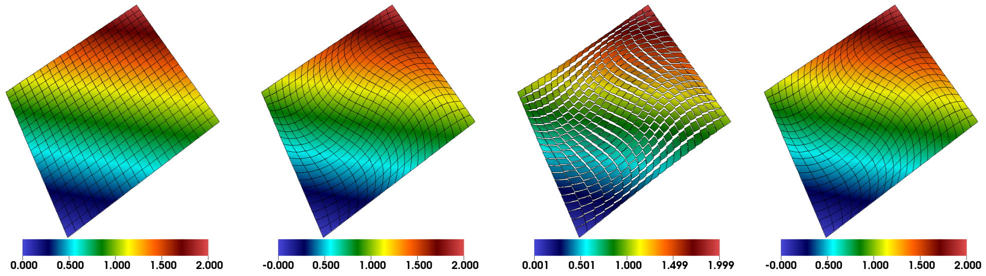
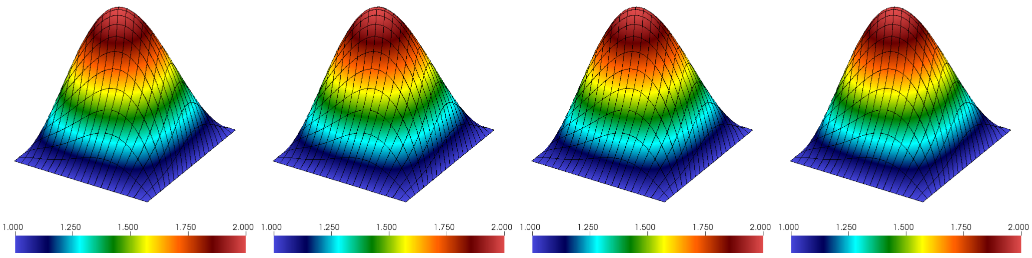


Figure 1. Results for remap of linear density field (*left to right*): Initial mesh and density, high order result, fully upwinded result, locally scaled diffusion result (representative of all monotone schemes).

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 2. 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 are shown in Figure 3. 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.



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

Figure 2. Results for remap of smooth field $\rho = 1 + \sin(\pi x)\sin(\pi y)$.

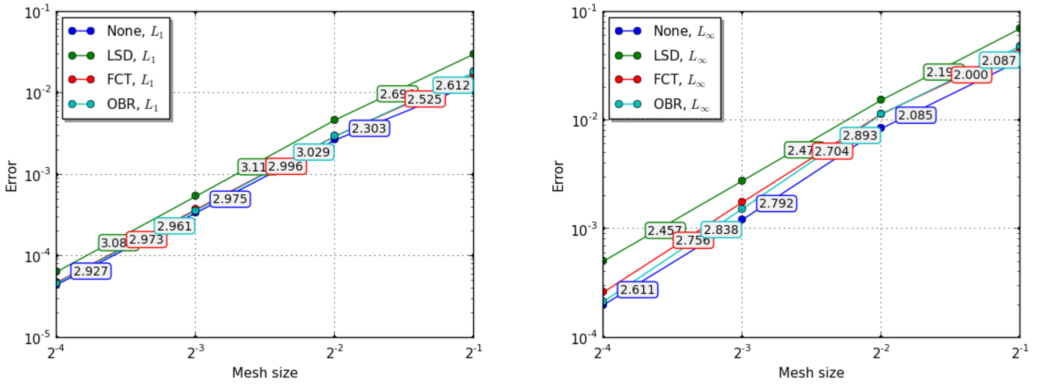
Next we examine behavior of three models of a transition between two states, of increasing discontinuity. These can be thought of as simple models for a shock profile or a contact discontinuity, that may be smeared by an artificial viscosity or by monotone projection of a material interface not aligned 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. The convergence behavior is shown in Figure 5.

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 Figure 6 and Figure 7. A coarser grid was used than in other test problems to make differences in the solution more apparent.

A test problem which exercises the monotonicity strategies is the remap of a step function, using the same prescribed rotational motion. The results are shown in Figure 8. The unmodified DG method exhibits undershoots and overshoots as expected, but all three of the monotone

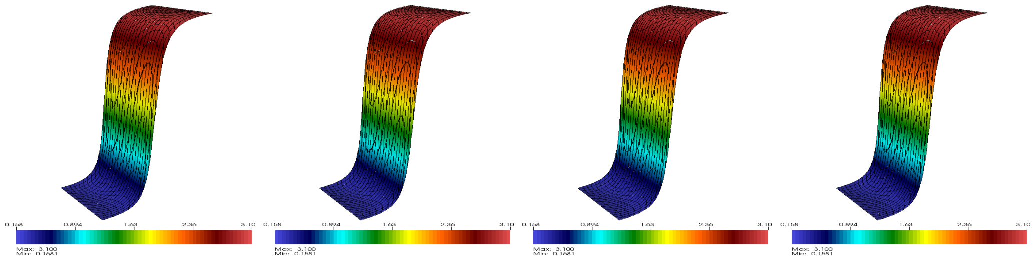
DOI: 10.1002/fld



(a) L_1

(b) L_∞

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



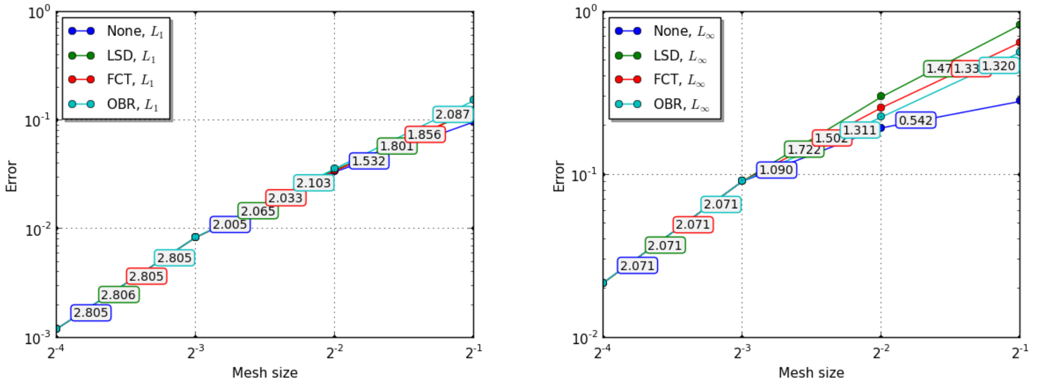
(a) Unmodified DG

(b) LSD

(c) FCT

(d) OBR

Figure 4. Results for remap of arctan field.



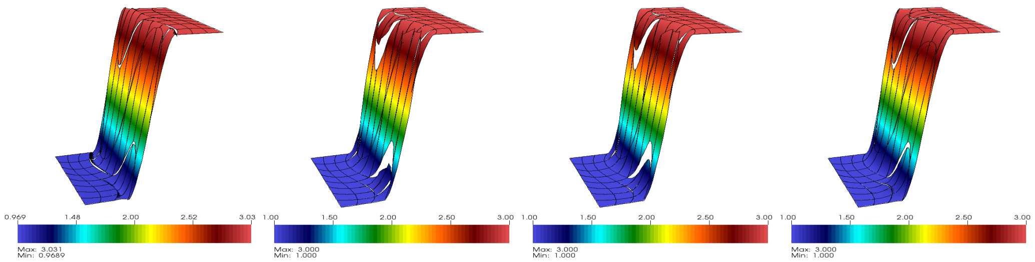
(a) L_1

(b) L_∞

Figure 5. Convergence behavior of remap strategies of arctan field.

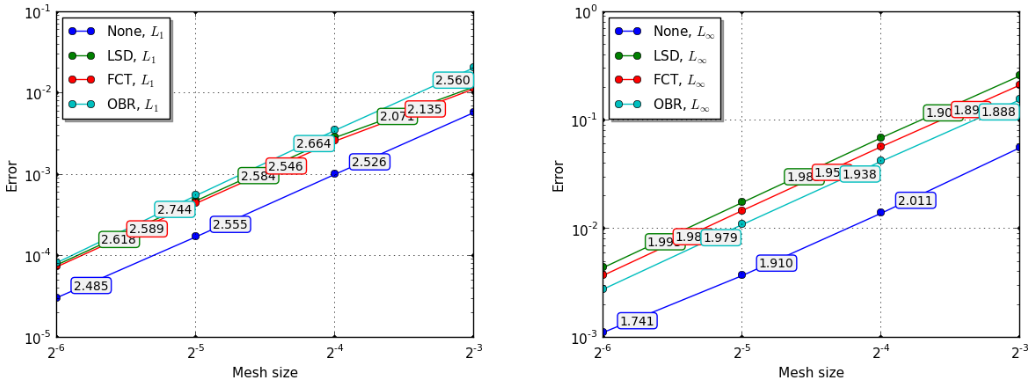
schemes preserve the minimum and maximum values in the initial condition. While there are some qualitative differences in the solution, none of the methods stands out as clearly superior by inspection. The convergence behavior of the methods is shown in Figure 9. All methods have similar convergence rates in L_1, while the OBR method seems to have slightly larger error magnitudes at any given resolution.

DOI: 10.1002/fld



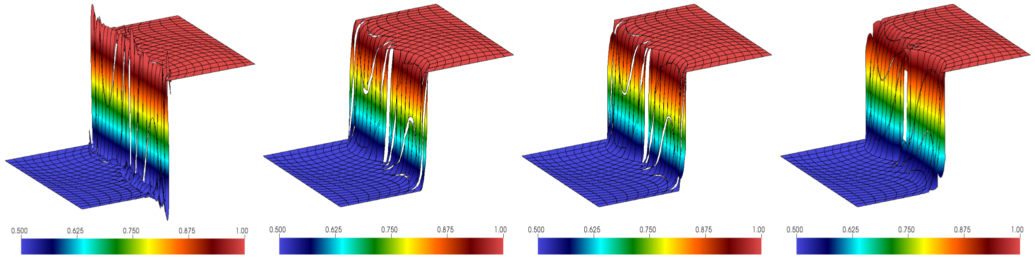
(a) Unmodified DG (b) LSD (c) FCT (d) OBR

Figure 6. Results for remap of a piecewise quadratic field.



(a) L_1 (b) L_∞

Figure 7. Convergence behavior of remap strategies of piecewise quadratic field.



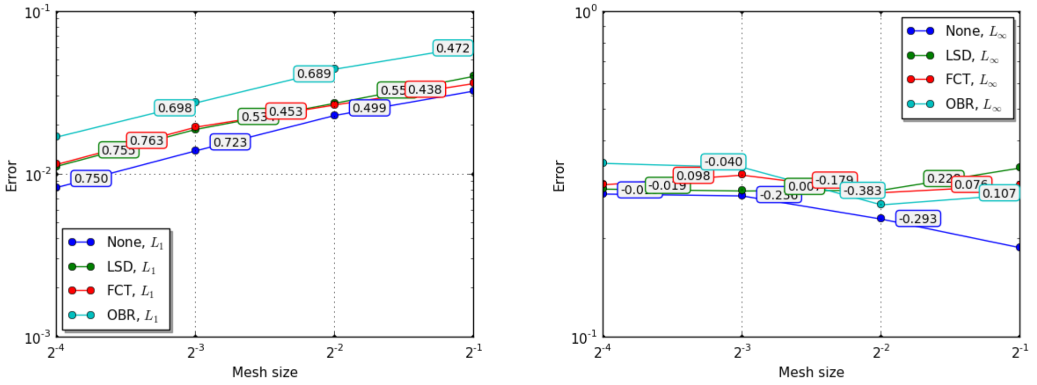
(a) Unmodified DG (b) LSD (c) FCT (d) OBR

Figure 8. Results for remap of a step function.

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 (see Figure 10) as initial data:

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



(a) L_1

(b) L_∞

Figure 9. Convergence behavior of remap strategies on step function.

where we set

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

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.

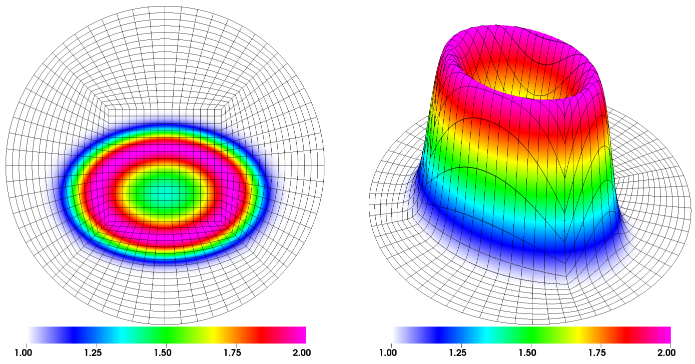


Figure 10. Initial smooth field: view from the top (left) and a 3D view (right).

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 degrees in counter-clockwise 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 disc 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 strong stability preserving (SSP) Runge-Kutta method with three stages, [10], in order to match the third order spatial discretization.

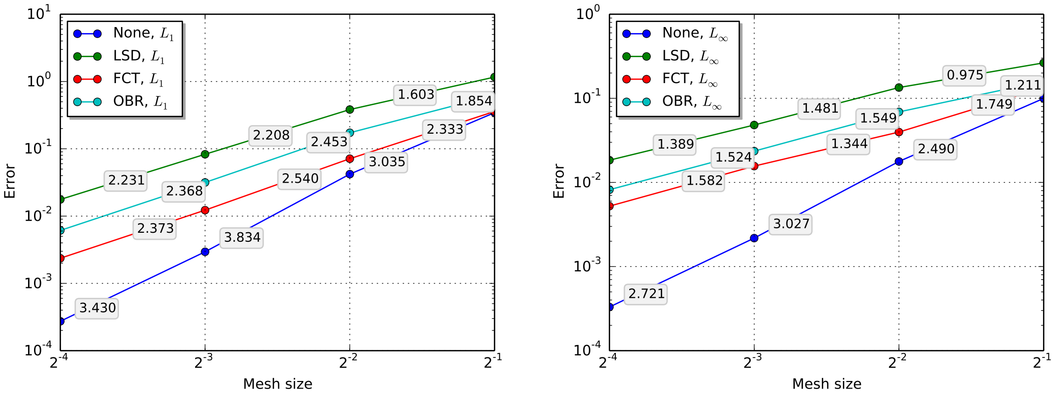


Figure 11. Convergence plots for the smooth field rotation test; L_1 (left) and L_∞ (right) norms. The number indicate the convergence rates.

In Figure 11, 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 reason for this reduction, as illustrated in Figure 12, 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 13: 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) due to 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, 7] where three shapes are projected onto a mesh as an initial condition, then subject to a 360 degree rotation of the mesh. In this case, we consider an unstructured disc 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

$$\rho(x, y) = cone(x, y, \vec{r}_1) + hump(x, y, \vec{r}_2) + cyl(x, y, \vec{r}_3) + 1, \tag{40}$$

where

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

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

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

DOI: 10.1002/fld

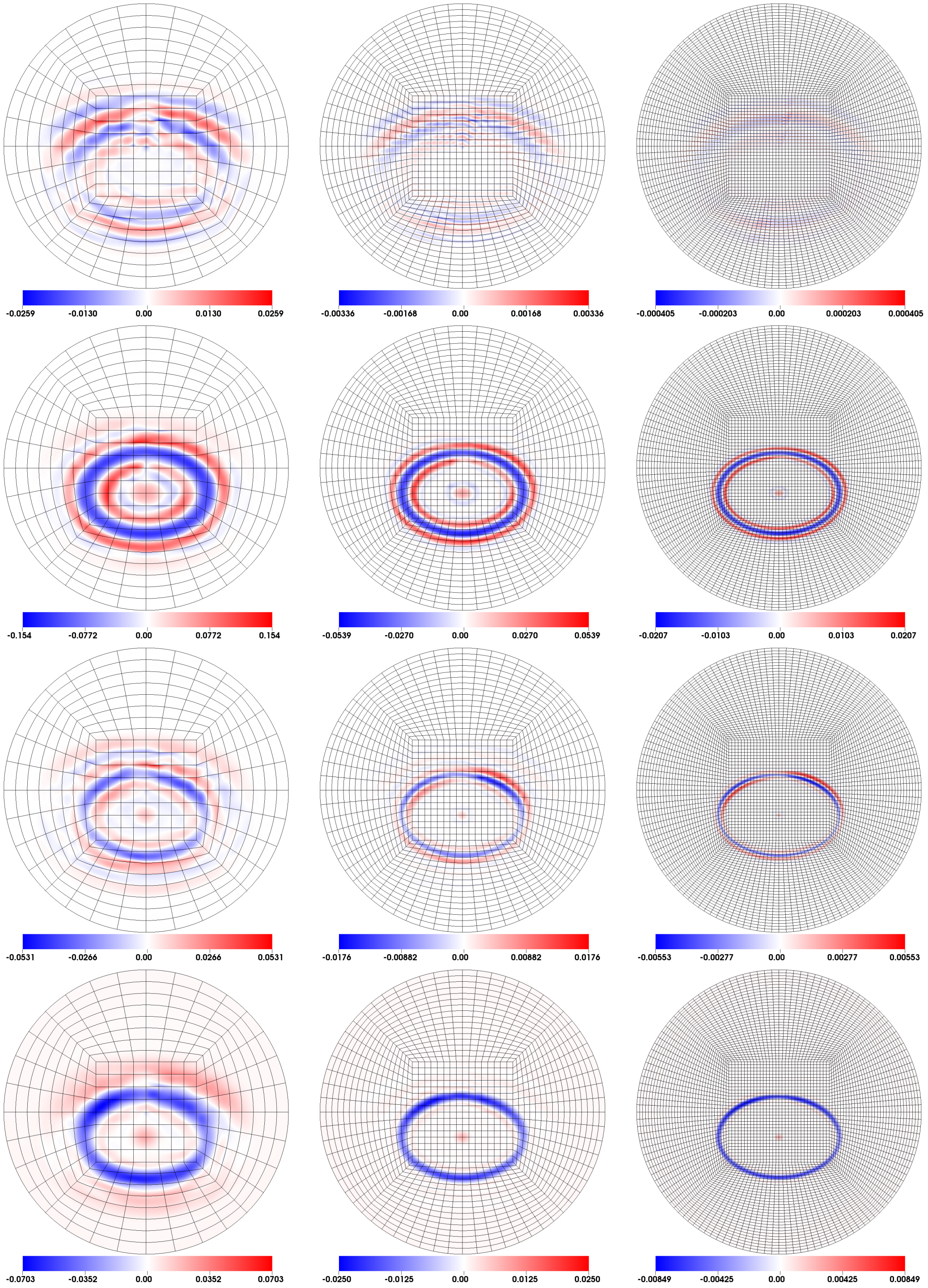


Figure 12. Plots of the error after one full 360 degree rotation; left to right: three levels of uniform mesh refinement; top to bottom: monotonicity algorithm: none, LSD, FCT, and OBR, respectively. Each plot uses the exact range of values (symmetric around zero) spanned by the error function.

and the distance function is defined as

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

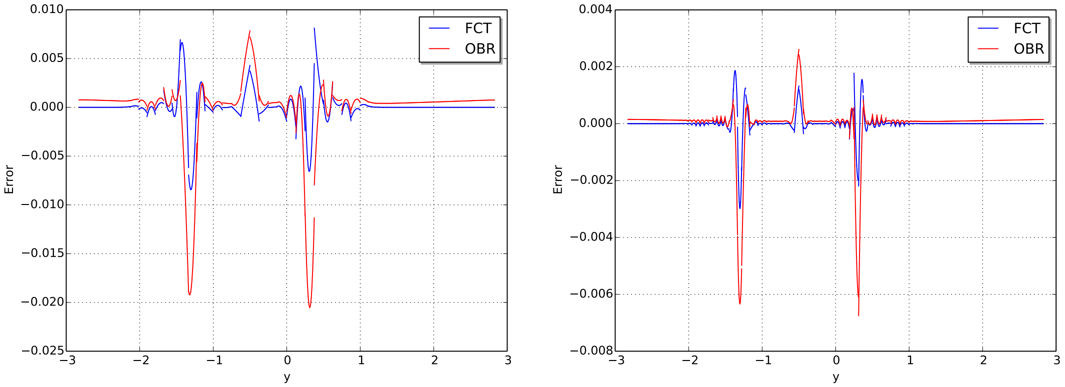


Figure 13. Comparison of the errors for the FCT and OBR monotonicity algorithms, cross section at $x = 10^{-3}$ for mesh sizes $h = 1/8$ (left) and $h = 1/16$ (right). Notice that the error for OBR is shifted away from zero in the regions away from the maximum and minimum.

We consider two cases for the initial density function defined via (40), each with the same total number of degrees of freedom (dof):

1. A fine mesh using Q_2 discontinuous elements (9 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 14. We advect the density field for a full 360 degree 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 2 pseudo-time steps per remap step. For the FCT method, we use 2 iterations for the Q_2 case and 4 iterations for the Q_4 case.

The results for the Q_2 and Q_4 cases are shown in Figure 15 and Figure 16 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 Figure 15 and Figure 16, 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 3× the cost of OBR and FCT) in terms of run time due to 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. Arbitrary Lagrangian-Eulerian 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 [14]). 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

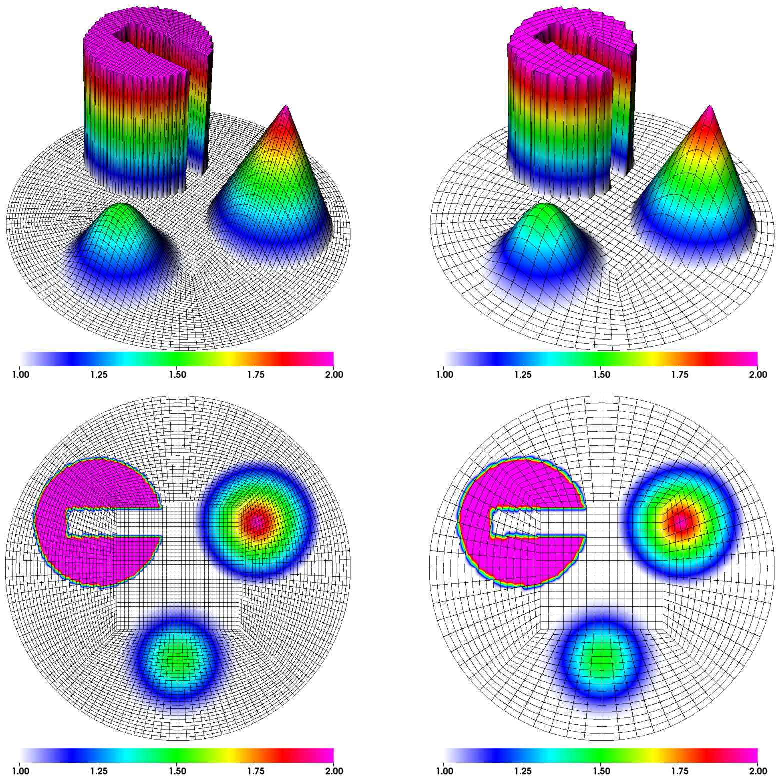


Figure 14. 3D view (*top*) and 2D view (*bottom*) of initial mesh and density using Q_2 elements on a fine mesh (*left*) and Q_4 elements on a coarse mesh (*right*)

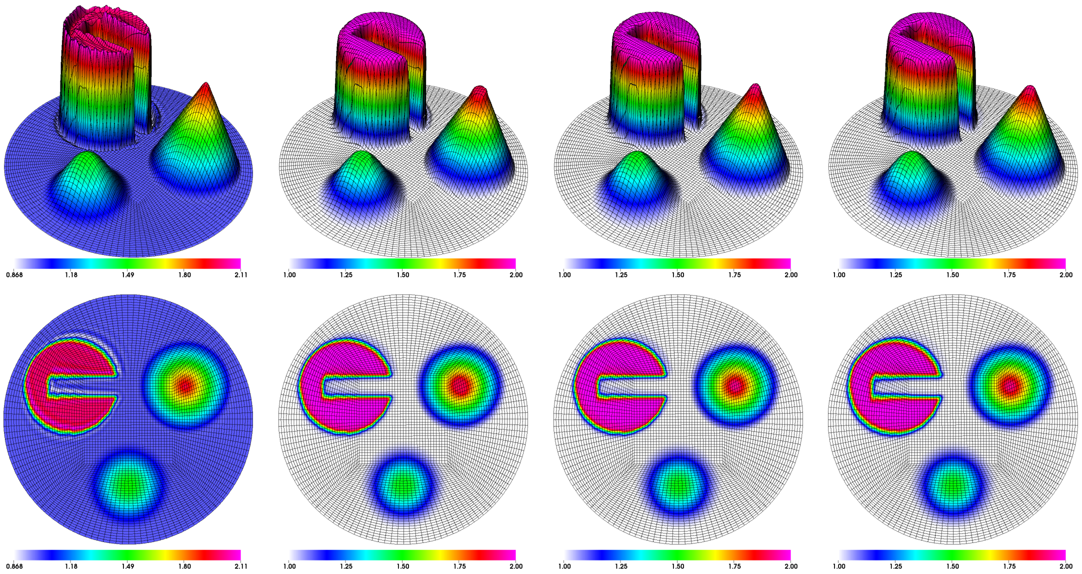


Figure 15. 3D view (*top*) and 2D view (*bottom*) of results for solid body rotation test on unstructured mesh using Q_2 elements (*left to right*): high order result, LSD result, FCT result and OBR result.

conserving RK2 dynamic time stepping algorithm for the Lagrangian phase of the ALE

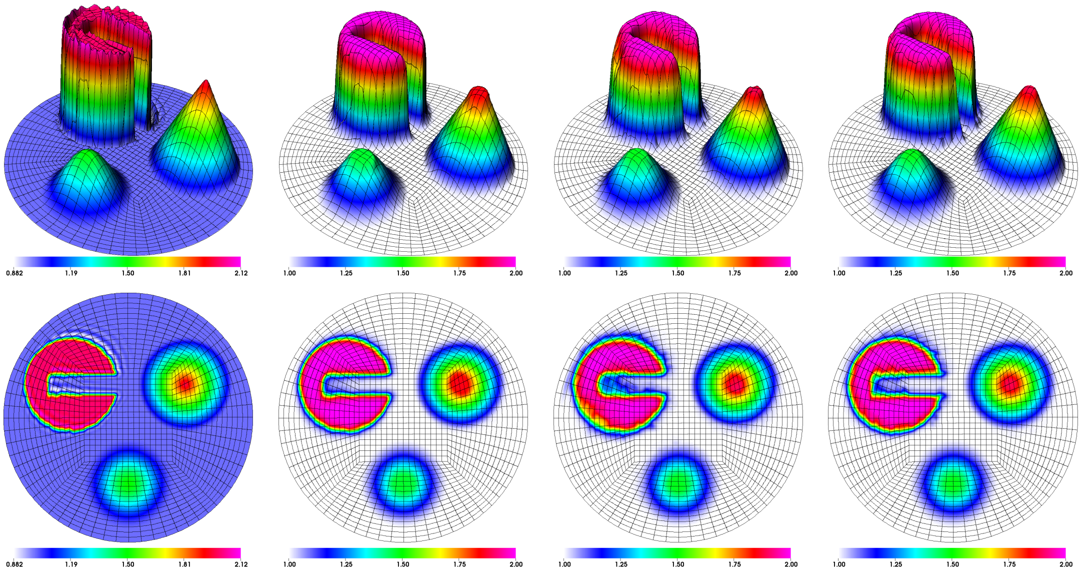


Figure 16. 3D view (*top*) and 2D view (*bottom*) of results for solid body rotation test on unstructured mesh using Q_4 elements (*left to right*): high order result, LSD result, FCT result and OBR result.

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

calculation. Every 20 Lagrange cycles, we apply 2 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 remap, we apply our three different monotone methods (and also consider the unmodified versions 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 17 we show 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 yield 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 in fewer overall cycles to complete the calculation, resulting in an overall speedup of the calculation for FCT and OBR. The 3× 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.

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	Num. 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

We have presented three new approaches for ensuring a monotone solution for advection equations using high-order DG finite element methods. 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-Lagrange 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 (though 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 FCT methods excel while the LSD method is substantially more expensive due to its multi-pass nature. Based on these results, we favor the FCT approach for both its run-time performance (which is slightly worse than OBR) and its local nature (unlike OBR which can lead to non-local changes).

ACKNOWLEDGMENTS

The authors would like to thank Denis Ridzal and Pavel Bochev for helpful discussions about the optimization based remap approach, and for providing the software for solving the optimization problem (39).

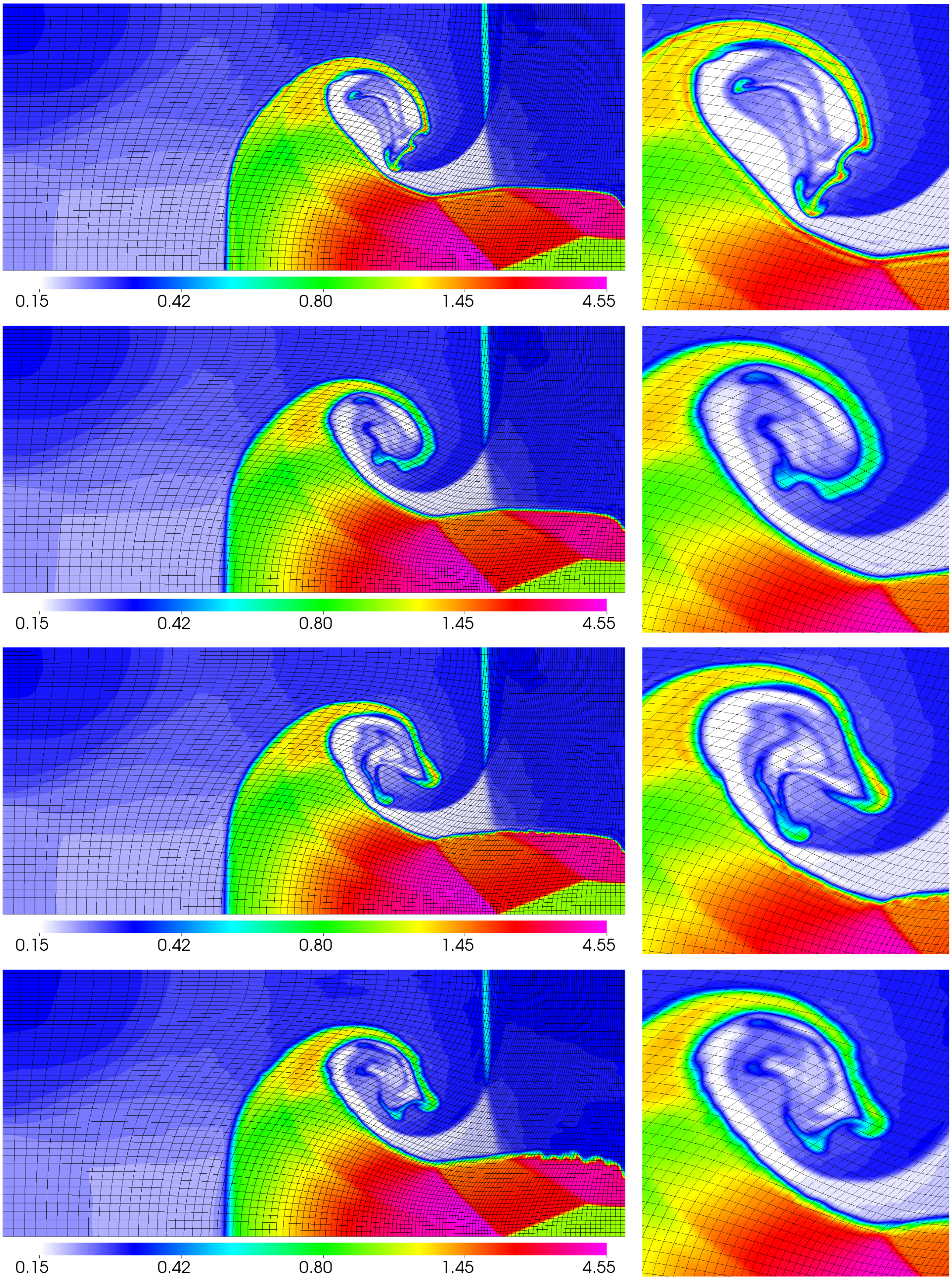


Figure 17. Results for ALE shock triple-point problem using Q_4 elements (*top to bottom*): high order result, LSD result, FCT result and OBR result.

REFERENCES

1. S. K. Godunov. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sbornik*, 47:271–306, 1959. 1
2. R. J. LeVeque. High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Num. Anal.*, 33(2):627–665, 1996. 2, 16
3. D. Kuzmin, R. Lohner, and S. Turek. *Flux-Corrected Transport.* Principals, Algorithms and Applications. Springer Verlag, first edition, 2005. 2, 5, 6
4. V. A. Dobrev, Tz. V. Kolev, and R. N. Rieben. High order curvilinear finite element methods for Lagrangian hydrodynamics. *SIAM J. Sci. Comp.*, 5(34):B606–B641, 2012. 2, 3, 4
5. P. Bochev, D. Ridzal, and K. Peterson. Optimization-based remap and transport: A divide and conquer strategy for feature-preserving discretizations. *J. Comput. Phys.*, 257:1113–1139, 2013. 2, 11
6. R. N. Rieben, D. A. White, B. K. Wallin, and J. M. Solberg. An arbitrary Lagrangian-Eulerian discretization of MHD on 3D unstructured grids. *J. Comput. Phys.*, 226:534–570, 2007. 3
7. D. Kuzmin and S. Turek. High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. *J. Comput. Phys.*, 198(1):131–158, 2004. 5, 6, 7, 16
8. R. J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhauser-Verlag, 1990. 6
9. D. Kuzmin. Slope limiting for discontinuous galerkin approximations with a possibly non-orthogonal taylor basis. *Internat. J. Numer. Methods Fluids*, 71(9):1178–1190, 2013. 6
10. S. Gottlieb, C. Shu, and E. Tadmor. Strong stability preserving high order time discretization methods. *SIAM Rev.*, 43:89–112, 2001. 6, 15
11. D. Kuzmin and S. Turek. Flux correction tools for finite elements. *J. Comput. Phys.*, 175(2):525–558, 2002. 9
12. Pavel Bochev, Denis Ridzal, Guglielmo Scovazzi, and Mikhail Shashkov. Constrained-optimization based data transfer. In Dmitri Kuzmin, Rainald Lhner, and Stefan Turek, editors, *Flux-Corrected Transport*, Scientific Computation, pages 345–398. Springer Netherlands, 2012. 11
13. Pavel Bochev, Denis Ridzal, and Mikhail Shashkov. Fast optimization-based conservative remap of scalar fields through aggregate mass transfer. *Journal of Computational Physics*, 246(0):37 – 57, 2013. 11
14. S. Galera, P-H. Maire, and J. Breil. A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction. *J. Comput. Phys.*, 229(16):5755–5787, 2010. 18

DOI: 10.1002/fld