

High-Order Local Maximum Principle Preserving (MPP) Discontinuous Galerkin Finite Element Method for the Transport Equation

R. Anderson, V. Dobrev, Tz. Kolev, D. Kuzmin, M. Quezada de Luna, R. Rieben and V. Tomov

Abstract

In this work we present a FCT-*like* Maximum-Principle Preserving (MPP) method to solve the transport equation. We use high-order polynomial spaces; in particular, we consider up to 5th order spaces in two and three dimensions and 23rd order spaces in one dimension. The method combines the concepts of positive basis functions for discontinuous Galerkin finite element spatial discretization, locally defined solution bounds, element-based flux correction, and non-linear local mass redistribution. We consider a simple 1D problem with non-smooth initial data to explain and understand the behavior of different parts of the method. Convergence tests in space indicate that high-order accuracy is achieved. Numerical results from several benchmarks in two and three dimensions are also reported.

1 Introduction

We are interested in Maximum-Principle Preserving (MPP) high-order finite element discontinuous Galerkin (DG) discretizations of the transport equation. It is known that the high polynomial degree nature of such discretizations is prone to monotonicity violations. While small oscillations could be accepted in some cases, in many applications they can lead to unphysical values, e.g., values outside of $[0, 1]$ should not occur for material volume fractions in the context of multi-material simulations. Since it is impossible to achieve both monotonicity and high-order accuracy by a linear method [6], one approach to achieve both is to develop a non-linear method that blends both high and low (first order) solutions [12, 13, 15].

This paper is motivated by the previous work for MPP high-order finite element ALE remap [2]. While the FCT method presented in [2] (denoted by DG-FCT throughout this manuscript) is formally MPP and satisfactory up to \mathbb{Q}_3 finite elements, using higher order polynomial spaces leads to oscillations as the ones shown in Figure 1. The main cause of these oscillations is that the sparsity pattern of the DG-FCT advection matrix includes a high number of degrees of freedom, causing high variations in the definition of maximum and minimum admissible values. It is clear that special effort is needed to address the case of higher order (greater than \mathbb{Q}_3) polynomial spaces.

In this paper we address the aforementioned issue by presenting a new FCT-*like* MPP method that is applicable to both advection remap and the transport equation. After reviewing the underlying DG

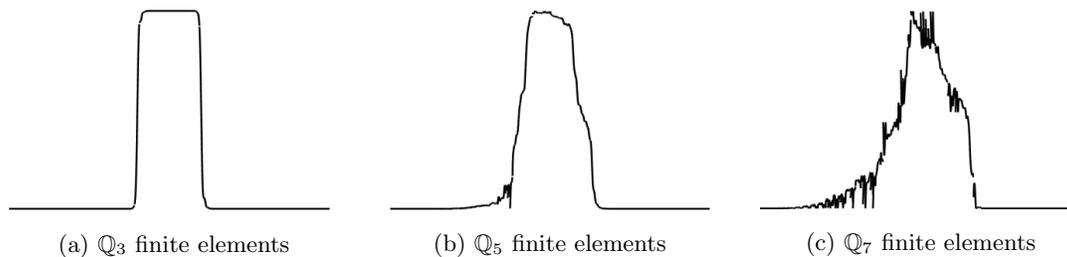


Figure 1: Example of the problematic behavior of the DG-FCT method on a 1D advection problem with refinement in polynomial order. All solutions are within their original $[0, 1]$ bounds.

finite element discretization and the DG-FCT method, we introduce the concept of localized stencils that define tighter bounds for each degree of freedom. This is combined with an element-based FCT formulation (denoted DG-EFCT). In DG-EFCT, instead of adjusting all incoming fluxes, one adjusts the value at each degree of freedom, causing a mass error in each cell. These mass errors are corrected by a sub-zonal mass redistribution process that results in a non-linear problem to be solved in each cell.

The remainder of this manuscript has the following organization. In Section 2 we review the basic transport equation we are interested in solving as well the general high-order DG formulation (which is non-MPP). Low order (or 1st order) MPP methods are the foundations for any non-linear, high-order FCT method and in Section 3 we show that the choice of basis functions in the DG formulation has a large effect on the quality of the low-order MPP method, and in particular, that use of the Bernstein polynomials yields a better low-order method than the traditional Gauss-Legendre or Gauss-Lobatto bases. Review of the DG-FCT approach in the context of high-order DG, followed by description of the main challenge this paper addresses, is presented in Section 4. In Section 5 we present a “localized” DG-FCT method which uses a reduced order stencil for computing bounds. In Section 6 we review the DG-EFCT approach. Unlike DG-FCT, DG-EFCT does not naturally conserve mass, so an additional correction step is required. We describe two approaches for recovering mass conservation, one based on a uniform (element-wise) flux rescaling (DG-EFCT-U) and one based on a non-linear mass redistribution solve (DG-EFCT-N). Section 7 describes how the new methods can be used for the purposes of ALE remap, where the velocity field is interpreted as the mesh displacement field. Finally, in Section 8 we present numerical results on a set 2D and 3D benchmark problems.

2 Preliminaries

We consider the transport equation given by

$$\partial_t u(\mathbf{x}, t) = \nabla \cdot (\mathbf{v}(\mathbf{x}, t)u(\mathbf{x}, t)), \quad \forall(\mathbf{x}, t) \in \Omega \times [0, T], \quad (1a)$$

$$u(\mathbf{x}, t = 0) = u_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega, \quad (1b)$$

where $\Omega \subset \mathbb{R}^d$ is an open domain, $0 < T \in \mathbb{R}$ is the final time, $u : \Omega \times [0, T] \rightarrow \mathbb{R}$ is the transported solution, $\mathbf{v} : \Omega \times [0, T] \rightarrow \mathbb{R}^d$ is divergence-free velocity field with $d \in \{1, 2, 3\}$ being the spatial dimension, and $u_0 : \Omega \rightarrow \mathbb{R}$ is the initial condition. In this study we consider periodic boundary conditions for 1D problems, and $\mathbf{v}(\mathbf{x} \in \partial\Omega) \cdot \mathbf{n} = 0$ for 2D and 3D problems.

It is our aim to obtain a solution that preserves the maximum principle locally. Given a finite element solution $u_h^n = \sum_i \phi_i U_i^n$ at time $t = t_n$, the solution $u_h^{n+1} = \sum_i \phi_i U_i^{n+1}$ of a Maximum-Principle Preserving (MPP) method is said to satisfy the discrete maximum principle locally if

$$\min_{j \in N_i} U_j^n =: U_i^{\min} \leq U_i^{n+1} \leq U_i^{\max} := \max_{j \in N_i} U_j^n, \quad \forall i, \quad (2)$$

where N_i defines some neighborhood of the i -th degree of freedom (DOF). Conventionally, N_i is defined through the sparsity pattern of the discrete advection operator. If continuous Galerkin finite elements are used, this sparsity pattern is given by the support of the i -th shape function, ϕ_i . With discontinuous Galerkin finite elements, this sparsity pattern consists of all DOFs on the given cell and on adjacent cells sharing a face with it.

2.1 High-Order non-MPP Spatial Discretization

Consider a computational mesh \mathcal{T}_h with internal faces \mathcal{F}_h . We define the discontinuous finite dimensional space $X_h = \{\phi(\mathbf{x}) \in L^2(\Omega) : \phi|_K \in \mathbb{Q}|_K, \forall K \in \mathcal{T}_h\}$ where $\mathbb{Q}|_K$ is a polynomial space over the element K . Let $\{\phi_1, \dots, \phi_N\}$ be a basis of X_h , where $N = \dim(X_h)$, such that $\sum_i^N \phi_i(\mathbf{x}) = 1$.

Consider the transport equation (1), multiply it by $\phi \in X_h$, integrate over Ω and integrate by parts the advection term to obtain

$$\int_{\Omega} (\partial_t u) \phi d\mathbf{x} = - \sum_{K \in \mathcal{T}_h} \int_K u(\mathbf{v} \cdot \nabla \phi) d\mathbf{x} + \sum_{f \in \mathcal{F}_h} \int_f u(\mathbf{v} \cdot \mathbf{n}_f) \phi ds, \quad (3)$$

where $\mathbf{s} \in \mathbb{R}^{d-1}$ and \mathbf{n}_f is the unit normal vector at face f . Let $u_h \in X_h$ be the finite element approximation of u . Since u_h is discontinuous across f we can't replace u by u_h in (3) or we would obtain multiple values over f . Therefore, we define numerical fluxes associated with the internal faces to get

$$\int_{\Omega} (\partial_t u_h) \phi d\mathbf{x} = - \sum_{K \in \mathcal{T}_h} \int_K u_h(\mathbf{v} \cdot \nabla \phi) d\mathbf{x} + \sum_{f \in \mathcal{F}_h} \int_f \{u_h \mathbf{v} \cdot \mathbf{n}_f\}_* [[\phi]] ds, \quad (4a)$$

where $[[\phi]] := \phi^- - \phi^+$, $\phi^\pm(\mathbf{x}) = \lim_{\xi \rightarrow 0^+} \phi(\mathbf{x} \pm \xi \mathbf{n}_f(\mathbf{x}))$ and

$$\{u_h \mathbf{v} \cdot \mathbf{n}_f\}_* = (\mathbf{v} \cdot \mathbf{n}_f) \left(\frac{u_h|_{K_1} + u_h|_{K_2}}{2} \right) - \frac{1}{2} |\mathbf{v} \cdot \mathbf{n}_f| [[u_h]], \quad (4b)$$

which is known as Godunov (upwind) flux [17, 19]. Method (4) can be recast in matrix-vector form as

$$M \frac{dU^H(t)}{dt} = KU^H(t), \quad (5a)$$

where $U^H(t) = [U_1^H(t), \dots, U_N^H(t)]^t$ are the DOFs of the finite element solution $u_h(\mathbf{x}, t)$ at time t and M and K are the mass and transport matrices whose ij -th elements are given by:

$$M_{ij} = \int_{\Omega} \phi_i \phi_j d\mathbf{x}, \quad (5b)$$

$$K_{ij} = - \int_{\Omega} \phi_j (\mathbf{v} \cdot \nabla \phi_i) d\mathbf{x} + \sum_{f \in \mathcal{F}_h} \int_f \{ \phi_j \mathbf{v} \cdot \mathbf{n}_f \}_* [[\phi_i]] ds. \quad (5c)$$

The method (5) is mass conservative in the following sense:

$$\int_{\Omega} u_h(\mathbf{x}, t) d\mathbf{x} = \int_{\Omega} u_h(\mathbf{x}, t=0) d\mathbf{x} \iff \sum_j U_j^H(t) m_j = \sum_j U_j^H(t=0) m_j,$$

where $m_j = \sum_i \int_{\Omega} \phi_i \phi_j d\mathbf{x} = \int_{\Omega} \phi_j d\mathbf{x}$. To see this consider the row sum of (5),

$$\begin{aligned} \sum_i \sum_j M_{ij} \left(\frac{dU_j^H(t)}{dt} \right) &= \sum_i \sum_j K_{ij} U_j^H(t) \implies \sum_j m_j \left(\frac{dU_j^H(t)}{dt} \right) = \sum_j U_j^H(t) \sum_i K_{ij} = 0 \\ &\implies \sum_j U_j^H(t) m_j = \sum_j U_j^H(t=0) m_j, \end{aligned}$$

where K has zero column sums since

$$\sum_i K_{ij} = - \int_{\Omega} \phi_j \left[\mathbf{v} \cdot \nabla \left(\sum_i \phi_i \right) \right] d\mathbf{x} + \sum_{f \in \mathcal{F}_h} \int_f \{ \phi_j \mathbf{v} \cdot \mathbf{n}_f \}_* [[\sum_i \phi_i]] ds = 0$$

by partition of unity, i.e., $\sum_i \phi_i = 1$. Note that K also has zero row sums since $\nabla \cdot \mathbf{v} = 0$ is assumed.

2.2 Time Discretization

For simplicity we consider Forward Euler integration in time. However, we extend the results to high-order approximations via Strong Stability Preserving (SSP) methods [7]. Moreover, all numerical experiments, unless otherwise noted, are performed via a third order (three stage) Runge-Kutta SSP method. The time discretization of (5) via Forward Euler is given by:

$$M \left(\frac{U^H - U^n}{\Delta t} \right) = K U^n, \quad (6)$$

where U^n and U^H are the DOFs of the high-order finite element solution $u_h(\mathbf{x}, t)$ at time $t = t_n$ and $t = t_{n+1}$, respectively.

3 Low-Order MPP Method

We consider the first-order MPP approach in [15, 16]. This method is based on applying a *discrete upwinding* to the transport matrix K of a high-order scheme and lumping the mass matrix M . This leads to

$$M^* \left(\frac{U^L - U^n}{\Delta t} \right) = K^* U^n, \quad (7a)$$

where M^* and K^* are the lumped mass matrix and the upwinded transport matrix respectively. They are given as follows:

$$M^* = M + L, \quad (7b)$$

$$K^* = K + D, \quad (7c)$$

where L and D are given by

$$L_{ij} = -M_{ij}, \quad D_{ij} = \max(0, -K_{ij}, -K_{ji}), \quad (7d)$$

for the off-diagonal elements and

$$L_{ii} = -\sum_{j \neq i} L_{ij}, \quad D_{ii} = -\sum_{j \neq i} D_{ij}, \quad (7e)$$

otherwise. Note that the lumping and upwinding are algebraic processes, and therefore the corresponding low-order solution depends on the choice of the basis functions. The matrices L and $-D$ are algebraic diffusion matrices, i.e., they are symmetric, have non-positive off-diagonal entries and have zero row and column sum. These are the typical characteristics of the discretization of the Laplace operator $-\Delta$ [15]. A critical property of our DG method is that the matrices L and D are block diagonal with $L_{ij} = D_{ij} = 0$ if nodes i and j belong to different mesh cells (see Section 6.1). The size of diagonal blocks is given by the number of DOFs per element. The method is first order accurate. Mass conservation in (7) follows from the fact that both K and D have zero column sums.

One can see that the above method is MPP by rewriting (7) as

$$U_i^L = \sum_j R_{ij} U_j^n.$$

Here the off-diagonal entries of $R_{ij} = [(M^*)^{-1}(M^* + \Delta t K^*)]_{ij}$ are positive by the construction of M^* and K^* , and the diagonal ones can be made positive by choosing Δt small enough. In addition, if $\mathbb{1}$ is the vector of ones, then

$$R\mathbb{1} = (M^*)^{-1}(M^* + \Delta t K^*)\mathbb{1} = (\mathbb{1} + \Delta t (M^*)^{-1} K^* \mathbb{1}) = \mathbb{1},$$

which is true since $K^* \mathbb{1} = (K + D)\mathbb{1} = 0$. Therefore, for any $i = 1, \dots, N$, U_i^L is a convex combination of U^n .

3.1 Low-order MPP method with positive v.s. non-positive basis functions

In this subsection we motivate our choice of the positive Bernstein polynomials, see Section 2.2 in [2], as finite element basis functions. Additional arguments are given in Section 6.1, which shows that the use of Bernstein polynomials results in high-order flux corrections that conserve the mass cell-wise, a crucial characteristic we exploit in this scheme.

Given an MPP solution $\{U_1, \dots, U_N\}$, one might be interested in the solution at locations different than those of the DOFs; for example, if the solution is going to be used to solve other equations, one might need the solution at the quadrature points. It is desired for this solution to be also in bounds. Consider a finite dimensional space X_h and let $\{\phi_1, \dots, \phi_N\}$ be a basis of X_h . The solution $u_h \in X_h$, at a point $\mathbf{x} \in \Omega$, is obtained via the interpolation $u_h(\mathbf{x}) = \sum_i U_i \phi_i(\mathbf{x})$. The solution $u_h(\mathbf{x})$ is guaranteed to be in bounds provided the interpolation $\sum_i U_i \phi_i(\mathbf{x})$ is a convex combination of $\{U_1, \dots, U_N\}$; i.e., provided the shape functions $\{\phi_1, \dots, \phi_N\}$ are positive $\forall \mathbf{x} \in \Omega$ and form a partition of unity. Finite element spaces based on Bernstein polynomials are positive and form a partition of unity.

Furthermore, positive basis functions give low-order solutions of better quality. To demonstrate this, we consider $\Omega = (0, 1) \subset \mathbb{R}$, with velocity $\mathbf{v} = 1$ and initial condition given by $u(\mathbf{x}, t = 0) = \cos(2\pi(x - 0.5))$. Periodic boundary conditions are imposed, and the initial condition is used as exact solution at $T = 1$. Figure 2 compares the Bernstein solutions to the ones obtained by the Gauss-Legendre and Gauss-Lobatto nodal bases. The solution is qualitatively similar for lower order polynomials. However, as we increase the order, the quality of the solution with nodal basis functions is highly reduced. With Gauss-Legendre the solution is extremely dissipated for the larger order polynomials. With Gauss-Lobatto the solution is also more dissipated than if positive basis functions are used but not as much as with Gauss-Legendre; however, the solution is less smooth than before.

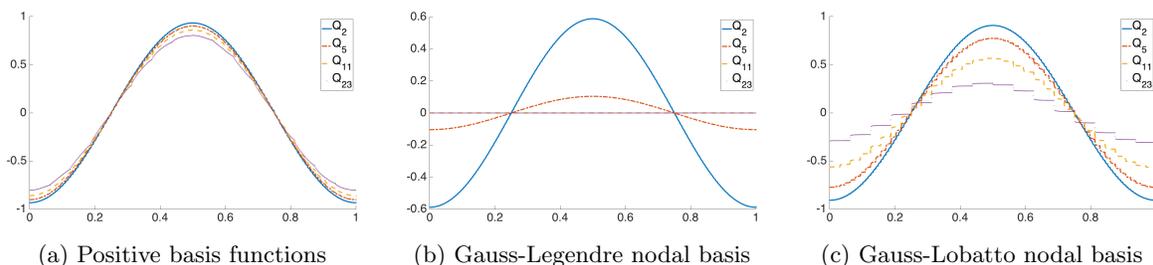


Figure 2: 1D transport of a smooth function via the low-order method (7) using positive v.s. nodal basis functions. In each case we consider polynomial spaces \mathbb{Q}_2 , \mathbb{Q}_5 , \mathbb{Q}_{11} and \mathbb{Q}_{23} . The number of cells is adjusted to have 384 DOFs in all simulations.

It is also important to note that even with positive basis functions the solution is more dissipated as we increase the order of the polynomials. To further study this we consider the same experiment with the same polynomial spaces but increase the number of DOFs. In Figure 3 we show the solution with 768, 1536 and 3072 DOFs. In all cases we obtained more dissipated solutions as the order of the polynomial space is increased. However, the solutions get closer as we increase the order. Convergence

study for the Bernstein basis is presented in Table 1. One can see in this table that the convergence rate is slightly increased as the order is increased. From these two results we expect that using higher-order polynomials eventually gives better results. However, the resolution needed might be too large. This is important to consider when the low-order method is used within the FCT methodology; i.e., this behavior influences the quality of the high-order solution as we increase the order of the polynomial space.

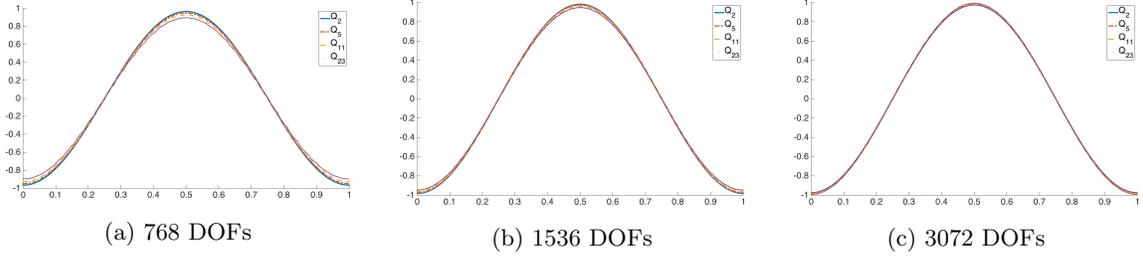


Figure 3: 1D transport of a smooth function via the low-order method (7) using positive basis functions with polynomial spaces \mathbb{Q}_2 , \mathbb{Q}_5 , \mathbb{Q}_{11} and \mathbb{Q}_{23} . The number of cells is adjusted to have (a) 768, (b) 1536 and (c) 3072 DOFs.

Cells	\mathbb{Q}_1	conv	\mathbb{Q}_2	conv	\mathbb{Q}_3	conv
32	1.708E-01		1.534E-01		1.385E-01	
64	9.163E-02	0.898	8.186E-02	0.906	7.340E-02	0.916
128	4.752E-02	0.947	4.231E-02	0.952	3.780E-02	0.957
256	2.420E-02	0.974	2.151E-02	0.976	1.918E-02	0.979
Cells	\mathbb{Q}_5	conv	\mathbb{Q}_{11}	conv	\mathbb{Q}_{23}	conv
32	1.189E-01		8.942E-02		6.585E-02	
64	6.247E-02	0.928	4.641E-02	0.946	3.383E-02	0.961
128	3.202E-02	0.964	2.364E-02	0.973	1.715E-02	0.981
256	1.622E-02	0.982	1.193E-02	0.987	8.630E-03	0.990

Table 1: $L^1(\Omega)$ -convergence of low-order method (7).

4 Edge-Based Flux Corrected Transport (DG-FCT)

In this section we revisit the Flux Corrected Transport methodology by [4] and [20], we also refer to [13] for more details. This method interpolates between a low-order MPP and a high-order non-MPP solution. The resulting method is denoted as DG-FCT throughout this work.

The high-order method (6) can be rewritten as

$$m_i(U_i^H - U_i^L) = \sum_j (M^* - M)_{ij}(U_j^H - U_j^n) - \Delta t D_{ij} U_j^n, \quad (8)$$

where U_i^L is the low-order solution given by (7), and the right hand side is a flux correction. Note that for any $i = 1, \dots, N$, $\sum_j (M^* - M)_{ij} = 0$ and $\sum_j D_{ij} = 0$, by the properties of the lumped matrix M^* and the diffusive operator D . Therefore,

$$\sum_j D_{ij} U_j^n = \sum_{j \neq i} D_{ij} U_j^n + D_{ii} U_i^n = \sum_{j \neq i} (U_j^n - U_i^n) D_{ij} = \sum_j (U_j^n - U_i^n) D_{ij}.$$

Since D is symmetric, the pairs $[(U_j^n - U_i^n) D_{ij}, (U_i^n - U_j^n) D_{ji}]$ form an anti-symmetric matrix. Similarly,

$$\sum_j (M^* - M)_{ij}(U_j^H - U_j^n) = \sum_j (M^* - M)_{ij}(\delta U_j - \delta U_i),$$

where $\delta U := U^H - U^n$. Here the pairs $[(M^* - M)_{ij}(\delta U_j - \delta U_i), (M^* - M)_{ji}(\delta U_i - \delta U_j)]$ also form an anti-symmetric matrix. Let

$$f_{ij} := (M^* - M)_{ij}(\delta U_j - \delta U_i) - \Delta t D_{ij}(U_j^n - U_i^n).$$

Then the high-order method (6) can be written as

$$U_i^H = U_i^L + m_i^{-1} \sum_{j \neq i} f_{ij}. \quad (9)$$

In this form, it is clear that flux correction improves the accuracy of the low-order method to make it high-order. In addition, it is responsible for the high-order solution to be in bounds. The idea behind DG-FCT is to limit this correction whenever the solution bounds are violated. Doing this we get

$$U_i^{n+1} = U_i^L + m_i^{-1} \sum_{j \neq i} \alpha_{ij} f_{ij}, \quad (10)$$

where α_{ij} 's are the flux limiters computed as follows:

$$\alpha_{ij} := \begin{cases} \min(R_i^+, R_j^-) & \text{if } f_{ij} \geq 0, \\ \min(R_i^-, R_j^+) & \text{otherwise,} \end{cases} \quad (11a)$$

where

$$R_i^+ := \begin{cases} \min\left(1, \frac{Q_i^+}{P_i^+}\right), & P_i^+ \neq 0, \\ 1 & \text{otherwise,} \end{cases} \quad R_i^- := \begin{cases} \min\left(1, \frac{Q_i^-}{P_i^-}\right), & P_i^- \neq 0, \\ 1 & \text{otherwise,} \end{cases} \quad (11b)$$

$$P_i^+ := \sum_j \max(0, f_{ij}), \quad P_i^- := \sum_j \min(0, f_{ij}), \quad (11c)$$

$$Q_i^+ := m_i(U_i^{\max} - U_i^L), \quad Q_i^- := m_i(U_i^{\min} - U_i^L). \quad (11d)$$

Here U_i^{\max} and U_i^{\min} are defined as local maxima and minima of U^L (see below).

We refer to [2] for a numerical validation of the DG-FCT method using the low- and the high-order methods (7) and (6), respectively.

Mass conservation follows from the symmetry properties of α_{ij} and f_{ij} , namely, the row sum of (10) is

$$\sum_i m_i (U_i^{n+1} - U_i^L) = \sum_i \sum_{j \neq i} \alpha_{ij} f_{ij} = \sum_{i,j \neq i} \alpha_{ij} f_{ij} + \alpha_{ji} f_{ji} = \sum_{i,j \neq i} \alpha_{ij} (f_{ij} - f_{ij}) = 0.$$

Remark 4.0.1 (Maximum Principle Preservation). *Assume that U^L satisfies the local discrete maximum principle; i.e., $U_i^{\min} \leq U_i^L \leq U_i^{\max}$ for all $i = 1, \dots, N$. Then the solution of (10) satisfies the local discrete maximum principle; i.e., $U_i^{\min} \leq U_i^{n+1} \leq U_i^{\max}$ for all $i = 1, \dots, N$. To see this we follow [8, 15]. Assume that $P_i^+ \neq 0$, then using (11) we get*

$$\begin{aligned} m_i (U_i^{n+1} - U_i^L) &= \sum_j \alpha_{ij} f_{ij} \leq \sum_{j \neq i, f_{ij} \geq 0} \alpha_{ij} f_{ij} = \sum_{j \neq i, f_{ij} \geq 0} \min(R_i^+, R_j^-) f_{ij} \leq \sum_{j \neq i, f_{ij} \geq 0} R_i^+ f_{ij} \\ &\leq \frac{Q_i^+}{P_i^+} \sum_{j \neq i, f_{ij} \geq 0} f_{ij} = \frac{Q_i^+}{P_i^+} \sum_{j \neq i} \max(0, f_{ij}) = Q_i^+ = m_i (U_i^{\max} - U_i^L); \end{aligned}$$

therefore, $U_i^{n+1} \leq U_i^{\max}$. If $P_i^+ = 0$, then

$$m_i (U_i^{n+1} - U_i^L) \leq \sum_{j \neq i, f_{ij} \geq 0} R_i^+ f_{ij} = R_i^+ \sum_{j \neq i} \max(0, f_{ij}) = R_i^+ P_i^+ = 0$$

for any R_i^+ . Provided $U_i^L \leq U_i^{\max}$, we get $P_i^+ = 0 \leq m_i (U_i^{\max} - U_i^L)$, which implies $U_i^{n+1} \leq U_i^{\max}$. The lower bound $U_i^{\min} \leq U_i^{n+1}$ is proven similarly.

The DG-FCT methodology, as revisited here, produces good quality results and recovers the full accuracy of the high-order methods when the polynomial spaces are of relatively low-order. However, as the order of the polynomials is increased spurious oscillations are introduced. This is true even though the method is still MPP. To illustrate this problem we consider a 1D problem with non-smooth initial data

$$u_h(x, t = 0) = \begin{cases} 1, & \forall x \in (0.4, 0.6) \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

over $\Omega = (0, 1) \subset \mathbb{R}$ and velocity given by $v = 1$ and obtain the solution using different polynomial spaces. The results are shown in Figure 4.

In addition, we consider a more complicated non-smooth initial condition in 2D, shown in Figure 5, with $\Omega = (0, 100) \times (0, 100) \subset \mathbb{R}^2$ and velocity $\mathbf{v} = (10, 10)$. We compute the solution at time $t = 4$ using \mathbb{Q}_2 and \mathbb{Q}_5 spaces with the cells adjusted to have the same number of DOFs. The results are shown in Figure 5.

The problem is clear, as we consider higher-order spaces the oscillations become more drastic making the solution unacceptable. In the remainder of this work we propose various approaches to reduce these oscillations.

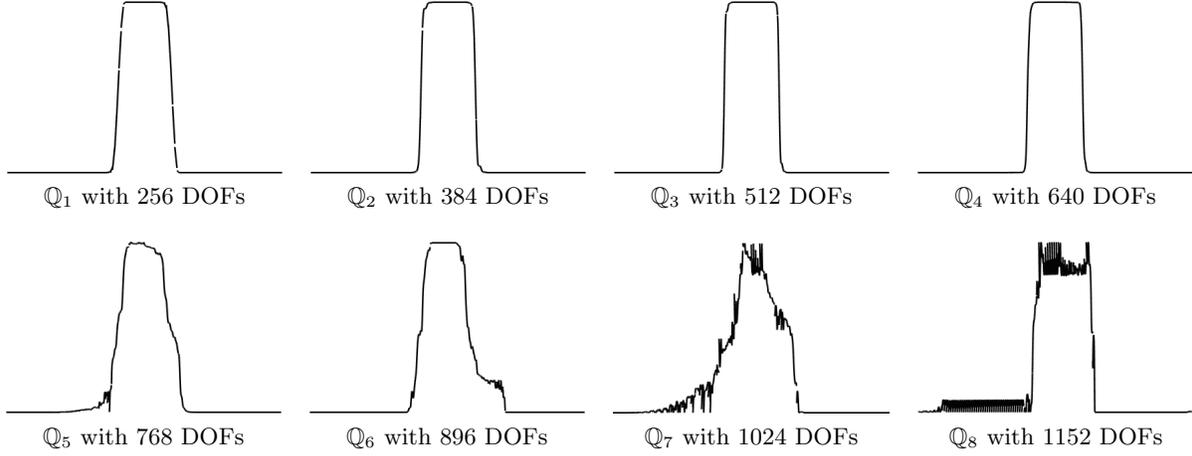


Figure 4: 1D simulations with non-smooth initial data using the DG-FCT method (10) for different polynomial spaces. In all simulations 128 cells are used and, therefore, they have different number of DOFs.

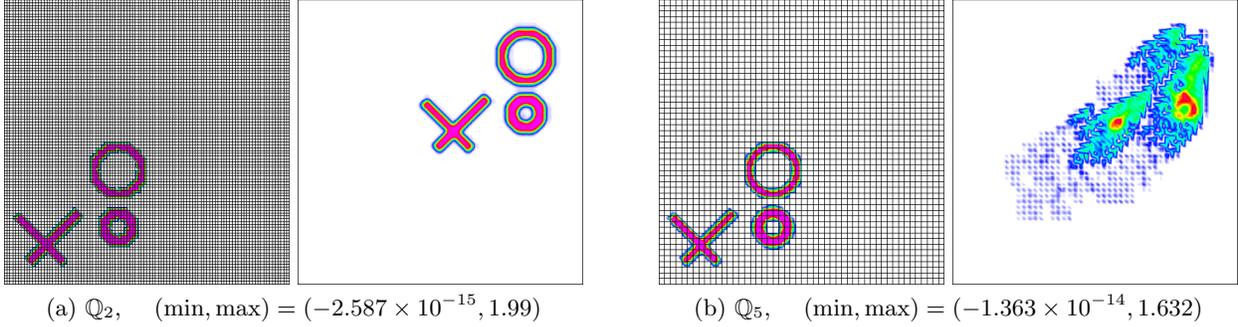


Figure 5: 2D simulations with non-smooth initial data using the DG-FCT method (10) for Q_2 and Q_5 spaces. The number of cells is adjusted so that 90000 DOFs are used in both simulations. For each case, we show (left) the initial condition with the grid and (right) the solution at $t = 4$.

5 Localized DG-FCT

The DG-FCT method from Section 4 is local in the sense given by the sparsity pattern of the transport matrix K . For Discontinuous Galerkin finite elements this sparsity pattern includes all DOFs on a given cell and on cells sharing a face with it, see Figure 6a. When the order of the polynomial space is small, the sparsity pattern includes few DOFs; however, as we increase the order, the number of DOFs in the sparsity pattern increases. The method *loses* locality with respect to DOFs; nevertheless, it is fixed with respect to number of cells. In an extreme case we can consider a single cell with a polynomial space of order as large as needed to have roughly certain number of DOFs; in this situation, the local and global maximum principles are equivalent, since the sparsity pattern includes all DOFs in the finite element space. This motivates the idea of considering tighter bounds. We propose to localize the bounds by mimicking the stencil of a first-order space; i.e., for a given i -th DOF we consider those at locations adjacent to i . Let N_i be the conventional neighborhood for the i -th DOF. For the finite element space \mathbb{Q}_k , we use tighter bounds given by the stencil

$$N_i^* = \left\{ j \in N_i : \text{dist}(i, j) \leq \frac{1}{k} \sqrt{d} \right\}, \quad (13)$$

where $d \in \{1, 2, 3\}$ is the space dimension and $\text{dist}(i, j)$ is the Euclidean distance between the two DOFs' images on the reference element. Defining N_i^* with respect to the reference element makes the approach applicable to unstructured grids. In Figure 6, we consider a representative DOF in thick blue and show the conventional or full stencil via the sparsity pattern of the transport matrix K ; in addition, we show the tighter bounds (13), mimicking a stencil for a first-order space. Note that since DG discretization is used we have two DOFs at the faces, this is denoted by using a red circle and a black cross in those locations.

Remark 5.0.2 (Low-order method is non-MPP in the tighter bounds). *The low-order method (7) is guaranteed to produce an MPP solution in the conventional bounds; i.e., including all DOFs in the sparsity pattern of K . Since the tighter bounds considers a smaller set of DOFs there is no guarantee the low-order solution is MPP in this set.*

Due to remark 5.0.2, we can't use the DG-FCT methodology with the bounds in (2) given by the tighter stencil (13). To overcome this we modify the bounds in (2) to be

$$U_i^{\min} = \min \left(U_i^L, \min_{j \in N_i^*} U_j^n \right), \quad (14a)$$

$$U_i^{\max} = \max \left(U_i^L, \max_{j \in N_i^*} U_j^n \right), \quad (14b)$$

which guarantees the low-order solution is in bounds and, therefore, we can apply the DG-FCT methodology. We now repeat the simulations from Section 4 using the tighter bounds (14). The results are shown in Figures 7 and 8. We observe the oscillatory behavior, although not completely eliminated, is highly reduced. It is clear also the high amount of dissipation introduced as the order of the polynomial space is increased. In the next section we propose an FCT-like methodology that reduces even more the oscillatory behavior and yields less dissipated solutions.

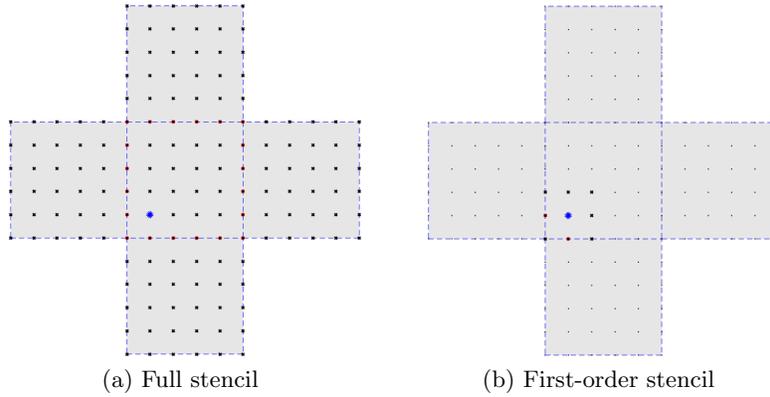


Figure 6: Stencil to compute bounds for a representative DOF. In (a) we show the conventional or full stencil in a DG discretization. In (b) we mimic a \mathbb{Q}_1 space. The thick blue mark represents the DOF for which we compute the bounds, the thick black marks represent the DOFs included to compute the bounds, the non-thick black dots in (b) represent all DOFs in the sparsity pattern of K and the red marks indicate a double DOF for the given location (they are also considered for the computation of the bounds).

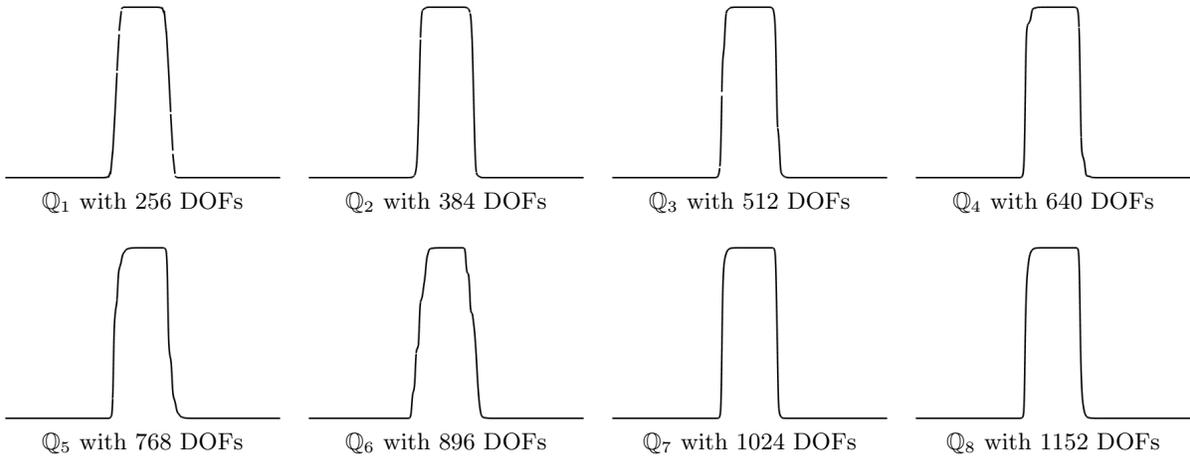


Figure 7: 1D simulations with non-smooth initial data using the localized DG-FCT with bounds (14). The low- and high-order methods are given by (7) and (6), respectively. We consider different polynomial spaces. In all simulations 128 cells are used and, therefore, they have different number of DOFs.

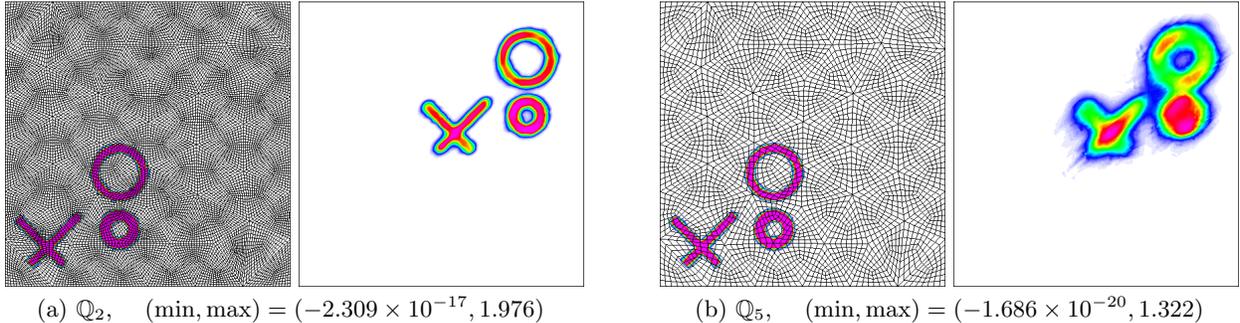


Figure 8: 2D simulations with non-smooth initial data using the localized DG-FCT method with bounds (14) on an unstructured grid. The low- and high-order methods are given by (7) and (6), respectively. We use \mathbb{Q}_2 and \mathbb{Q}_5 spaces with the number of cells adjusted so that 127872 DOFs are used in both simulations. For each case, we show (left) the initial condition with the grid and (right) the solution at $t = 4$.

6 Element-Based Flux Corrected Transport (DG-EFCT)

In the DG-FCT method revisited in Section 4, we start with two methods that are mass conservative. One is low-order and MPP, and the other is high-order, but non-MPP. Then, an interpolation is made from the low- to the high-order solution to obtain a solution that is MPP. For any DOF U_i^{n+1} , there are as many interpolating parameters as DOFs in the neighborhood of U_i^{n+1} . These interpolating parameters are designed in a way that conserves mass.

In this work we present an FCT-like method (which we denote by DG-EFCT) that considers two MPP solutions. One is low-order and mass conservative, and the other is (presumably) high-order, but non-conservative. Then, we interpolate from the low- to the high-order solution to recover mass conservation cell-wise. In contrast to DG-FCT, in this method, for any DOF U_i^{n+1} we have just one interpolating parameter. This interpolating parameter is designed to maintain the solution in bounds. It is important to emphasize that the recovery in mass conservation is obtained per cell and not globally. Moreover, we propose a methodology to localize even more this redistribution of mass inside a cell. Recovering the conservation of mass within a cell is possible due to local mass properties of the low- and high-order methods we consider in this work and by using Discontinuous Galerkin spatial discretization. We explain this in more detail in the next section.

6.1 Mass conservation of low- and high-order methods

In this section we show that the low- and the high-order solutions in (7) and (6) have the same mass on any given cell $K \in \mathcal{T}_h$; i.e., $\sum_{i \in N_K} m_i U_i^H = \sum_{i \in N_K} m_i U_i^L$, where $N_K = \{i : \mathbf{x}_i \in K\}$ is the set of node numbers for the DOFs associated with K . To prove this, we rely on the specific DG finite element formulation as well as the use of the positive shape functions described in Section 2. First,

we rewrite the high-order method as

$$m_i(U_i^H - U_i^L) = f_i^H, \quad (15a)$$

where f_i^H is a high-order flux correction given by

$$f_i^H = \sum_{j \in N_i} (M^* - M)_{ij} \delta U_j - \Delta t \sum_{j \in N_i} D_{ij} U_j^n. \quad (15b)$$

Here $\delta U_j := U_j^H - U_j^n$. Given any cell $K \in \mathcal{T}_h$ consider

$$\sum_{i \in N_K} f_i^H = \sum_{j \in N_i} \delta U_j \sum_{i \in N_K} (M^* - M)_{ij} - \Delta t \sum_{j \in N_i} U_j \sum_{i \in N_K} D_{ij}.$$

It is our aim to show that $\sum_{i \in N_K} f_i^H = 0 \implies \sum_{i \in N_K} m_i U_i^H = \sum_{i \in N_K} m_i U_i^L$. Since we use a DG discretization, all shape functions are supported on a single cell. Therefore, $M_{jj}^* = \sum_{i=1, \dots, N} M_{ij} = \sum_{i \in N_K} M_{ij}$ and, hence, $\sum_{i \in N_K} (M^* - M)_{ij} = 0$.

From Section 3, we recall that $\sum_{i \in [1, \dots, N]} D_{ij} = 0$. Now we show that $D_{ij} = 0$ whenever i and j belong to different cells. Recall the definition of D_{ij} from Section 3:

$$D_{ij} = \max(0, -K_{ij}, -K_{ji}) \text{ if } i \neq j, \quad D_{ii} = -\sum_{j \neq i} D_{ij} \quad (16)$$

We just need to consider the off-diagonal elements and assume they belong to different cells. From Section 2.1, the i, j -th element of the transport matrix is given by

$$K_{ij} = -\int_{\Omega} \phi_j(\mathbf{v} \cdot \nabla \phi_i) d\mathbf{x} + \sum_{f \in \mathcal{F}_h} \int_f \{\phi_j \mathbf{v} \cdot \mathbf{n}_f\}_* [[\phi_i]] ds, \quad (17)$$

where the first integral is zero since i and j belong to different cells and each shape function is supported on its corresponding cell. Recall the definition of the numerical flux:

$$\{\phi \mathbf{v} \cdot \mathbf{n}_F\}_* = (\mathbf{v} \cdot \mathbf{n}_F) \left(\frac{\phi|_{K_1} + \phi|_{K_2}}{2} \right) - \frac{1}{2} |\mathbf{v} \cdot \mathbf{n}_F| [[\phi]], \quad (18a)$$

$$[[\phi]] = \phi^- - \phi^+, \quad (18b)$$

$$\phi^\pm(\mathbf{x}) = \lim_{\xi \rightarrow 0^+} \phi(\mathbf{x} \pm \xi \mathbf{n}_f(\mathbf{x})). \quad (18c)$$

Suppose the normal vector \mathbf{n}_F points from cell K_1 to cell K_2 . Then we get $[[\phi]] = \phi|_{K_1} - \phi|_{K_2}$. Assume ϕ_j is supported on cell K_1 and ϕ_i on cell K_2 , then $[[\phi_j]] = \phi_j|_{K_1}$ and $[[\phi_i]] = -\phi_i|_{K_2}$, which leads to

$$\{\phi_j \mathbf{v} \cdot \mathbf{n}_F\}_* [[\phi_i]] = [|\mathbf{v} \cdot \mathbf{n}_F| - (\mathbf{v} \cdot \mathbf{n}_F)] \left(\frac{\phi_j|_{K_1} \phi_i|_{K_2}}{2} \right),$$

which is non-negative regardless of the sign of $\mathbf{v} \cdot \mathbf{n}_F$ provided the shape functions are positive. Similarly, if ϕ_j is supported on cell K_2 and ϕ_i on cell K_1 we get

$$\{\phi_j \mathbf{v} \cdot \mathbf{n}_F\}_* [[\phi_i]] = [(\mathbf{v} \cdot \mathbf{n}_F) + |\mathbf{v} \cdot \mathbf{n}_F|] \left(\frac{\phi_j|_{K_2} \phi_i|_{K_1}}{2} \right),$$

which is also non-negative provided we use positive shape functions. Therefore, $K_{ij} \geq 0$ whenever i and j don't belong to the same cell. This implies that $D_{ij} = 0$ whenever i and j don't belong to the same cell.

From $\sum_{i \in [1, \dots, N]} D_{ij} = 0$ and $D_{ij} = 0$ whenever i and j don't belong to the same cell we conclude that $\sum_{i \in N_K} D_{ij} = 0$. Therefore, we get

$$\sum_{i \in N_K} f_i^H = \sum_j \delta U_j \sum_{i \in N_K} (M^* - M)_{ij} - \Delta t \sum_j U_j \sum_{i \in N_K} D_{ij} = 0.$$

The property that the mass of the high-order flux correction is zero for any given cell is crucial for the method presented in this work. This allows us to consider non mass conservative flux corrections that assure the solution is in bounds and then adjust those fluxes to recover mass conservation per cell. To do this we need to adjust the fluxes on any cell without modifying fluxes in other cells. This is possible since we consider DG finite elements.

6.2 Clipped solution

The first stage of this method is to clip the solution considering some local bounds. We can consider different options depending on the stencil; i.e., we can consider the full or conventional stencil N_i (figure 6a) or the tighter stencil N_i^* from equation (13) (figure 6b). In either case we obtain $\{U_i^{\min}, U_i^{\max}\}$. Then we consider the high-order solution U^H from method (6) to get

$$U_i^* = \min(U_i^{\max}, \max(U_i^H, U_i^{\min})), \quad (19)$$

where U_i^* is the clipped solution. Note that, since $N_i^* \subset N_i$, the clipped solution U_i^* is in bounds in both the tight and the full stencil, i.e.,

$$\min_{j \in N_i} U_j^n \leq \min_{j \in N_i^*} U_j^n \leq U_i^* \leq \max_{j \in N_i^*} U_j^n \leq \max_{j \in N_i} U_j^n.$$

In Figure 9 we revisit the non-smooth 1D problem (12) and show the results of clipping the solution with the bounds computed via the full and the tighter stencil. We use \mathbb{Q}_5 and \mathbb{Q}_{11} spaces. It is clear that non-physical oscillations are present when the full stencil is considered. For this reason we always compute the bounds (2) using the tighter stencil N_i^* (13). Next, in Figure 10 we show results of the same problem considering different spaces and refinement levels. Two observations can be made from this figure. First, phase errors appear due to not conserving mass. Mass conservation is addressed in the next section. Second, the clipped solution becomes more dissipated as one considers higher order spaces.

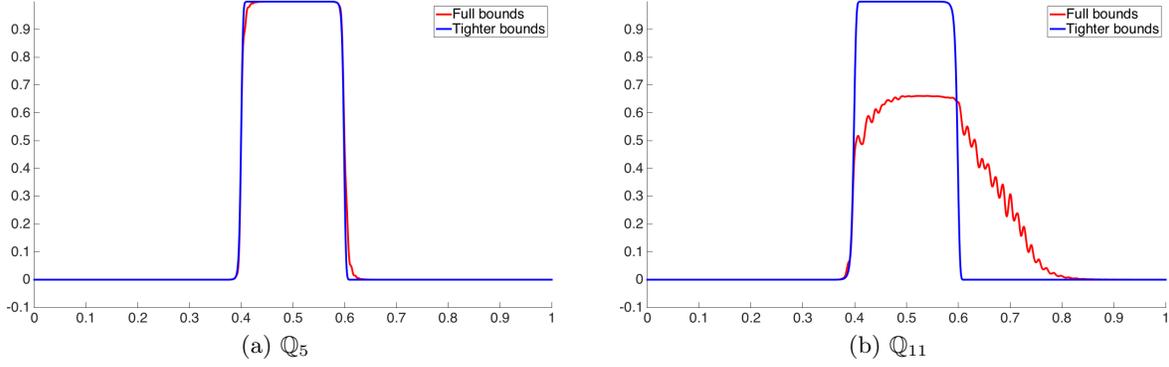


Figure 9: Solution clipping via (19) for non-smooth initial data. We consider Q_5 and Q_{11} spaces using the full and the tighter stencils. The number of cells is adjusted so that 768 DOFs are used in both simulations.

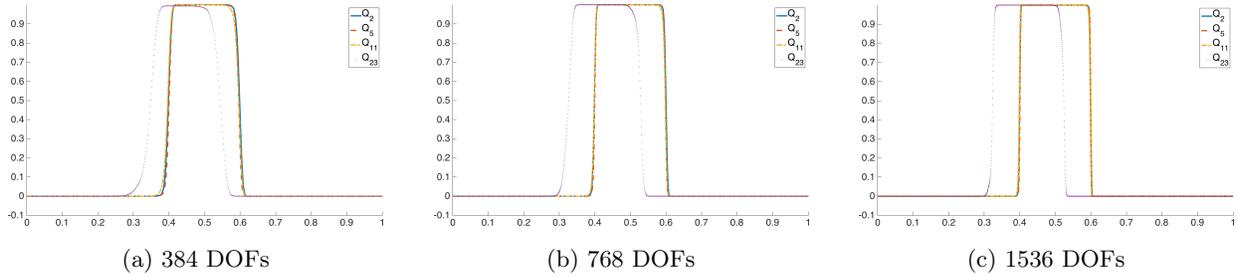


Figure 10: Solution clipping via (19) for non-smooth initial data. We consider Q_2 , Q_5 , Q_{11} and Q_{23} spaces using the tighter stencil. The number of cells is adjusted to have (a) 384, (b) 768 and (c) 1536 DOFs.

6.3 Local recovery of mass conservation

In this section we consider the clipped solution U_i^* and recover mass conservation per cell. In Section 6.1 we saw that the high-order flux correction

$$f_i^H = m_i(U_i^H - U_i^L) \quad (20)$$

has zero mass within a cell; i.e., $\sum_{i \in N_K} f_i^H = 0, \forall K \in \mathcal{T}_h$. Given the clipped solution U_i^* , define

$$f_i^* := m_i(U_i^* - U_i^L). \quad (21)$$

Here f_i^* is a flux correction from the low-order to the clipped solution. To recover mass conservation per cell we need to modify the fluxes $f_i^* \mapsto f_i$, so that $\sum_{i \in N_K} f_i = 0, \forall K \in \mathcal{T}_h$. The modification of the fluxes has to be done without creating violations of the Maximum Principle; i.e., the solution must remain in bounds.

6.4 Mass conservation via flux scaling

To enforce local mass conservation in element $K \in \mathcal{T}_h$, define the local degree of freedom U_i^{n+1} via

$$m_i(U_i^{n+1} - U_i^L) = \alpha_i f_i^*, \quad (22)$$

where $0 \leq \alpha_i \leq 1$. The nodal correction factors α_i are defined so that $\sum_{i \in K} \alpha_i f_i^* = 0$. Each factor α_i contributes to enforcing local mass conservation on the element K to which the local degree of freedom U_i belongs. Note that this is always possible. In particular, one might choose $\alpha_i = 0$ which gives back the low-order solution. Assuming the low-order solution is conservative, i.e., $\sum_i m_i U_i^L = \sum_i m_i U_i^0$, we get

$$\sum_i m_i U_i^{n+1} = \sum_i m_i U_i^0 \implies \int_{\Omega} u_h(\mathbf{x}, t) d\mathbf{x} = \int_{\Omega} u_h(\mathbf{x}, 0) d\mathbf{x},$$

i.e., the method (22) is mass conservative.

Theorem 6.4.1 (Maximum-Principle Preservation (MPP)). *Given $0 \leq \alpha_i \leq 1$ and provided U_i^* and U_i^L are in bounds; i.e., $U_i^{\min} \leq U_i^{*/L} \leq U_i^{\max}$, the method (22) is MPP; i.e., $U_i^{\min} \leq U_i^{n+1} \leq U_i^{\max}$, where $U_i^{\min} := \min_{j \in N_i} U_j^n$ and $U_i^{\max} := \max_{j \in N_i} U_j^n$.*

Proof. Rewrite (22) as

$$U_i^{n+1} = U_i^L + \alpha_i m_i^{-1} f_i^* = U_i^L + \alpha_i (U_i^* - U_i^L) = \alpha U_i^* + (1 - \alpha) U_i^L.$$

Since $0 \leq \alpha_i \leq 1$ and $U_i^{*/L} \leq U_i^{\max}$, we get

$$U_i^{n+1} \leq \alpha_i U_i^{\max} + (1 - \alpha_i) U_i^{\max} = U_i^{\max} \implies U_i^{n+1} \leq U_i^{\max}.$$

The lower bound is proven similarly. □

There are different strategies to choose the interpolating parameters. A first approach, which we refer to as uniform scaling, is to scale down the dominant fluxes by the same factor. Consider a representative cell $K \in \mathcal{T}_h$ and define

$$S_K^+ = \sum_{\substack{f_i^* > 0 \\ i \in N_K}} f_i^*, \quad S_K^- = \sum_{\substack{f_i^* < 0 \\ i \in N_K}} f_i^*.$$

If $S_K^+ + S_K^- > 0$, $i \in N_K$, we choose

$$\alpha_i := \begin{cases} -\frac{S_K^-}{S_K^+} & \text{if } f_i^* > 0, \\ 1 & \text{otherwise.} \end{cases} \quad (23a)$$

If $S_K^+ + S_K^- < 0$, $i \in N_K$, we choose

$$\alpha_i := \begin{cases} 1 & \text{if } f_i^* \geq 0, \\ -\frac{S_K^+}{S_K^-} & \text{otherwise,} \end{cases} \quad (23b)$$

and if $S_K^+ + S_K^- = 0$ then $\alpha_i = 1$. It is easy to see that $0 \leq \alpha_i \leq 1$ and $\sum_{i \in N_K} \alpha_i f_i^* = 0, \forall K \in \mathcal{T}_h$. We refer to the combination of the DG-EFCT method and this uniform scaling approach as the DG-EFCT-U method.

Another option for choosing the interpolating parameters $\{\alpha_i\}$ is to solve the following minimization problem:

$$\min_{\alpha_i} \sum_i (U_i^{n+1} - U_i^H)^2 = \min_{\alpha_i} \sum_i (\alpha_i f_i^* - f_i^H)^2 \quad (24a)$$

such that

$$0 \leq \alpha_i \leq 1, \quad \sum_{i \in N_K} \alpha_i f_i^* = 0, \quad (24b)$$

for all cells $K \in \mathcal{T}_h$. One can find many more strategies to find the interpolating parameters $\{\alpha_i\}$. Bochev et al. [3] used this kind of constrained optimization in the context of conservative remapping. However, in their *flux-variable flux-target* (FVFT) algorithm, local mass conservation was enforced by imposing the anti-symmetry constraint on the fluxes leading to a global inequality-constrained quadratic programming problem, which has relatively high computational cost. As an alternative the authors developed a globally conservative *mass-variable mass-target* (MVMT) remap method which is more efficient. In contrast, in our DG method, the optimization problems for $\{\alpha_i\}$ are decoupled (i.e. they are local to each element) and can be solved in parallel.

In Figure 11 we show results of the non-smooth 1D problem (12) computed by the DG-EFCT-U method (23) and the minimization problem (24) with \mathbb{Q}_2 , \mathbb{Q}_5 , \mathbb{Q}_{11} and \mathbb{Q}_{23} spaces. The number of

cells is adjusted to have 768 DOFs in all simulations. We can easily identify a problem, namely, the solution is more dissipated as we consider higher-order spaces. We recall from Figures 3 and 10 that the low-order method and the clipping process produce more dissipative results as we increase the order, which is part of the problem. In addition, the flux scaling used to recover mass conservation is introducing additional dissipation as the order is increased.

The recovery of mass conservation is performed cell-wise. When higher-order spaces are used, more DOFs have to be considered within a cell in order to achieve this. Therefore, locality is lost with respect to DOFs. Motivated by this, in the following section we propose a process to recover mass conservation that is more localized, that is, the distribution of mass is performed differently in different parts of a given cell.

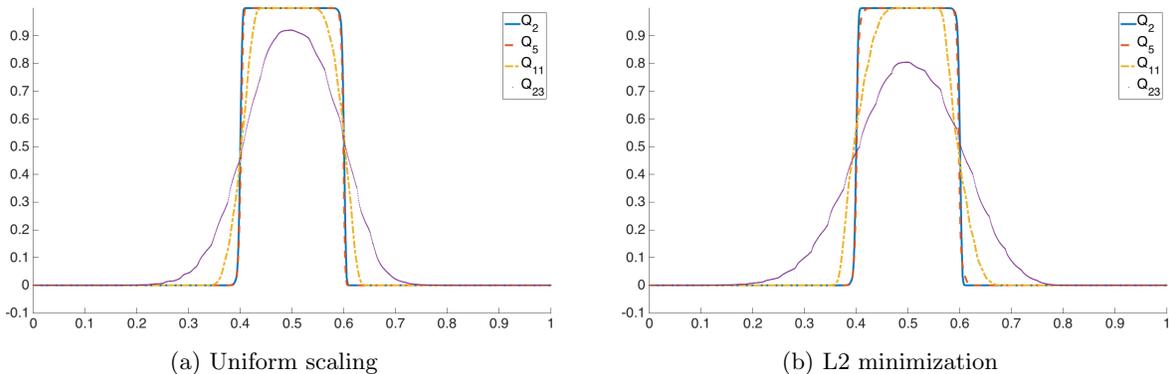


Figure 11: 1D simulations with non-smooth initial data via the uniform scaling (23) v.s. the L2 minimization problem (24) for the recovery of mass conservation. We consider Q_2 , Q_5 , Q_{11} and Q_{23} spaces. The number of cells is adjusted to have 768 DOFs in all simulations.

6.5 Mass conservation via penalization

In this section the recovery of mass conservation is localized at sub-cell level. We discuss two possible strategies.

In [5], a *repair* approach is used to obtain an MPP mass conservative method. This repair considers a given cell and those adjacent to it. If a criteria-satisfying solution cannot be found, more cells are considered, until a mass conservative, in bounds, solution is obtained. We can apply this idea to redistribute the mass restricted to a cell, i.e., consider a DOF within a cell and try to distribute the mass considering just adjacent DOFs in such a way that the mass for this set of DOFs equals the mass of the high-order flux f^H on the set. If that is impossible without violating the MPP, we would consider a larger set. In the worst case scenario, we would have to consider the entire cell and use an approach similar to those presented in Section 6.4.

Another approach to distribute the mass within a cell is as follows. In [12], the author obtains a solution in bounds by clipping the solution and doing a global fix in mass using a Lagrange multiplier.

We use that same idea, but restricted to a single cell. Suppose the mass error is positive in cell K , namely, $\delta_K := \sum_{i \in N_K} f_i^* > 0$. The final solution is based on the fluxes $\{\bar{f}_i\}$, which provide a mass-conservative penalization of the fluxes $\{f_i^*\}$:

$$m_i(U_i^{n+1} - U_i^L) = \bar{f}_i, \quad \text{where} \quad \bar{f}_i = \begin{cases} f_i^* & \text{if } f_i^* \leq 0, \\ f_i^* - \lambda_K w_i & \text{if } f_i^* > 0 \text{ and } f_i^* - \lambda_K w_i \geq 0, \\ 0 & \text{if } f_i^* > 0 \text{ and } f_i^* - \lambda_K w_i < 0. \end{cases} \quad (25a)$$

In this case (when $\delta_K > 0$) negative fluxes are not penalized, but every $f_i^* > 0$ is modified to $\bar{f}_i \in [0, f_i^*]$. Each quantity $w_i > 0$ (defined later) controls the amount of penalization applied to the flux f_i^* . The common factor $\lambda_K > 0$ (also defined later) is used to retain mass conservation. Before going into the details of w_i and λ_K , we point out that the formulation (25a) is MPP. More specifically, U_i^{n+1} is always between U_i^L and U_i^* , which are both in bounds.

Looking at (25a), one can see that the choice $w_i = f_i^*$ reduces to the method of uniform scaling (23), where all positive fluxes are scaled with the same constant. Starting from the idea that we prefer to penalize fluxes that are different from the high-order ones, we could construct a non-uniform, localized penalization terms $w_i = m_i |U_j^* - U_j^H|$. However, this choice is too aggressive, over-correcting in a way that is not smooth. The localization can be relaxed by taking a maximum over a local neighborhood, i.e., $w_i = \max_{j \in N_i^*} m_i |U_j^* - U_j^H|$. This choice localizes penalizations only to those values which were diffused to satisfy bounds plus the immediate stencil. An important requirement is that $w_i > 0$ whenever $f_i^* > 0$, because having some $w_i = 0$ implies that the flux f_i^* is not penalized, which may lead to a situation where mass conservation cannot be restored. The final step in producing a smoother penalization is to blend the localized and uniform formulations with a tunable parameter, so that:

$$w_i = (1 - \theta) f_i^* + \theta \max_{j \in N_i^*} m_j |U_j^* - U_j^H| > 0, \quad (25b)$$

where $\theta \in [0, 1)$ and N_i^* is the tighter stencil described in Figure 6. Higher θ increases of the influence of the localized penalization; i.e., big differences with the high-order solution increase the penalization amount, which in the worst case makes $\bar{f}_i = 0$, and thus $U^{n+1} = U^L$.

Having all penalization terms $w_i > 0$, we compute λ_K by solving the non-linear equation

$$\sum_{i \in N_K} \bar{f}_i(\lambda_K) = 0, \quad (26)$$

and thus enforcing mass conservation. Note that each function $\bar{f}_i(\lambda_K)$ is continuous, piecewise linear and non-increasing, hence the same holds for their sum. Furthermore,

$$\sum_{i \in N_K} \bar{f}_i(0) = \sum_{f_i^* < 0} f_i^* + \sum_{f_i^* \geq 0} f_i^* = \delta_k > 0, \quad \sum_{i \in N_K} \bar{f}_i(+\infty) = \sum_{f_i^* < 0} f_i^* \leq 0,$$

hence (26) defines a unique λ_K if at least one $f_i^* < 0$, otherwise the solutions for λ_K are an interval of the form $[\lambda^*, +\infty)$. Note that by definition $w_i > 0$ whenever $f_i^* > 0$.

The case $\delta_K < 0$ is treated similarly, so that penalization is applied to the negative fluxes:

$$m_i(U_i^{n+1} - U_i^L) = \bar{f}_i, \quad \text{where} \quad \bar{f}_i = \begin{cases} f_i^* & \text{if } f_i^* \geq 0, \\ f_i^* - \lambda_K w_i & \text{if } f_i^* < 0 \text{ and } f_i^* - \lambda_K w_i \leq 0, \\ 0 & \text{if } f_i^* < 0 \text{ and } f_i^* - \lambda_K w_i > 0, \end{cases} \quad (27a)$$

$$w_i = (1 - \theta)f_i^* - \theta \max_{j \in N_i^*} m_j |U_j^* - U_j^H| < 0. \quad (27b)$$

In this case equation (26) defines a unique λ_K , because each function $\bar{f}_i(\lambda_K)$ is continuous, piecewise linear and non-decreasing, and

$$\sum_{i \in N_K} \bar{f}_i(0) = \sum_{f_i^* > 0} f_i^* + \sum_{f_i^* \leq 0} f_i^* = \delta_k < 0, \quad \sum_{i \in N_K} \bar{f}_i(+\infty) = \sum_{f_i^* > 0} f_i^* \geq 0.$$

We refer to the combination of the DG-EFCT method and this non-linear penalization approach as the DG-EFCT-N method. In Figure 12, we consider the non-smooth 1D problem (12) and compare the results using the DG-EFCT-U and the DG-EFCT-N (with $\theta = 0.99$) methods. The improvement in the solutions is clear.

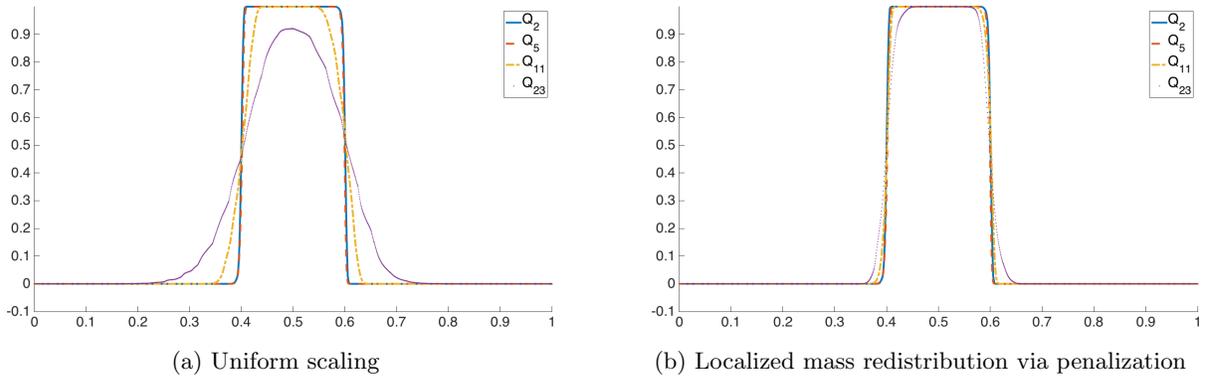


Figure 12: 1D simulations with non-smooth initial data. Comparison between the (a) DG-EFCT-U and (b) DG-EFCT-N methods. We consider \mathbb{Q}_2 , \mathbb{Q}_5 , \mathbb{Q}_{11} and \mathbb{Q}_{23} spaces. The number of cells is adjusted to have 768 DOFs in all simulations.

7 Applicability to Advection Based Remap

Because the original motivation behind this study was in the context of advection based finite element remap, and remap results are presented in Section 8, this section is needed to establish the connection between advection remap and the spatial discretization from Section 2.1. Detailed description of the DG advection based remap is given in [2], and only the major points are repeated here for completeness.

The goal of remap is to transfer a field ρ , defined on initial spatial domain $\tilde{\Omega} \subset \mathbb{R}^d$, to a new domain, $\Omega \subset \mathbb{R}^d$. For two corresponding points $\tilde{\mathbf{x}} \in \tilde{\Omega}$ and $\mathbf{x} \in \Omega$, we define a continuous transition function $F(\tilde{\mathbf{x}}, \tau) : \tilde{\Omega} \times [0, 1] \rightarrow \mathbb{R}^d$ and pseudo-velocity $\mathbf{v}(\tilde{\mathbf{x}}, \tau)$, such that

$$F(\tilde{\mathbf{x}}, 0) = \tilde{\mathbf{x}}, \quad F(\tilde{\mathbf{x}}, 1) = \mathbf{x}, \quad \mathbf{v}(\tilde{\mathbf{x}}, \tau) = \frac{\partial F}{\partial \tau}.$$

Here τ is pseudo-time in which the domain $\tilde{\Omega}$ transitions to Ω . Then one can introduce the concept of pseudo-material derivative along the trajectories $\mathbf{x}(\tau) = F(\tilde{\mathbf{x}}, \tau)$ as

$$\frac{d}{d\tau} \rho(x(\tau), \tau) = \frac{\partial \rho}{\partial \tau} + \mathbf{v} \cdot \nabla \rho.$$

Because the goal of remap is to preserve the initial field with respect to an Eulerian frame, i.e. $\frac{\partial \rho}{\partial \tau} = 0$, one commonly used approach for remap is to define ρ by solving the pseudo-time advection equation

$$\frac{d\rho}{d\tau} = \mathbf{v} \cdot \nabla \rho. \quad (28)$$

Utilizing the Reynolds transport theorem, basis functions that follow the pseudo-time deformation (i.e. $\frac{d\phi}{d\tau} = 0$) and (28), one can obtain

$$\frac{d}{d\tau} \int_{U(\tau)} \rho \phi \, d\mathbf{x} = - \int_{\Omega(\tau)} \rho \mathbf{v} \cdot \nabla \phi \, d\mathbf{x} + \int_{\partial\Omega(\tau)} \rho \mathbf{v} \cdot \mathbf{n} \phi \, ds. \quad (29)$$

Then choosing $\rho_h \in X_h$ as a finite element approximation of ρ , and using the Section 2.1 definitions of jump and average across faces, the semi-discrete approximation of (29) becomes

$$\frac{d}{d\tau} \int_{\Omega} \rho_h \phi \, d\mathbf{x} = - \sum_{K \in \mathcal{T}_h} \int_K \rho_h (\mathbf{v} \cdot \nabla \phi) \, d\mathbf{x} + \sum_{f \in \mathcal{F}_h} \int_f \{\rho_h \mathbf{v} \cdot \mathbf{n}_f\}_* [[\phi]] \, ds. \quad (30)$$

Note that the corresponding matrix-vector form can be written in terms of the matrices M and K from (5), namely,

$$\frac{d}{d\tau} (M \boldsymbol{\rho}) = K \boldsymbol{\rho},$$

where $\boldsymbol{\rho}$ is the vector of unknowns. Since the mesh $x(\tau)$ is moving, M and K depend on τ . Therefore, the system of ordinary differential equations for $\boldsymbol{\rho}$ has the form

$$M \frac{d\boldsymbol{\rho}}{d\tau} = \left(K - \frac{dM}{d\tau} \right) \boldsymbol{\rho}, \quad (31)$$

with advection matrix

$$\left(K - \frac{dM}{d\tau} \right)_{ij} = \int_{\Omega(\tau)} \phi_i (\mathbf{v} \cdot \nabla \phi_j) \, d\mathbf{x} - \sum_{f \in \mathcal{F}_h(\tau)} \int_f (\phi_i \mathbf{v} \cdot \mathbf{n}_f)_d [[\phi_j]] \, ds,$$

where $(\phi_i \mathbf{v} \cdot \mathbf{n}_f)_d$ denotes the downwind flux:

$$(\phi_i \mathbf{v} \cdot \mathbf{n}_f)_d = (\mathbf{v} \cdot \mathbf{n}_f) \{\phi_i\} + \frac{1}{2} |\mathbf{v} \cdot \mathbf{n}_f| [[\phi_i]].$$

It is instructive to view this advection matrix as the standard matrix K plus a correction term that takes into account the mesh motion $\frac{d\mathbf{x}}{d\tau} = \mathbf{v}$. Without going into details (see Section 3 in [2]), discrete lumping of M and upwinding of $K - \frac{dM}{d\tau}$ results in a conservative and monotone low-order method. Therefore, the aforementioned FCT methods can be applied to the problem (31) in order to obtain monotone and conservative high-order solution.

8 Numerical Examples

First we summarize some of the results that were already shown in the preceding sections:

- The positive Bernstein basis is an appealing option in the context of high-order monotone advection, see Figures 2, 3, and Table 1.
- The localized DG-FCT method (Figures 7 and 8) controls oscillations much better than the DG-FCT method (Figures 4 and 5).
- In the DG-EFCT methods with high-order spaces, the clipped solution benefits from using the localized bounds, see Figure 9.
- For high-order spaces, the DG-EFCT-N method is sharper than the DG-EFCT-U method and the minimization based approach (24), see Figures 12 and 11.

In this section we show additional results for the DG-EFCT-N method. We begin by presenting the method's convergence properties on smooth and non-smooth problems. Next, we consider 2D advection of non-trivial Q_5 functions and compare the DG-FCT, localized DG-FCT, DG-EFCT-U and DG-EFCT-N methods. Finally, we show the DG-EFCT-N method's behavior in the context of 2D and 3D advection remap of complex Q_5 functions.

All presented simulations use the parameter $\theta = 0.99$ and finite element functions are represented in the Bernstein basis. All experiments, unless otherwise noted, are performed via a third order Runge-Kutta SSP method. All results are generated with the finite element methods library MFEM [1].

8.1 Convergence tests

In this section we perform a series of convergence tests on the DG-EFCT-N method.

2D smooth profile without local extrema

Consider an initial condition given by

$$u_h(x, y, t = 0) = \tanh((y - 0.5)/0.25), \quad (32a)$$

over $\Omega = (0, 1) \times (0, 1)$ with velocity

$$(u, v) = (\sin(\pi x) \cos(\pi y) \sin(2\pi t), -\cos(\pi x) \sin(\pi y) \sin(2\pi t)). \quad (32b)$$

Since the velocity field is periodic and the problem is linear the exact solution at $T = 1$ coincides with the initial condition. We consider \mathbb{Q}_1 , \mathbb{Q}_2 and \mathbb{Q}_3 spaces and show the corresponding convergence results in Table 2. We obtain the expected convergence rates.

h	L^1 error	conv	h	L^1 error	conv	h	L^1 error	conv
1.25E-01	6.85E-03		6.25E-02	5.30E-04		4.17E-02	5.12E-05	
6.25E-02	1.77E-03	1.94	3.13E-02	6.18E-05	3.09	2.08E-02	3.66E-06	3.80
3.13E-02	4.18E-04	2.08	1.56E-02	6.91E-06	3.16	1.04E-02	2.19E-07	4.06
1.56E-02	1.01E-04	2.05	7.81E-03	7.73E-07	3.15	5.21E-03	1.15E-08	4.25

(a) \mathbb{Q}_1 space (b) \mathbb{Q}_2 space (c) \mathbb{Q}_3 space

Table 2: $L^1(\Omega)$ -convergence using the DG-EFCT-N method with a smooth initial profile without local extrema.

1D non-smooth profile

Next we consider the problem with non-smooth initial data from Section 4, equation (12). The problem is solved using the DG-EFCT-N method and, for reference, the discontinuous Galerkin discretization without limitation, i.e., the high-order method from Section 2.1. Table 3 shows convergence rates for \mathbb{Q}_1 , \mathbb{Q}_2 and \mathbb{Q}_3 spaces. Table 4 presents a similar comparison for \mathbb{Q}_2 , \mathbb{Q}_5 and \mathbb{Q}_{11} spaces. Note that, in Table 4, the number of cells is adjusted to have the same number of DOFs in each row, allowing a fair comparison between the errors produced by the different spaces. The DG-EFCT-N method generally produces larger errors on coarser grids, which is expected, since the low order and the clipped solutions are more dissipated for higher order polynomials. For all spaces we obtain convergence rates close to the optimal rate of 1. Up to \mathbb{Q}_5 , we observe better rates as we increase the polynomial degree.

1D smooth profile with local extrema

Finally we consider as initial condition

$$u_h(x, t = 0) = \cos(2\pi(x - 0.5)),$$

NDOFs	L^1 error	conv	NDOFs	L^1 error	conv	NDOFs	L^1 error	conv
64	6.17E-02		96	3.24E-02		128	2.24E-02	
128	3.69E-02	0.74	192	1.85E-02	0.81	256	1.22E-02	0.87
256	2.19E-02	0.75	384	1.05E-02	0.81	512	6.75E-03	0.85
512	1.30E-02	0.75	768	5.91E-03	0.83	1024	3.64E-03	0.89

(a) DG with \mathbb{Q}_1 (b) DG with \mathbb{Q}_2 (c) DG with \mathbb{Q}_3

NDOFs	L^1 error	conv	NDOFs	L^1 error	conv	NDOFs	L^1 error	conv
64	5.37E-02		96	3.36E-02		128	2.73E-02	
128	3.11E-02	0.78	192	1.82E-02	0.88	256	1.38E-02	0.98
256	1.81E-02	0.77	384	9.85E-03	0.88	512	7.35E-03	0.90
512	1.06E-02	0.77	768	5.38E-03	0.87	1024	3.88E-03	0.92

(d) DG-EFCT-N with \mathbb{Q}_1 (e) DG-EFCT-N with \mathbb{Q}_2 (f) DG-EFCT-N with \mathbb{Q}_3

Table 3: $L^1(\Omega)$ -convergence of (a)-(c) discontinuous Galerkin discretization and (d)-(f) the DG-EFCT-N method for a non-smooth initial profile. We consider \mathbb{Q}_1 , \mathbb{Q}_2 and \mathbb{Q}_3 spaces.

NDOFs	L^1 error	conv	NDOFs	L^1 error	conv	NDOFs	L^1 error	conv
192	1.85E-02		192	1.24E-02		192	1.39E-02	
384	1.05E-02	0.81	384	6.70E-03	0.89	384	6.35E-03	1.12
768	5.91E-03	0.83	768	3.89E-03	0.78	768	3.50E-03	0.86
1536	3.32E-03	0.83	1536	2.00E-03	0.96	1536	1.63E-03	1.09

(a) DG with \mathbb{Q}_2 (b) DG with \mathbb{Q}_5 (c) DG with \mathbb{Q}_{11}

NDOFs	L^1 error	conv	NDOFs	L^1 error	conv	NDOFs	L^1 error	conv
192	1.82E-02		192	2.25E-02		192	3.51E-02	
384	9.85E-03	0.88	384	1.07E-02	1.06	384	2.00E-02	0.81
768	5.38E-03	0.87	768	5.65E-03	0.92	768	1.08E-02	0.88
1536	2.95E-03	0.86	1536	2.69E-03	1.07	1536	6.14E-03	0.81

(d) DG-EFCT-N with \mathbb{Q}_2 (e) DG-EFCT-N with \mathbb{Q}_5 (f) DG-EFCT-N with \mathbb{Q}_{11}

Table 4: $L^1(\Omega)$ -convergence of (a)-(c) discontinuous Galerkin discretization and (d)-(f) the DG-EFCT-N method for a non-smooth initial profile. We consider \mathbb{Q}_2 , \mathbb{Q}_5 and \mathbb{Q}_{11} spaces and adjust the number of cells to have the same number of DOFs for each refinement of the convergence test.

over $\Omega = (0, 1)$ with velocity $v = 1$. We impose periodic boundary conditions and use the initial condition as exact solution at $T = 1$. We use \mathbb{Q}_1 , \mathbb{Q}_2 and \mathbb{Q}_3 to perform a convergence test via the DG-EFCT-N method and, for reference, the discontinuous Galerkin discretization from Section 2.1. The results are shown in Table 5. One can see that no better than (slightly higher than) second order is achieved. This issue is already discussed in [2], where the authors show that the dominating error is localized in the extremal regions, while high-order accuracy is obtained in the rest of the domain. Additional details about this problem can be found in [9], where it is shown that Total Variation Diminishing (TVD) methods can't achieve better than second order convergence (in the L^1 norm) around local extrema. To resolve this problem within the context of finite volumes, it is common to allow small violations on the total variation near local extrema. Popular examples are UNO [9], ENO [10, 11] and WENO [18] methods. In [21, 22], finite volumes and discontinuous Galerkin methods are used to obtain a solution that satisfies a strict (or global) maximum principle. To achieve high-order accuracy at local extrema, the authors reconstruct a polynomial inside cells from where the bounds are computed. A parameter-free smoothness indicator based on a hierarchical slope limiter for high-order DG methods may be used as regularity criterion for deactivation of FCT corrections at smooth extrema [14].

NDOFs	L^1 error	conv	NDOFs	L^1 error	conv	NDOFs	L^1 error	conv
64	1.52E-03		96	1.94E-05		128	2.40E-07	
128	3.57E-04	2.08	192	2.42E-06	3.00	256	1.50E-08	4.00
256	8.63E-05	2.04	384	3.02E-07	3.00	512	9.35E-10	4.00
512	2.12E-05	2.02	768	3.78E-08	3.00	1024	5.93E-11	3.97

(a) DG with \mathbb{Q}_1 (b) DG with \mathbb{Q}_2 (c) DG with \mathbb{Q}_3

NDOFs	L^1 error	conv	NDOFs	L^1 error	conv	NDOFs	L^1 error	conv
64	2.93E-03		96	3.61E-03		128	1.99E-03	
128	6.73E-04	2.12	192	6.96E-04	2.37	256	3.82E-04	2.38
256	1.55E-04	2.12	384	1.29E-04	2.43	512	7.04E-05	2.43
512	3.57E-05	2.11	768	2.33E-05	2.46	1024	1.22E-05	2.53

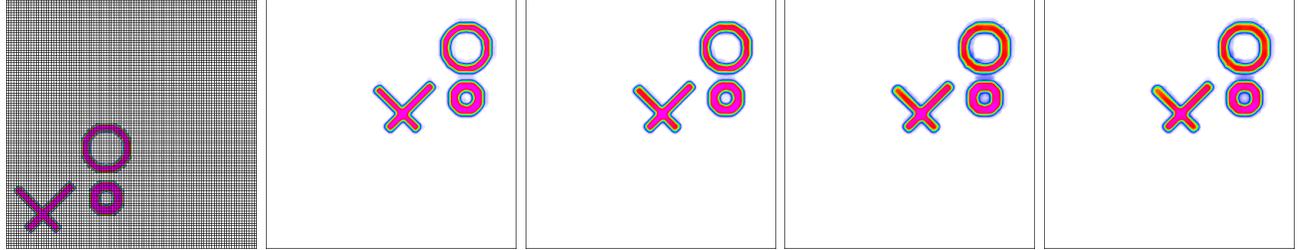
(d) DG-EFCT-N with \mathbb{Q}_1 (e) DG-EFCT-N with \mathbb{Q}_2 (f) DG-EFCT-N with \mathbb{Q}_3

Table 5: $L^1(\Omega)$ -convergence of (a)-(c) discontinuous Galerkin discretization and (d)-(f) the DG-EFCT-N method for a smooth initial profile with local extrema. We consider \mathbb{Q}_1 , \mathbb{Q}_2 and \mathbb{Q}_3 spaces.

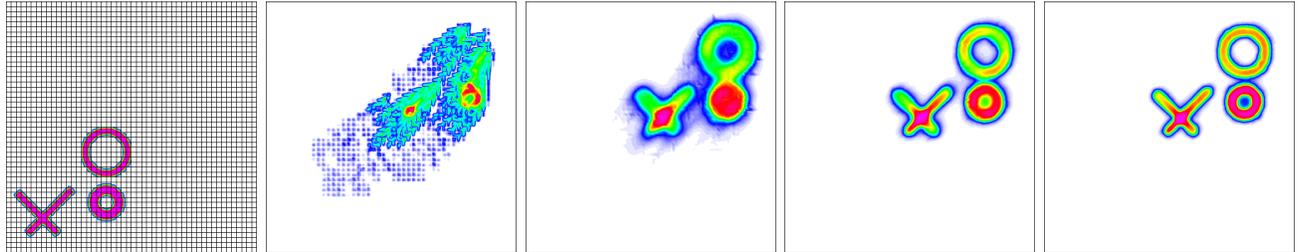
8.2 2D advection with constant velocity field

We consider $\Omega = (0, 100) \times (0, 100) \subset \mathbb{R}^2$, velocity $\mathbf{v} = (10, 10)$ and the discontinuous initial profile shown in the left panel in Figure 13. The cross shape is specified by the -45 degree rotation of the region $(x, y) \in (7, 10) \times (32, 13) \cup (14, 3) \times (17, 26)$, The lower ring's origin is at $(x, y) = (40, 20)$ and its radii are 3 and 7. The upper ring's origin is at $(x, y) = (40, 40)$ and its radii are 7 and 10.

The advected function is initialized to 1 in the aforementioned subdomains, and to 0 in the rest of the domain. We compute the solution using the DG-EFCT-U method (equations (22) and (23)) and DG-EFCT-N method (equations (25) and (27)). The results are shown in Figure 13. For comparison we also show the results of the DG-FCT revisited in Section 4 and the localized DG-FCT from Section 5. For all situations we use \mathbb{Q}_2 and \mathbb{Q}_5 spaces with number of cells adjusted to have 90000 DOFs.



(a) \mathbb{Q}_2 . From left to right: (min,max) = $(0, 2)$, $(-1.294 \times 10^{-15}, 1.99)$, $(-2.309 \times 10^{-17}, 1.976)$, $(-1.690 \times 10^{-18}, 1.952)$ and $(-8.207 \times 10^{-17}, 1.962)$.



(b) \mathbb{Q}_5 . From left to right: (min,max) = $(0, 2)$, $(-1.686 \times 10^{-20}, 1.322)$, $(-6.813 \times 10^{-15}, 1.642)$, $(-5.444 \times 10^{-18}, 1.582)$ and $(-1.271 \times 10^{-16}, 1.772)$.

Figure 13: 2D advection with constant velocity field. We consider (a) \mathbb{Q}_2 and (b) \mathbb{Q}_5 spaces with number of cells adjusted to have 90000 DOFs in all situations. **Left:** initial condition with the mesh. **Middle-left:** solution via the DG-FCT method (10). **Middle-middle:** solution via the localized DG-FCT method from Section 5. **Middle-right:** solution via the DG-EFCT-U method (22), (23). **Right:** solution via the DG-EFCT-N method (25), (27).

8.3 2D solid body rotation on unstructured mesh

To be consistent with the previously presented results in [2], we examine the behavior of the DG-EFCT-N method for the solid body rotation benchmark. For this test problem the DG-EFCT-N method is utilized as a tool for remapping fields, as described in Section 7. Description of the initial conditions is given in [2], Section 4.3.

Lagrangian mesh motion is prescribed through the velocity field $\mathbf{v}(x, y) = 0.1(y, -x)$. Each Lagrangian step rotates the mesh without modifying the discrete representation of the field. Each Lagrangian step is followed by a remesh / remap step where the mesh nodes are returned to their

original locations and the field is remapped between the two resulting meshes. This procedure results in the field's full 360 degree rotation about the origin at final time $2\pi/0.1$.

Two cases are considered, 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_5 discontinuous elements (36 dof/element).

For each case the DG-EFCT-N method is compared to the DG-FCT method used in [2]. All simulations use a fixed Lagrangian time step of size 5×10^{-2} . Each remap procedure performs five pseudo-time steps, and each pseudo-time step utilizes an explicit RK2 time integration.

Initial conditions and fields at final time for Q_2 and Q_5 are shown in Figure 14 and Figure 15, respectively. In Figure 14, one can see that both methods produce results of similar quality, with the DG-FCT method being slightly sharper. However, for the Q_5 case, it is clear in Figure 15 that the DG-FCT method develops spurious (but monotone) oscillations that eventually distort the original shapes. The improvement in quality with the DG-EFCT-N method is clear.

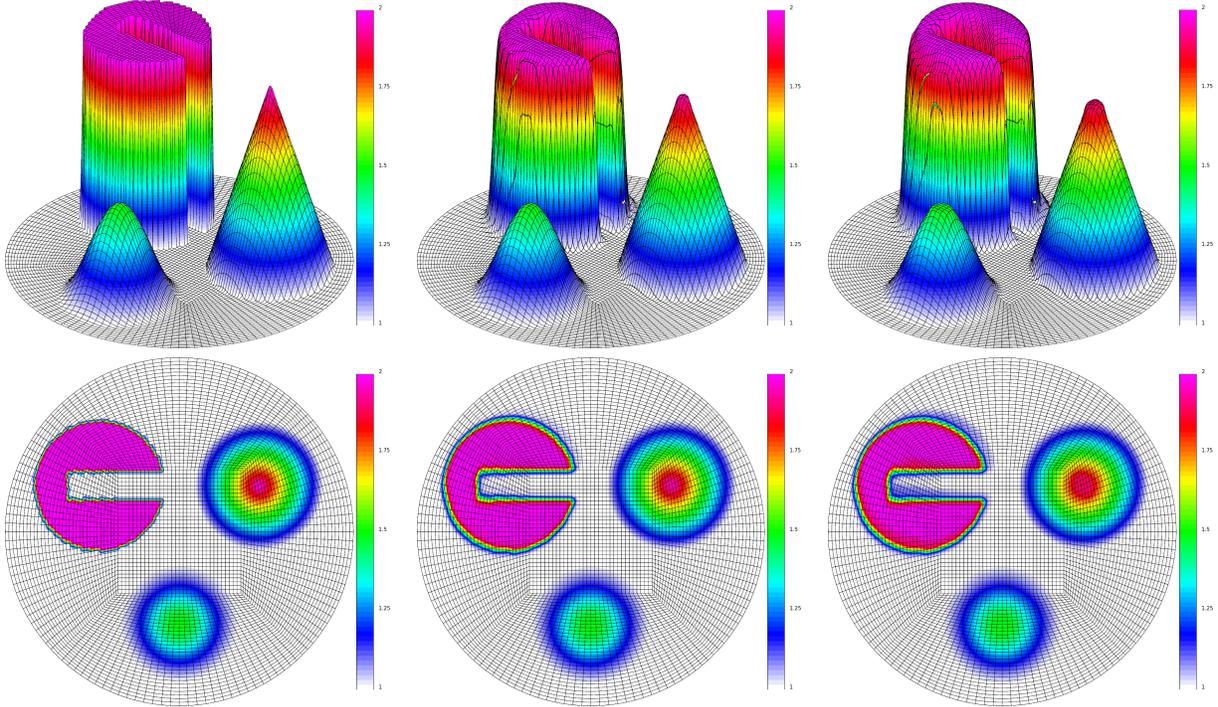


Figure 14: 3D view (top) and 2D view (bottom) of Q_2 fields (left to right): initial condition, DG-FCT result, DG-EFCT-N result.

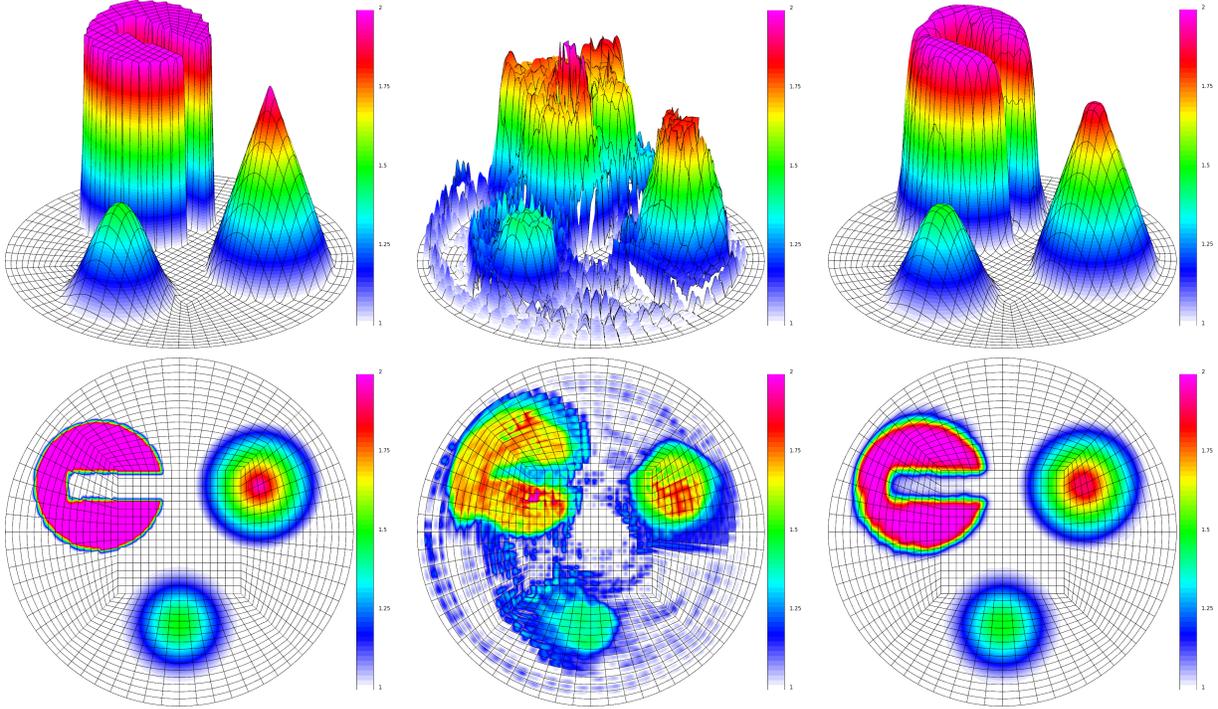


Figure 15: 3D view (top) and 2D view (bottom) of Q_5 fields (left to right): initial condition, DG-FCT result, DG-EFCT-N result.

8.4 3D Advection of “Balls and Jacks”

Here we consider a full 3D version of the “balls and jacks” advection benchmark. As with the previous section, we use the DG-EFCT method as a tool for remapping fields, as described in Section 7. This non-trivial test problem addresses the case of multi-material remap.

The computational mesh consists of a box of 32^3 elements in the domain $\Omega = (0, 100)^3$. Lagrangian mesh motion is prescribed through the velocity field $v(x, y, z) = \frac{1}{\sqrt{2}}(10, 10, 10)$ and the problem is run to a final time of $t = 6$.

We consider three non-overlapping subdomains. The subdomain Ω_2 consists of the shape

$$(x, y, z) \in (7, 32) \times (10, 13) \times (10, 13) \cup (14, 17) \times (3, 26) \times (10, 13) \cup (14, 17) \times (10, 13) \times (3, 26),$$

rotated by -45 degrees in the xy -plane, together with the shell (difference of two balls) centered at $(x, y, z) = (40, 20, 20)$ with radii 3 and 7, and the shell centered at $(x, y, z) = (40, 40, 40)$ with radii 7 and 10. The subdomain Ω_3 consists of the shape

$$(x, y, z) \in (2, 27) \times (30, 33) \times (30, 33) \cup (9, 12) \times (23, 46) \times (30, 33) \cup (9, 12) \times (30, 33) \times (23, 46),$$

together with the ball centered at $(x, y, z) = (40, 20, 20)$ with radius 3, the ball centered at $(x, y, z) = (40, 40, 40)$ with radius 7, and the shell centered at $(x, y, z) = (40, 20, 20)$ with radii 7 and 10. The

last subdomain is $\Omega_1 = \Omega / (\Omega_2 \cup \Omega_3)$. We advect three functions that correspond to these subdomains, namely,

$$\eta_1(x) = \begin{cases} 1 & \text{if } x \in \Omega_1, \\ 0 & \text{otherwise,} \end{cases} \quad \eta_2(x) = \begin{cases} 1 & \text{if } x \in \Omega_2, \\ 0 & \text{otherwise,} \end{cases} \quad \eta_3(x) = \begin{cases} 1 & \text{if } x \in \Omega_3, \\ 0 & \text{otherwise.} \end{cases}$$

We use Q_5 discontinuous elements which consist of 216 DoF per element in 3D. All simulations use a fixed Lagrangian time step of size 2×10^{-3} . The remap process occurs every 20 Lagrangian steps using an explicit RK2 time integration with a dynamically calculated pseudo timestep. Two cases are considered: using DG-FCT and using the new DG-EFCT-N method.

Initial conditions and results at the final time for each Q_5 calculation are shown in Figure 16, which illustrates the field

$$\eta = \eta_1 + 2\eta_2 + 3\eta_3.$$

As with the previous 2D result for the Q_5 case, it is clear in Figure 16 that the DG-FCT method develops spurious (but monotone) oscillations that completely distort the original shapes. The improvement in quality with the new DG-EFCT-N method is clear.

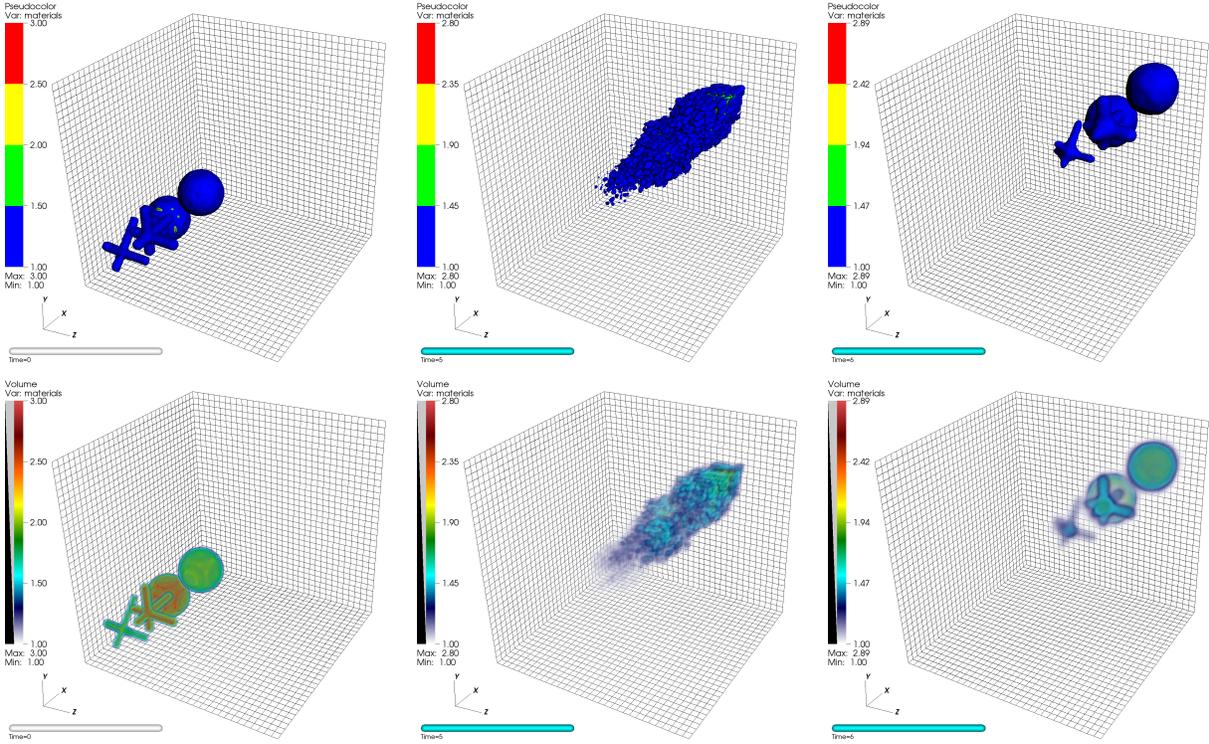


Figure 16: Iso-surface plot (top) and 3D volume rendering with transparency (bottom) of Q_5 fields (left to right): initial condition, DG-FCT result, DG-EFCT-N result.

9 Conclusion

We have presented a new method that addresses robustness issues with monotone advection of high-order (above Q_3) DG finite element spaces. The DG-EFCT-N method is based on the combined effects of Bernstein polynomial basis functions, DG approximation, localized bounds, element-based flux corrections, non-linear local mass redistribution. Results have been presented for finite element spaces up to Q_{23} in 1D and Q_5 in 2D. The DG-EFCT-N method obtains optimal convergence rates for both smooth and non-smooth fields, produces monotone solutions, and eliminates spurious oscillations. produces maximum principle preserving solutions and highly reduces spurious oscillations; indeed, in the numerical experiments we performed we didn't observe oscillatory behavior.

A future area of research will be to fully incorporate the new method in the context of high-order curvilinear ALE hydrodynamics, where the goal is to remap composite fields in a synchronized way, e.g., remap of mass while preserving bounds for density. We also plan to extend the presented methods with interface sharpening techniques.

10 Acknowledgments

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-684083. The work of D. Kuzmin was supported by the German Research Association (DFG) under grant KU 1530/15-1.

References

- [1] MFEM: Modular parallel finite element methods library. <http://mfem.org>.
- [2] Robert W Anderson, Veselin A Dobrev, Tzanio V Kolev, and Robert N Rieben. Monotonicity in high-order curvilinear finite element arbitrary Lagrangian–Eulerian remap. International Journal for Numerical Methods in Fluids, 77(5):249–273, 2015.
- [3] Pavel Bochev, Denis Ridzal, and Mikhail Shashkov. Fast optimization-based conservative remap of scalar fields through aggregate mass transfer. Journal of Computational Physics, 246:37–57, 2013.
- [4] Jay P Boris and David L Book. Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works. Journal of Computational Physics, 11(1):38–69, 1973.
- [5] Rao Garimella, Milan Kuchařík, and Mikhail Shashkov. Efficient algorithm for local-bound-preserving remapping in ALE methods. In Numerical Mathematics and Advanced Applications, pages 358–367. Springer, 2004.
- [6] Sergei Konstantinovich Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. Matematicheskii Sbornik, 89(3):271–306, 1959.

- [7] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor. Strong stability-preserving high-order time discretization methods. SIAM review, 43(1):89–112, 2001.
- [8] Jean-Luc Guermond, Murtazo Nazarov, Bojan Popov, and Yong Yang. A second-order maximum principle preserving lagrange finite element technique for nonlinear scalar conservation equations. SIAM Journal on Numerical Analysis, 52(4):2163–2182, 2014.
- [9] Ami Harten and Stanley Osher. Uniformly high-order accurate nonoscillatory schemes. i. SIAM Journal on Numerical Analysis, 24(2):279–309, 1987.
- [10] Ami Harten, Stanley Osher, Björn Engquist, and Sukumar R Chakravarthy. Some results on uniformly high-order accurate essentially nonoscillatory schemes. Applied Numerical Mathematics, 2(3):347–377, 1986.
- [11] Ami Harten, Stanley Osher, Björn Engquist, and Sukumar R Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. Journal of Computational Physics, 71(2):231–303, 1987.
- [12] Dmitri Kuzmin. A high-resolution finite element scheme for convection-dominated transport. Communications in numerical methods in engineering, 16(3):215–223, 2000.
- [13] Dmitri Kuzmin, Rainald Löhner, and Stefan Turek. Flux-Corrected Transport: Principles, Algorithms, and Applications. Scientific Computation. Springer, 2005.
- [14] Dmitri Kuzmin and Friedhelm Schieweck. A parameter-free smoothness indicator for high-resolution finite element schemes. Central European Journal of Mathematics, 11(8):1478–1488, 2013.
- [15] Dmitri Kuzmin and Stefan Turek. Flux correction tools for finite elements. Journal of Computational Physics, 175(2):525–558, 2002.
- [16] Dmitri Kuzmin and Stefan Turek. High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. Journal of Computational Physics, 198(1):131–158, 2004.
- [17] Pierre Lesaint and Pierre-Arnaud Raviart. On a finite element method for solving the neutron transport equation. In C. de Boor, editor, Mathematical Aspects of Finite elements in Partial Differential Equations, pages 89–123. Academic Press, New York, 1974.
- [18] Xu-Dong Liu, Stanley Osher, and Tony Chan. Weighted essentially non-oscillatory schemes. Journal of Computational Physics, 115(1):200–212, 1994.
- [19] William H Reed and TR Hill. Triangular mesh methods for the neutron transport equation. Los Alamos Report LA-UR-73-479, 1973.
- [20] Steven T Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. Journal of Computational Physics, 31(3):335–362, 1979.

- [21] Xiangxiong Zhang and Chi-Wang Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. Journal of Computational Physics, 229(9):3091–3120, 2010.
- [22] Xiangxiong Zhang, Yinhua Xia, and Chi-Wang Shu. Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes. Journal of Scientific Computing, 50(1):29–62, 2012.