A MOMENT LIMITER FOR THE DISCONTINUOUS GALERKIN METHOD ON UNSTRUCTURED TRIANGULAR MESHES

ANDREW GIULIANI AND LILIA KRIVODONOVA

Abstract. We propose a second order limiter for the discontinuous Galerkin method applied to hyperbolic conservation laws. The limiter works by finding directions in which the solution coefficients can be separated and limits them independently of one another by comparing to forward and backward reconstructed differences. The limiter has a precomputed stencil of constant size, which provides computational advantages in terms of implementation and run time. We provide examples that demonstrate stability and second order accuracy of solutions.

Controlling oscillations near solution discontinuities in an accurate, robust, and efficient manner is one of the challenges for high order methods for solution of hyperbolic conservation laws in higher dimensions. In one dimension, the issue has been largely resolved for second order methods [1, 2, 3, 4], where the main tool to achieve non-oscillatory solutions is the enforcement of a total variation diminishing (TVD) property. Designing a limiter is more difficult in higher dimensions than in one dimension. This is because the number of derivatives that must be controlled increases, especially for high order methods, as mixed derivatives appear. Further, it is not evident in which direction the solution must be limited and there is not a unique way of defining which elements are neighbors. An attempt at defining the TVD property on Cartesian grids led to schemes of at most first order accuracy [5]. Another approach for ensuring stability is the enforcement of a local maximum principle (LMP) on the numerical solution. The numerical solution satisfies the local maximum principle in the means if

$$\min_{j\in V_i} \overline{U}_j^n \leq \overline{U}_i^{n+1} \leq \max_{j\in V_i} \overline{U}_j^n, \tag{1}$$

where V_i is a set containing the index of Ω_i and the indices of elements neighboring Ω_i, and $\overline{U}_i^n$, $\overline{U}_j^n$ are cell averages. We will focus on this method of controlling oscillations in this work.

In the context of finite volume (FV) methods, multidimensional limiting was used in [6] by scaling the x and y-components of a reconstructed gradient by a constant multiplier. In [7], a less diffusive method was proposed whereby the components of the gradient were scaled separately by different constant multipliers. These multipliers were determined by solving a small linear program on each element. Multislope FV

methods have been studied whereby the numerical solution's directional derivatives are reconstructed and limited separately on each face of an element [8, 9].

Limiters from the FV framework can sometimes be modified to act on DG solutions, though there are limiters devised to stabilize DG solutions specifically [10, 11]. These limiters compare the DG solution values on the edges to values reconstructed from averages on neighboring elements. In this work, we take a different approach.

The limiter that we present here can be viewed as a first step in the generalization of the moment limiter in [12, 13] to unstructured meshes, or as a standalone second order limiter with proven stability and accuracy properties. We start by noting that the second order DG solution is written in terms of an orthonormal basis that contains a constant function and two linear functions. We find two directions in which the directional derivative of the solution is proportional to either of the two solution coefficients corresponding to the non-constant basis functions. Each separated coefficient can be limited independently from the other by comparing it to a suitably reconstructed approximation to this directional derivative, as opposed to scaling them both by a constant multiplier [6].

The result of this analysis is a limiter on two-dimensional unstructured meshes that is composed of two independent one-dimensional limiters. The implementation of the limiter is straightforward as it uses the minmod function to compare the solution coefficients to suitable forward and backward differences. The mesh preprocessing stage determines the directional derivatives to be limited and the neighboring elements involved in the reconstruction. The stability analysis provides a set of constraints on the solution coefficients, i.e., a set of inequalities. Finding the optimal limited solution satisfying these constraints will result in the least diffusive limiter, but would be computationally costly. Instead, we derive a simplified region in the space of limiting coefficients that ensures that the numerical solution is second order accurate and that it satisfies the local maximum principle, similar to the Sweby's second order TVD region [1]. From this region, we choose a limiter that is easy to code and compute, as opposed to finding the least diffusive limiter.

We derive a local time step restriction for which application of the limiter guarantees that the cell averages remain within a locally defined interval for one forward Euler time step. This can be extended to high order time stepping, i.e., strong stability preserving (SSP) Runge Kutta methods. The derived CFL number is larger than the one needed for linear stability [14] and we show that using the time step restriction derived from the LMP can lead to stable but inaccurate solutions.

We now present the DG method along with our limiter.

1. Discontinuous Galerkin method

Two dimensional hyperbolic conservation laws are partial differential equations (PDEs) of the form

$$\mathbf{u}_t + \nabla \cdot \mathbf{F}(\mathbf{u}) = 0, \tag{2}$$

where the solution $\mathbf{u}(\mathbf{x}, t) = (u_1, u_2, ..., u_M)^\intercal$ is defined on $\Omega \times [0, T]$, T is the final time, $\mathbf{x} = (x, y)$ is the spatial coordinate such that $\mathbf{x} \in \Omega \subset \mathbb{R}^2$, and $\mathbf{F}(\mathbf{u}) = (\mathbf{F}_1(\mathbf{u}), \mathbf{F}_2(\mathbf{u}))$ is the flux function. The initial condition

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x})$$

is provided along with suitable boundary conditions.

The discontinuous Galerkin method can be formulated by first dividing the domain Ω into an unstructured mesh of triangles such that $\Omega = \bigcup_i \Omega_i$. The weak form of the conservation law is obtained by multiplying (2) by a test function $v \in H^1(\Omega_i)$ and integrating on element Ω_i. After applying the divergence theorem, we obtain

$$\int_{\Omega_i} \mathbf{u}_t v d\mathbf{x} - \int_{\Omega_i} \mathbf{F}(\mathbf{u}) \cdot \nabla v d\mathbf{x} + \int_{\partial\Omega_i} v \mathbf{F}(\mathbf{u}) \cdot \mathbf{n} dl = 0, \ \forall v \in H^1(\Omega_i), \tag{3}$$

where $\mathbf{n}$ is the unit outward facing normal on the element's boundary $\partial\Omega_i$.

Each element Ω_i is mapped to the canonical triangle Ω_0, having vertices at $(0, 0), (1, 0), (0, 1)$, using the transformation

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x_{i,1} & x_{i,2} & x_{i,3} \\ y_{i,1} & y_{i,2} & y_{i,3} \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 - r - s \\ r \\ s \end{pmatrix}, \tag{4}$$

where $(x_i, y_i)_{1,2,3}$ are the vertices of Ω_i in physical space. We label the edge defined by $(0, 0)$ and $(1, 0)$ of the canonical triangle edge 1, $(1, 0)$ and $(0, 1)$ edge 2, and $(0, 1)$ and $(0, 0)$ edge 3 (Figure 4). The Jacobian of the transformation is

$$J_i = \begin{pmatrix} x_{i,2} - x_{i,1} & x_{i,3} - x_{i,1} \\ y_{i,2} - y_{i,1} & y_{i,3} - y_{i,1} \end{pmatrix}. \tag{5}$$

We define $S(\Omega_0)$ to be the space of linear polynomials on Ω_0 and $\{\varphi_k\}_{k=0,1,2}$ to be a set of orthonormal basis functions on $S(\Omega_0)$:

$$\begin{aligned}\varphi_0 &= \sqrt{2},\\ \varphi_1 &= -2 + 6r,\\ \varphi_2 &= -2\sqrt{3} + 2\sqrt{3}r + 4\sqrt{3}s.\end{aligned} \tag{6}$$

The exact solution on element Ω_i is approximated by $\mathbf{U}_i$, which is a linear combination of the basis functions φ_k, i.e. $\mathbf{U}_i = \sum_{k=0}^{2} \mathbf{c}_{i,k}\varphi_k$, where the degrees of freedom (DOFs) are $\mathbf{c}_{i,k} = [c_{i,k}^1, c_{i,k}^2, \ldots, c_{i,k}^M]^\intercal$. As continuity between elements is not imposed, the solution is multivalued in the boundary integral. We therefore introduce a numerical flux $\mathbf{F}^*(\mathbf{U}_i, \mathbf{U}_j)$ to allow information exchange between adjacent cells Ω_i and Ω_j. We assume that the numerical flux is consistent, monotone, and differentiable. With v chosen to be φ_k, equation (3) now becomes

$$\frac{d}{dt}\mathbf{c}_{i,k} = \frac{1}{\det J_i}\int_{\Omega_0} \mathbf{F}(\mathbf{U}_i)\cdot(J_i^{-1}\nabla)\varphi_k \det J_i \, d\mathbf{r} - \frac{1}{\det J_i}\sum_{j\in N_i^e}\int_{\partial\Omega_{i,j}} \varphi_k \mathbf{F}^*(\mathbf{U}_i,\mathbf{U}_j)\cdot\mathbf{n}_{i,j}\, dl, \quad k = 0,1,2, \tag{7}$$

where N_i^e is the set of indices of elements sharing an edge with Ω_i, $\partial\Omega_{i,j}$ is the edge shared by Ω_i and Ω_j and $\mathbf{n}_{i,j}$ is the outward pointing unit normal on that edge. We propagate (7) in time using an explicit two-stage second order Runge-Kutta (RK) method, known as Heun's method.

Let us consider the case where (2) is a scalar conservation law. With $k = 0$, (7) becomes

$$\frac{d}{dt}c_{i,0} = -\frac{1}{\det J_i}\sum_{j\in N_i^e}\int_{\partial\Omega_{i,j}} \varphi_0 \mathbf{F}^*(U_i,U_j)\cdot\mathbf{n}_{i,j}\, dl. \tag{8}$$

Multiplying the above by $\varphi_0 = \sqrt{2}$ and recognizing that the cell average of U_i is $\overline{U}_i = c_{i,0}\varphi_0$, we obtain

$$\frac{d}{dt}\overline{U}_i = -\frac{1}{|\Omega_i|}\sum_{j\in N_i^e}\int_{\partial\Omega_{i,j}} \mathbf{F}^*(U_i,U_j)\cdot\mathbf{n}_{i,j}\, dl, \tag{9}$$

where $|\Omega_i|$ is the area of the cell and $\det J_i = 2|\Omega_i|$. This is an equation for the propagation of the solution average on Ω_i in time. We apply one forward Euler time step to (9) to obtain

$$\overline{U}_i^{n+1} = \overline{U}_i^n - \frac{\Delta t}{|\Omega_i|}\sum_{j\in N_i^e}\int_{\partial\Omega_{i,j}} \mathbf{F}^*(U_i^n,U_j^n)\cdot\mathbf{n}_{i,j}\, dl. \tag{10}$$

For nonlinear fluxes, the DG method requires a quadrature rule of order at least three to preserve the accuracy of the scheme [11]. An efficient choice is the two-point Gauss-Legendre quadrature rule, with $\mathbf{x}_{i,j,q}$ being the qth quadrature point on the edge shared by Ω_i and Ω_j. Replacing the boundary integral in

(10) with the quadrature rule gives

$$\overline{U}_i^{n+1} = \overline{U}_i^n - \sum_{j\in N_i^e} \frac{1}{2}\Delta t \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} \sum_{q=1,2} \mathbf{F}^*(U_i^n(\mathbf{x}_{i,j,q}), U_j^n(\mathbf{x}_{i,j,q}))\cdot \mathbf{n}_{i,j}. \tag{11}$$

For a linear flux, this becomes

$$\overline{U}_i^{n+1} = \overline{U}_i^n - \sum_{j\in N_i^e} \Delta t \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} \mathbf{F}^*(U_i^n(\mathbf{x}_{i,j}), U_j^n(\mathbf{x}_{i,j}))\cdot \mathbf{n}_{i,j}, \tag{12}$$

where $\mathbf{x}_{i,j}$ is the midpoint of the edge shared by Ω_i and Ω_j.

2. Limiting algorithm

In order to enforce the local maximum principle (1), we apply a limiter to the solution coefficients $c_{i,1}^n$ and $c_{i,2}^n$. We consider the directional derivative of $U_i^n(\mathbf{r})$ in the direction of the unit vector $\mathbf{w}$, in the canonical coordinate system $\mathbf{r} = (r, s)$

$$D_{\mathbf{w}} U_i^n(\mathbf{r}) = \nabla_{rs} U_i^n \cdot \mathbf{w} = \left(c_{i,1}^n \nabla_{rs}\varphi_1 + c_{i,2}^n \nabla_{rs}\varphi_2\right)\cdot \mathbf{w}.$$

Computing the gradient of the basis functions φ_1 and φ_2 in (6), yields

$$D_{\mathbf{w}} U_i^n(\mathbf{r}) = \left(c_{i,1}^n \left(6\ ,\ 0\right) + c_{i,2}^n \left(2\sqrt{3}\ ,\ 4\sqrt{3}\right)\right)\cdot \mathbf{w}. \tag{13}$$

In the directions $\mathbf{w}_1 = \frac{2}{\sqrt{5}}\left(1, -\frac{1}{2}\right)$ and $\mathbf{w}_2 = (0, 1)$, the directional derivatives are

$$D_{\mathbf{w}_1} U_i^n = 6\left(\frac{2}{\sqrt{5}}\right) c_{i,1}^n \quad \text{and} \quad D_{\mathbf{w}_2} U_i^n = 4\sqrt{3} c_{i,2}^n.$$

They depend on either $c_{i,1}^n$, or $c_{i,2}^n$, i.e. we have found the directions in which the DOFs are uncoupled. This will allow us to limit each solution coefficient separately by comparing them to forward and backward approximations of the derivatives in the directions $\mathbf{w}_1$ and $\mathbf{w}_2$.

Using (4), we map $\mathbf{w}$ into the physical space and normalize to obtain on Ω_i

$$\mathbf{v}_i = \frac{J_i \mathbf{w}}{||J_i \mathbf{w}||}.$$

For $\mathbf{w}_1$ and $\mathbf{w}_2$ in canonical coordinates, the corresponding vectors in the physical space are $\mathbf{v}_{i,1}$ and $\mathbf{v}_{i,2}$, and the directional derivatives are

$$D_{\mathbf{v}_{i,1}} U_i^n = c_{i,1}^n \frac{6}{\frac{\sqrt{5}}{2}||J_i \mathbf{w}_1||} \quad \text{and} \quad D_{\mathbf{v}_{i,2}} U_i^n = c_{i,2}^n \frac{4\sqrt{3}}{||J_i \mathbf{w}_2||}.$$

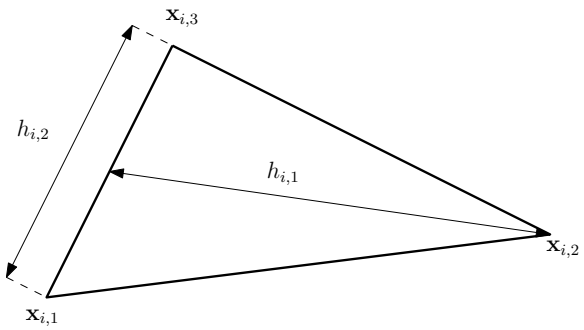


FIGURE 1. $h_{i,1}$ is the length of the segment connecting $\mathbf{x}_{i,2}$ and the midpoint of the edge defined by $\mathbf{x}_{i,1}$ and $\mathbf{x}_{i,3}$. $h_{i,2}$ is the length of the edge defined by $\mathbf{x}_{i,1}$ and $\mathbf{x}_{i,3}$.

Let $h_{i,1} = \frac{\sqrt{5}}{2}||J_i\mathbf{w}_1||$ and $h_{i,2} = ||J_i\mathbf{w}_2||$. From (5), it follows that $h_{i,1}$ is the distance between $\mathbf{x}_{i,2}$ and the midpoint of the opposite edge, and $h_{i,2}$ is the distance between $\mathbf{x}_{i,3}$ and $\mathbf{x}_{i,1}$ (Figure 1). In terms of derivatives $D_{\mathbf{v}_{i,1}}U_i^n$ and $D_{\mathbf{v}_{i,2}}U_i^n$, the solution coefficients $c_{i,1}^n$ and $c_{i,2}^n$ are written as

$$c_{i,1}^n = \frac{h_{i,1}}{6}D_{\mathbf{v}_{i,1}}U_i^n \quad \text{and} \quad c_{i,2}^n = \frac{h_{i,2}}{4\sqrt{3}}D_{\mathbf{v}_{i,2}}U_i^n. \tag{14}$$

We reconstruct the slopes of the numerical solution in these two directions using solution averages on neighboring elements. We start by compiling a list of all elements that share a vertex with Ω_i. We connect the centroids of the elements with linear segments to form a polygon, see Figure 2a, shaded region. We find the four points where this polygon is crossed by the lines with directions $\mathbf{v}_{i,1}$ and $\mathbf{v}_{i,2}$ that pass through the centroid of Ω_i. We name them $\mathbf{x}_{i,1}^b$, $\mathbf{x}_{i,2}^b$, and $\mathbf{x}_{i,1}^f$, $\mathbf{x}_{i,2}^f$, respectively (Figure 2a). Next, using linear interpolation, we reconstruct the values of the numerical solution at the forward and backward points of intersection. The reconstructed numerical solution in the forward and backward direction of $\mathbf{v}_{i,1}$ are $U_{i,1}^f$ and $U_{i,1}^b$. Likewise, in the forward and backward direction of $\mathbf{v}_{i,2}$, they are $U_{i,2}^f$ and $U_{i,2}^b$, respectively. For example, in Figure 2b the forward interpolated solution value $U_{i,1}^f$ is given by

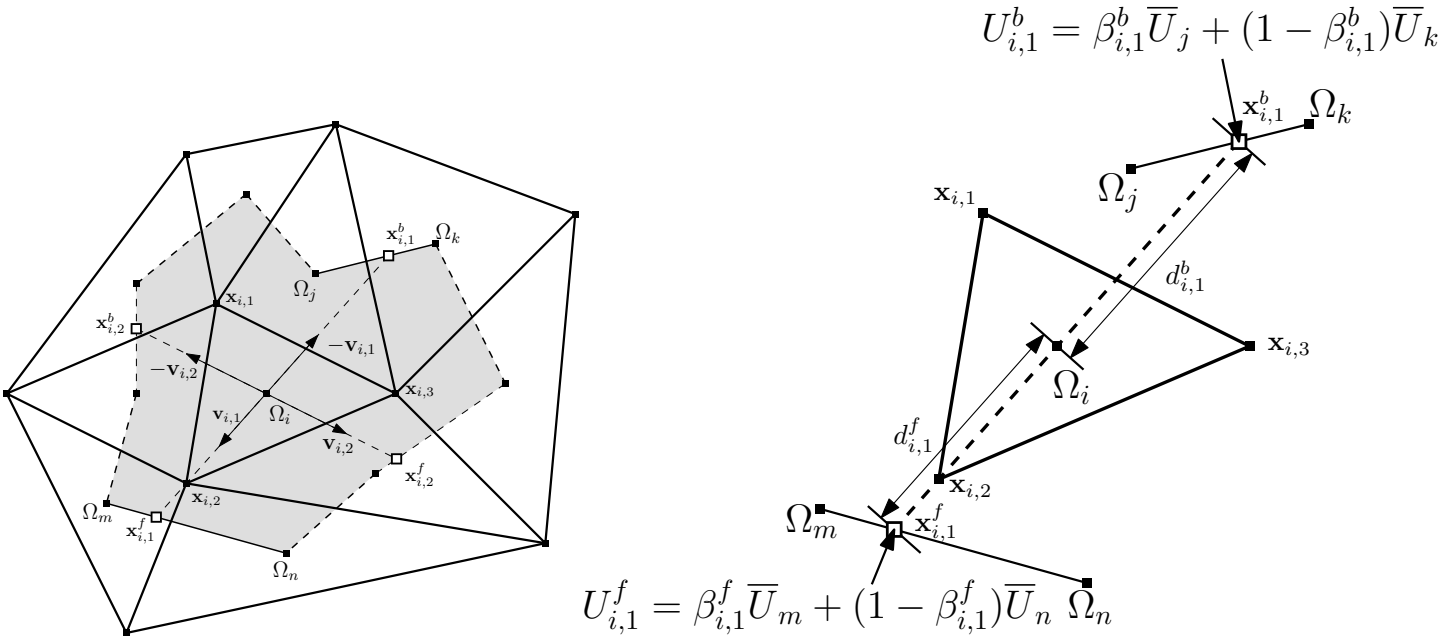
$$U_{i,1}^f = \beta_{i,1}^f\overline{U}_m^n + (1-\beta_{i,1}^f)\overline{U}_n \quad \text{with } 0 \le \beta_{i,1}^f \le 1.$$

The reconstructed forward differences $\Delta_{i,1}^f$ and $\Delta_{i,2}^f$ are defined as

$$\Delta_{i,1}^f = \frac{1}{d_{i,1}^f}(U_{i,1}^f - \overline{U}_i) \text{ and } \Delta_{i,2}^f = \frac{1}{d_{i,2}^f}(U_{i,2}^f - \overline{U}_i), \tag{15}$$

where $d_{i,1}^f$, $d_{i,2}^f$ are the distances from $\mathbf{x}_{i,1}^f$ and $\mathbf{x}_{i,2}^f$ to the cell centroid $\mathbf{x}_i$, respectively. Similarly, the backward differences $\Delta_{i,1}^b$ and $\Delta_{i,2}^b$ are

$$\Delta_{i,1}^b = \frac{1}{d_{i,1}^b}(\overline{U}_i - U_{i,1}^b) \text{ and } \Delta_{i,2}^b = \frac{1}{d_{i,2}^b}(\overline{U}_i - U_{i,2}^b), \tag{16}$$



(a) Reconstruction neighborhood. (b) Forward and backward reconstruction in the direction of $\mathbf{v}_{i,1}$.

FIGURE 2. Approximation of directional derivatives on $\Omega_i = (\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \mathbf{x}_{i,3})$.

where $d^b_{i,1}$, $d^b_{i,2}$ are the distances measured from $\mathbf{x}^b_{i,1}$ and $\mathbf{x}^b_{i,2}$ to the cell centroid $\mathbf{x}_i$, respectively. We limit $c^n_{i,1}$ by comparing $D_{\mathbf{v}_{i,1}}U^n_i$ to the reconstructed forward and backward differences. The same is done for $c^n_{i,2}$ and $D_{\mathbf{v}_{i,2}}U^n_i$. The limited degrees of freedom can be written in terms of the forward or backward differences

$$\tilde{c}^n_{i,1} = l^f_{i,1}\frac{h_{i,1}}{6}\Delta^f_{i,1} \quad \text{or} \quad \tilde{c}^n_{i,1} = l^b_{i,1}\frac{h_{i,1}}{6}\Delta^b_{i,1},$$
$$\tilde{c}^n_{i,2} = l^f_{i,2}\frac{h_{i,2}}{4\sqrt{3}}\Delta^f_{i,2} \quad \text{or} \quad \tilde{c}^n_{i,2} = l^b_{i,2}\frac{h_{i,2}}{4\sqrt{3}}\Delta^b_{i,1},$$

where $l^f_{i,k}$ and $l^b_{i,k}$ for $k = 1, 2$ are non-negative limiting coefficients for forward and backward differences that we will derive. Introducing $r_{i,1}$ and $r_{i,2}$, the ratios of the backward and forward differences,

$$r_{i,1} = \frac{\Delta^b_{i,1}}{\Delta^f_{i,1}} = \left(\frac{d^f_{i,1}}{d^b_{i,1}}\right)\frac{\overline{U}_i - U^b_{i,1}}{U^f_{i,1} - \overline{U}_i} \quad r_{i,2} = \frac{\Delta^b_{i,2}}{\Delta^f_{i,2}} = \left(\frac{d^f_{i,2}}{d^b_{i,2}}\right)\frac{\overline{U}_i - U^b_{i,2}}{U^f_{i,2} - \overline{U}_i}, \tag{17}$$

we express the limited degrees of freedom in terms of the forward differences

$$\tilde{c}^n_{i,1} = l_{i,1}\frac{h_{i,1}}{6}\Delta^f_{i,1},$$
$$\tilde{c}^n_{i,2} = l_{i,2}\frac{h_{i,2}}{4\sqrt{3}}\Delta^f_{i,2}, \tag{18}$$

with non-negative limiting coefficients $l_{i,1}$ and $l_{i,2}$.

We want to derive upper bounds on the limiting coefficients such that the maximum principle (1) is satisfied under a suitable time step restriction. To do so, we write the solution mean at t^{n+1}, $\overline{U}_i^{n+1}$, in the following form

$$\overline{U}_i^{n+1} = \overline{U}_i^n + \sum_j d_j(U_j - \overline{U}_i^n), \tag{19}$$

where U_j are understood to be solution means in the neighborhood of Ω_i or reconstructed solution values, e.g. $U_{i,1}^f$, $U_{i,1}^b$, $U_{i,2}^f$, $U_{i,2}^b$. Next, we show that under the limiter (18) and a time step restriction, we ensure that the coefficients d_j are non-negative and that the sum of the coefficients is less than or equal to 1. In summary, we need to prove

1. sum property:

$$\sum_j d_j \leq 1, \tag{20}$$

2. non-negativity property:

$$d_j \geq 0. \tag{21}$$

Properties (20) and (21) mean that $\overline{U}_i^{n+1}$ in (19) is a convex combination of solution values at time t^n and the local maximum principle (1) holds for one forward Euler time step. The result can be extended to higher order Runge-Kutta methods if they are strong stability preserving (SSP). SSP-RK methods are convex combinations of stages of forward Euler time steps. Since each forward Euler step produces an intermediate solution with cell averages that do not violate those of the initial condition, then a convex combination of forward Euler steps will not either [15].

We present analysis for when (2) is a linear equation. The nonlinear case closely follows the linear one and is presented in Appendix A.

3. Linear advection equation

We use the upwind numerical flux, which is given by

$$\mathbf{F}^*(U_i^n(\mathbf{x}_{i,j}), U_j^n(\mathbf{x}_{i,j})) \cdot \mathbf{n}_{i,j} = \begin{cases} (\mathbf{a} \cdot \mathbf{n}_{i,j}) U_j^n(\mathbf{x}_{i,j}) \text{ if } j \in N_i^-, \\ (\mathbf{a} \cdot \mathbf{n}_{i,j}) U_i^n(\mathbf{x}_{i,j}) \text{ if } j \in N_i^+, \end{cases}$$

where $\mathbf{a}$ is the flow direction, N_i^- and N_i^+ are the sets of inflow and outflow neighbors that share an edge with Ω_i, respectively, i.e. $N_i^\pm = \{j : j \in N_i^e \text{ such that } \pm \mathbf{a} \cdot \mathbf{n}_{i,j} > 0\}$. For a neighboring element Ω_j with

$\mathbf{a}\cdot\mathbf{n}_{i,j}=0$, the flux term does not contribute to the right hand side of (9), and this element can be omitted from both $N_i^{\pm}$. Therefore, the scheme (12) becomes

$$\overline{U}_i^{n+1}=\overline{U}_i^n+\Delta t\sum_{j\in N_i^-}|\mathbf{a}\cdot\mathbf{n}_{i,j}|\frac{|\partial\Omega_{i,j}|}{|\Omega_i|}U_j^n(\mathbf{x}_{i,j})-\Delta t\sum_{j\in N_i^+}|\mathbf{a}\cdot\mathbf{n}_{i,j}|\frac{|\partial\Omega_{i,j}|}{|\Omega_i|}U_i^n(\mathbf{x}_{i,j}). \tag{22}$$

By the divergence theorem, we have the following relation

$$\sum_{j\in N_i^e}|\partial\Omega_{i,j}|\mathbf{a}\cdot\mathbf{n}_{i,j}=0. \tag{23}$$

Using (23) in (22), we have

$$\overline{U}_i^{n+1}=\overline{U}_i^n+\Delta t\sum_{j\in N_i^-}|\mathbf{a}\cdot\mathbf{n}_{i,j}|\frac{|\partial\Omega_{i,j}|}{|\Omega_i|}(U_j^n(\mathbf{x}_{i,j})-\overline{U}_i^n)-\Delta t\sum_{j\in N_i^+}|\mathbf{a}\cdot\mathbf{n}_{i,j}|\frac{|\partial\Omega_{i,j}|}{|\Omega_i|}(U_i^n(\mathbf{x}_{i,j})-\overline{U}_i^n).$$

Introducing the coefficients

$$v_{j,i}^-=-\Delta t\mathbf{a}\cdot\mathbf{n}_{i,j}\frac{|\partial\Omega_{i,j}|}{|\Omega_i|}\quad\text{and}\quad v_{i,j}^+=\Delta t\mathbf{a}\cdot\mathbf{n}_{i,j}\frac{|\partial\Omega_{i,j}|}{|\Omega_i|}, \tag{24}$$

we write the linear scheme as

$$\overline{U}_i^{n+1}=\overline{U}_i^n+\sum_{j\in N_i^-}v_{j,i}^-(U_j^n(\mathbf{x}_{i,j})-\overline{U}_i^n)-\sum_{j\in N_i^+}v_{i,j}^+(U_i^n(\mathbf{x}_{i,j})-\overline{U}_i^n). \tag{25}$$

Limiting the numerical solution at the time t^n gives

$$\overline{U}_i^{n+1}=\overline{U}_i^n+\sum_{j\in N_i^-}v_{j,i}^-(\tilde{U}_j^n(\mathbf{x}_{i,j})-\overline{U}_i^n)-\sum_{j\in N_i^+}v_{i,j}^+(\tilde{U}_i^n(\mathbf{x}_{i,j})-\overline{U}_i^n), \tag{26}$$

where $\tilde{U}_i^n$ and $\tilde{U}_j^n$ are the limited numerical solutions on Ω_i and Ω_j, respectively. We aim to rewrite the inflow terms in (26) for each $j\in N_i^-$ in the form

$$\tilde{U}_j^n(\mathbf{x}_{i,j})-\overline{U}_i^n=f_{j,i}(\overline{U}_j^n-\overline{U}_i^n)+f_{j,i,1}(U_{j,i,1}^--\overline{U}_i^n)+f_{j,i,2}(U_{j,i,2}^--\overline{U}_i^n), \tag{27}$$

where $U_{j,i,1}^-$ and $U_{j,i,2}^-$ are reconstructed solution values in the forward or backward directions $\pm\mathbf{v}_{i,1}$ and $\pm\mathbf{v}_{i,2}$, respectively, and $f_{j,i}$, $f_{j,i,1}$, and $f_{j,i,2}$ are non-negative constants. Likewise, we aim to rewrite the outflow terms for each $j\in N_i^+$ in (26) in the form

$$\tilde{U}_i^n(\mathbf{x}_{i,j})-\overline{U}_i^n=-\left[g_{i,j,1}(U_{i,j,1}^+-\overline{U}_i^n)+g_{i,j,2}(U_{i,j,2}^+-\overline{U}_i^n)\right], \tag{28}$$

where $U^+_{j,i,1}$ and $U^+_{j,i,2}$ are reconstructed solution values in the forward or backward directions $\pm\mathbf{v}_{i,1}$ and $\pm\mathbf{v}_{i,2}$, respectively, and $g_{i,j,1}$ and $g_{i,j,2}$ are non-negative constants.

3.1. **Inflow term.** We start with the inflow term in (26). Consider the limited numerical solution on cell Ω_j, $j \in N_i^-$, an inflow neighbor of Ω_i, at $\mathbf{x}_{i,j} = \mathbf{x}(\mathbf{r}_{j,i})$, where $\mathbf{r}_{j,i}$ is the quadrature point on Ω_0

$$\tilde{U}^n_j(\mathbf{x}(\mathbf{r}_{j,i})) = \overline{U}^n_j + \tilde{c}^n_{j,1}\varphi_1(\mathbf{r}_{j,i}) + \tilde{c}^n_{j,2}\varphi_2(\mathbf{r}_{j,i}).$$

Note that the physical point $\mathbf{x}_{i,j}$ might be mapped using (4) to different edges of the canonical triangle Ω_0 by Ω_i and Ω_j. On Ω_j, $\mathbf{x}_{i,j}$ is mapped to $\mathbf{r}_{j,i} \in \Omega_0$, while the same point on Ω_i is mapped to $\mathbf{r}_{i,j} \in \Omega_0$. Using (15) and (18), the limited solution can be rewritten in terms of the forward differences

$$\tilde{U}^n_j(\mathbf{x}(\mathbf{r}_{j,i})) = \overline{U}^n_j + l_{j,1}\varphi_1(\mathbf{r}_{j,i})\frac{h_{j,1}}{6}\frac{1}{d^f_{j,1}}(U^f_{j,1} - \overline{U}^n_j) + l_{j,2}\varphi_2(\mathbf{r}_{j,i})\frac{h_{j,2}}{4\sqrt{3}}\frac{1}{d^f_{j,2}}(U^f_{j,2} - \overline{U}^n_j). \tag{29}$$

Consider the second term in the right hand side of (29). To satisfy the non-negativity property (21), we require that the multiplier of the difference $U^f_{j,1} - \overline{U}^n_j$ be non-negative. If $\varphi_1(\mathbf{r}_{j,i}) \geq 0$, then this requirement is satisfied. Otherwise, using (17), we replace the forward difference with the backward difference to obtain

$$\varphi_1(\mathbf{r}_{j,i})\frac{1}{d^f_{j,1}}(U^f_{j,1} - \overline{U}^n_j) = \varphi_1(\mathbf{r}_{j,i})\frac{1}{r_{j,1}d^b_{j,1}}(\overline{U}^n_j - U^b_{j,1}) = |\varphi_1(\mathbf{r}_{j,i})|\frac{1}{r_{j,1}d^b_{j,1}}(U^b_{j,1} - \overline{U}^n_j).$$

This results in a non-negative multiplier in front of $U^b_{j,1} - \overline{U}^n_j$, if the forward and backward differences are of the same sign. If the differences are of opposite sign, then $l_{j,1}$ is zero and, consequently, $f_{j,i,1}$ is equal to zero. We introduce the following notation for convenience

$$\alpha^-_{j,i,1} = \begin{cases} \frac{h_{j,1}}{6}\frac{\varphi_1(\mathbf{r}_{j,i})}{d^f_{j,1}} \text{ if } \varphi_1(\mathbf{r}_{j,i}) \geq 0, \\ \frac{h_{j,1}}{6}\frac{|\varphi_1(\mathbf{r}_{j,i})|}{r_{j,1}d^b_{j,1}} \text{ otherwise,} \end{cases} \quad \text{and} \quad U^-_{j,i,1} = \begin{cases} U^f_{j,1} \text{ if } \varphi_1(\mathbf{r}_{j,i}) \geq 0, \\ U^b_{j,1} \text{ otherwise.} \end{cases} \tag{30}$$

Similarly, for the last term on the right hand side of (29) we introduce

$$\alpha^-_{j,i,2} = \begin{cases} \frac{h_{j,2}}{4\sqrt{3}}\frac{\varphi_2(\mathbf{r}_{j,i})}{d^f_{j,2}} \text{ if } \varphi_2(\mathbf{r}_{j,i}) \geq 0, \\ \frac{h_{j,2}}{4\sqrt{3}}\frac{|\varphi_2(\mathbf{r}_{j,i})|}{r_{j,2}d^b_{j,2}} \text{ otherwise,} \end{cases} \quad \text{and} \quad U^-_{j,i,2} = \begin{cases} U^f_{j,2} \text{ if } \varphi_2(\mathbf{r}_{j,i}) \geq 0, \\ U^b_{j,2} \text{ otherwise.} \end{cases} \tag{31}$$

Using (30) and (31) in (29), we obtain

$$\tilde{U}^n_j(\mathbf{x}_{i,j}) = \overline{U}^n_j + l_{j,1}\alpha^-_{j,i,1}(U^-_{j,i,1} - \overline{U}^n_j) + l_{j,2}\alpha^-_{j,i,2}(U^-_{j,i,2} - \overline{U}^n_j). \tag{32}$$

Subtracting $\overline{U}_i^n$ from both sides of (32), then adding and subtracting $\overline{U}_i^n$ in the last two terms on the right, we have

$$\tilde{U}_j^n(\mathbf{x}_{i,j}) - \overline{U}_i^n = \overline{U}_j^n - \overline{U}_i^n + l_{j,1}\alpha_{j,i,1}^-(U_{j,i,1}^- - \overline{U}_i^n + \overline{U}_i^n - \overline{U}_j^n) + l_{j,2}\alpha_{j,i,2}^-(U_{j,i,2}^- - \overline{U}_i^n + \overline{U}_i^n - \overline{U}_j^n),$$

which gives

$$\tilde{U}_j^n(\mathbf{x}_{i,j}) - \overline{U}_i^n = (1 - l_{j,1}\alpha_{j,i,1}^- - l_{j,2}\alpha_{j,i,2}^-)(\overline{U}_j^n - \overline{U}_i^n) + l_{j,1}\alpha_{j,i,1}^-(U_{j,i,1}^- - \overline{U}_i^n) + l_{j,2}\alpha_{j,i,2}^-(U_{j,i,2}^- - \overline{U}_i^n). \tag{33}$$

Thus, (33) is in the form of (27), with

$$f_{j,i} = 1 - l_{j,1}\alpha_{j,i,1}^- - l_{j,2}\alpha_{j,i,2}^-,$$
$$f_{j,i,1} = l_{j,1}\alpha_{j,i,1}^-,$$
$$f_{j,i,2} = l_{j,2}\alpha_{j,i,2}^-.$$

Sum and non-negativity. The coefficients $f_{j,i,1}$ and $f_{j,i,2}$ are non-negative by (30) and (31). Requiring the coefficient $f_{j,i}$ to be non-negative gives the following condition

$$1 - l_{j,1}\alpha_{j,i,1}^- - l_{j,2}\alpha_{j,i,2}^- \geq 0 \quad \forall j \in N_i^-. \tag{34}$$

Note that (34) imposes a restriction on the neighboring inflow element Ω_j, rather than on Ω_i itself. The sum of the coefficients over the inflow edge is

$$f_{j,i} + f_{j,i,1} + f_{j,i,2} = l_{j,1}\alpha_{j,i,1}^- + l_{j,2}\alpha_{j,i,2}^- + (1 - l_{j,1}\alpha_{j,i,1}^- - l_{j,2}\alpha_{j,i,2}^-) = 1. \tag{35}$$

3.2. **Outflow term.** We now deal with the outflow term in (26). Consider the limited numerical solution on cell Ω_i at the quadrature point $\mathbf{x}_{i,j} = \mathbf{x}(\mathbf{r}_{i,j})$, $j \in N_i^+$,

$$\tilde{U}_i^n(\mathbf{x}(\mathbf{r}_{i,j})) = \overline{U}_i^n + \tilde{c}_{i,1}^n\varphi_1(\mathbf{r}_{i,j}) + \tilde{c}_{i,2}^n\varphi_2(\mathbf{r}_{i,j}).$$

Using (15) and (18), the limited solution can be rewritten in terms of the forward differences

$$\tilde{U}_i^n(\mathbf{x}(\mathbf{r}_{i,j})) - \overline{U}_i^n = l_{i,1}\varphi_{i,1}(\mathbf{r}_{i,j})\frac{h_{i,1}}{6}\frac{1}{d_{i,1}^f}(U_{i,1}^f - \overline{U}_i^n) + l_{i,2}\varphi_{i,2}(\mathbf{r}_{i,j})\frac{h_{i,2}}{4\sqrt{3}}\frac{1}{d_{i,2}^f}(U_{i,2}^f - \overline{U}_i^n). \tag{36}$$

As for the inflow term, we introduce the following notation for the first term in the right hand side of (36)

$$\alpha_{i,1}^{+} = \begin{cases} \frac{h_{i,1}}{6} \frac{|\varphi_1(\mathbf{r}_{i,j})|}{d_{i,1}^{f}} \text{ if } \varphi_1(\mathbf{r}_{i,j}) \leq 0, \\ \frac{h_{i,1}}{6} \frac{\varphi_1(\mathbf{r}_{i,j})}{r_{i,1} d_{i,1}^{b}} \text{ otherwise,} \end{cases} \quad \text{and } U_{i,j,1}^{+} = \begin{cases} U_{i,1}^{f} \text{ if } \varphi_1(\mathbf{r}_{i,j}) \leq 0 \\ U_{i,1}^{b} \text{ otherwise,} \end{cases} \tag{37}$$

and for the second term in the right hand side of (36)

$$\alpha_{i,2}^{+} = \begin{cases} \frac{h_{i,2}}{4\sqrt{3}} \frac{|\varphi_2(\mathbf{r}_{i,j})|}{d_{i,2}^{f}} \text{ if } \varphi_2(\mathbf{r}_{i,j}) \leq 0 \\ \frac{h_{i,2}}{4\sqrt{3}} \frac{\varphi_2(\mathbf{r}_{i,j})}{r_{i,2} d_{i,2}^{b}} \text{ otherwise.} \end{cases} \quad \text{and } \; U_{i,j,2}^{+} = \begin{cases} U_{i,2}^{f} \text{ if } \varphi_2(\mathbf{r}_{i,j}) \leq 0 \\ U_{i,2}^{b} \text{ otherwise.} \end{cases} \tag{38}$$

Therefore, (36) becomes

$$\tilde{U}_i^n(\mathbf{x}_{i,j}) - \overline{U}_i^n = -\left[l_{i,1} \alpha_{i,j,1}^{+} (U_{i,j,1}^{+} - \overline{U}_i^n) + l_{i,2} \alpha_{i,j,2}^{+} (U_{i,j,2}^{+} - \overline{U}_i^n) \right]. \tag{39}$$

This is of the form (28) with $g_{i,j,1} = l_{i,1}\alpha_{i,j,1}^{+}$ and $g_{i,j,2} = l_{i,2}\alpha_{i,j,2}^{+}$. As in the inflow case, $l_{i,1}$ or $l_{i,2}$ are zero when backward and forward differences are of opposite sign.

Sum and non-negativity. The multipliers $g_{i,j,1}$ and $g_{i,j,2}$ are non-negative by (37) and (38). The sum of the coefficients is given by

$$g_{i,j,1} + g_{i,j,2} = l_{i,1}\alpha_{i,j,1}^{+} + l_{i,2}\alpha_{i,j,2}^{+}. \tag{40}$$

3.3. **Putting it all together.** The inflow and outflow terms have been expanded into sums of the form (27) and (28) in (33) and (39), respectively. Substituting these sums into (26) gives

$$\begin{aligned} \overline{U}_i^{n+1} = \overline{U}_i^n &+ \sum_{j \in N_i^-} v_{j,i}^{-} \left[f_{j,i}(\overline{U}_j^n - \overline{U}_i^n) + f_{j,i,1}(U_{j,i,1}^{-} - \overline{U}_i^n) + f_{j,i,2}(U_{j,i,2}^{-} - \overline{U}_i^n) \right] \\ &+ \sum_{j \in N_i^+} v_{i,j}^{+} \left[g_{i,j,1}(U_{i,j,1}^{+} - \overline{U}_i^n) + g_{i,j,2}(U_{i,j,2}^{+} - \overline{U}_i^n) \right]. \end{aligned}$$

This is of the form (19). We require that the sum of coefficients in front of the differences above is less than or equal to one. Using (35) and (40), we write this requirement as

$$\sum_{j \in N_i^-} v_{j,i}^{-} \left[f_{j,i} + f_{j,i,1} + f_{j,i,2} \right] + \sum_{j \in N_i^+} v_{i,j}^{+} \left[g_{i,j,1} + g_{i,j,2} \right] = \sum_{j \in N_i^-} v_{j,i}^{-} + \sum_{j \in N_i^+} v_{i,j}^{+} \left(l_{i,1}\alpha_{i,j,1}^{+} + l_{i,2}\alpha_{i,j,2}^{+} \right) \leq 1. \tag{41}$$

For (41) to be satisfied, we enforce on each outflow edge of Ω_i

$$l_{i,1}\alpha_{i,j,1}^{+} + l_{i,2}\alpha_{i,j,2}^{+} \leq 1, \quad \forall j \in N_i^+, \tag{42}$$

and on all elements

$$\sum_{j\in N_i^-} v_{j,i}^- + \sum_{j\in N_i^+} v_{i,j}^+ \leq 1, \quad \forall \Omega_i. \tag{43}$$

For non-negativity of the expansion coefficient $f_{j,i}$ we must enforce the constraint (34) on the inflows edges of Ω_i, i.e. on the outflow edges of elements sharing an edge with Ω_i,

$$l_{j,1}\alpha_{j,i,1}^- + l_{j,2}\alpha_{j,i,2}^- \leq 1, \quad \forall j \in N_i^-. \tag{44}$$

3.4. **Time step restriction.** Using the definition of $v_{j,i}^-$ and $v_{i,j}^+$ in (24), (43) becomes

$$\Delta t \left(\sum_{j\in N_i^-} |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} + \sum_{j\in N_i^+} |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} \right) \leq 1, \tag{45}$$

since $\mathbf{a}\cdot\mathbf{n}_{i,j} < 0$ for $j \in N_i^-$ and $\mathbf{a}\cdot\mathbf{n}_{i,j} > 0$ for $j \in N_i^+$. From (23), we obtain

$$-\sum_{j\in N_i^-} |\partial\Omega_{i,j}| \mathbf{a}\cdot\mathbf{n}_{i,j} = \sum_{j\in N_i^+} |\partial\Omega_{i,j}| \mathbf{a}\cdot\mathbf{n}_{i,j}.$$

Because $\mathbf{a}\cdot\mathbf{n}_{i,j} < 0$ for $j \in N_i^-$ and $\mathbf{a}\cdot\mathbf{n}_{i,j} > 0$ for $j \in N_i^+$, this becomes

$$\sum_{j\in N_i^-} |\partial\Omega_{i,j}||\mathbf{a}\cdot\mathbf{n}_{i,j}| = \sum_{j\in N_i^+} |\partial\Omega_{i,j}||\mathbf{a}\cdot\mathbf{n}_{i,j}|. \tag{46}$$

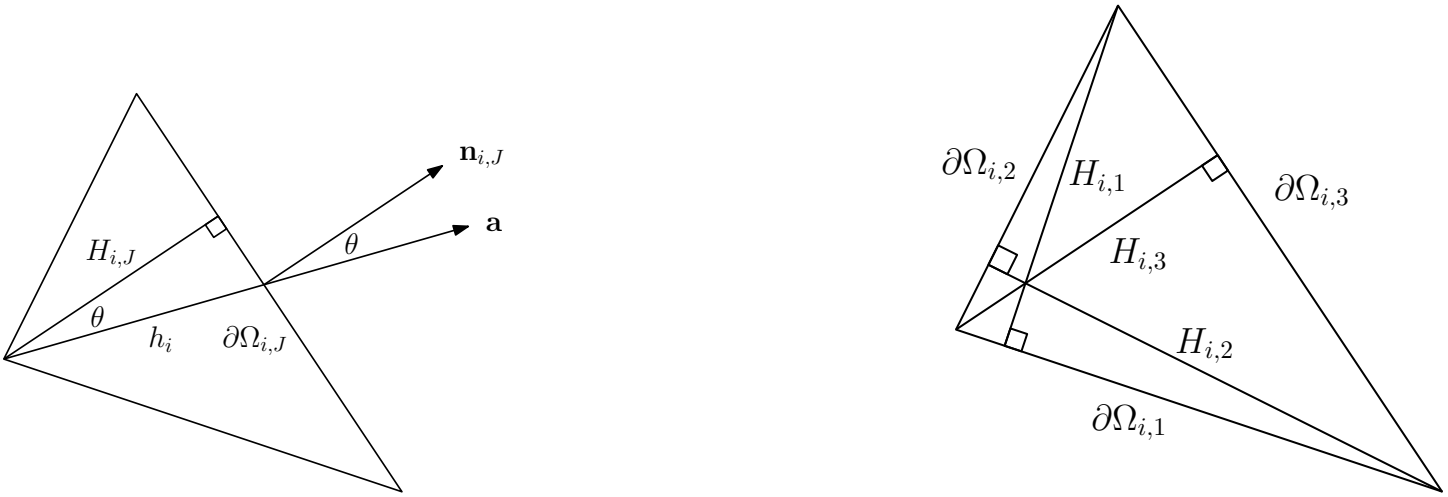
If $\mathbf{a}$ is not parallel to one of the edges, Ω_i has either a single inflow edge or a single outflow edge, which we will refer to as $\partial\Omega_{i,j}$. Otherwise, either the inflow or outflow edge can be chosen to be $\partial\Omega_{i,J}$. In terms of $\partial\Omega_{i,J}$, identity (46) now becomes

$$|\partial\Omega_{i,J}||\mathbf{a}\cdot\mathbf{n}_{i,J}| = \sum_{j\in N_i^-} |\partial\Omega_{i,j}||\mathbf{a}\cdot\mathbf{n}_{i,j}| = \sum_{j\in N_i^+} |\partial\Omega_{i,j}||\mathbf{a}\cdot\mathbf{n}_{i,j}|.$$

Using the above in (45), we obtain

$$2\Delta t \frac{|\partial\Omega_{i,J}||\mathbf{a}\cdot\mathbf{n}_{i,J}|}{|\Omega_i|} \leq 1. \tag{47}$$

The area of the cell Ω_i is $\frac{1}{2}|\partial\Omega_{i,j}|H_{i,J}$, where $H_{i,J}$ is the height of the cell measured from the edge $\partial\Omega_{i,j}$ (Figure 3a). Further, a simple geometric consideration reveals that $||\mathbf{a}||H_{i,J} = h_i|\mathbf{a}\cdot\mathbf{n}_J|$, where h_i is the



(a) Cell size in the direction of flow h_i.

(b) Cell size for systems $h_i = \min(H_{i,1}, H_{i,2}, H_{i,3})$.

FIGURE 3. Measures of cell size for time step restriction (49).

width of the cell in the direction of $\mathbf{a}$. Using this, (47) simplifies to

$$2\Delta t \frac{|\partial\Omega_{i,J}||\mathbf{a}\cdot\mathbf{n}_{i,J}|}{|\Omega_i|} = 4\Delta t \frac{||\mathbf{a}||}{h_i} \leq 1. \tag{48}$$

The time step restriction on the entire mesh is then given by

$$\Delta t \leq \frac{1}{4}\min_i \frac{h_i}{||\mathbf{a}||}. \tag{49}$$

For systems of equations, in general, information can be propagated along multiple directions. Therefore, we take the smallest cell width, i.e.,

$$h_i = \min(H_{i,1}, H_{i,2}, H_{i,3}), \tag{50}$$

where $H_{i,1}$, $H_{i,2}$, and $H_{i,3}$ are the cell widths perpendicular to the three edges of the element, e_1, e_2, and e_3 (Figure 3b).

3.5. **Moment limiter.** By the analysis in Sections 3.1-3.3, there are two constraints on the slope of $U_i^n(\mathbf{x})$ for each outflow edge of Ω_i and no constraints on the inflow edges. One constraint is given by (42) and the other is imposed by Ω_i's neighbor via (44). Constraints (42) and (44) involve geometric constants and the values of the basis functions at the outflow edges' midpoints, which are listed in Table 1. Consequently, the constraints depend on how the physical triangles are mapped to the canonical triangle. This depends on how the vertices are labeled in the physical triangle. By (4), vertex $\mathbf{x}_{i,1}$ is always mapped to $(0,0)$ (Figure 4). However, which vertex of the triangle is listed as $\mathbf{x}_{i,1}$ depends on the mesh generator. For the

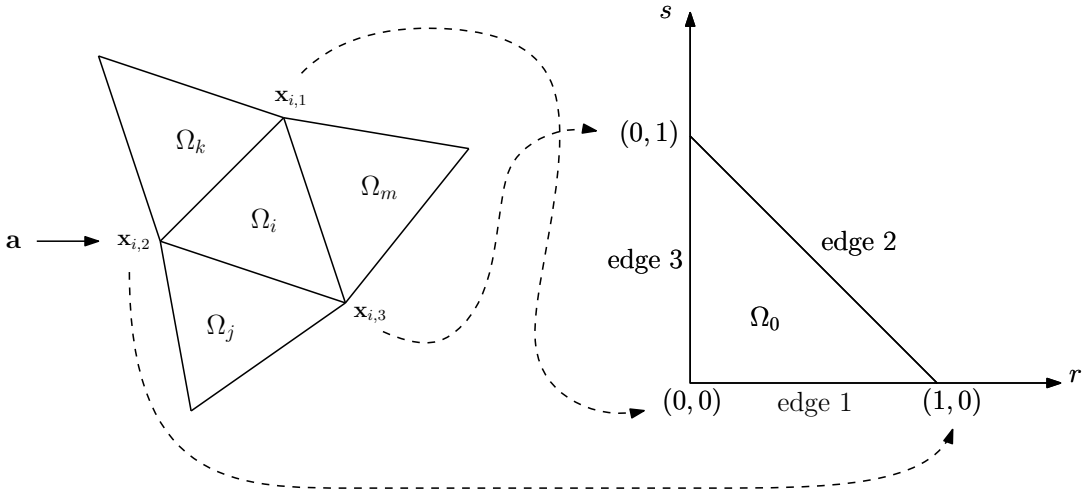


FIGURE 4. Mapping of Ω_i to the canonical triangle Ω_0 by (4).

edge number s, if it is an outflow edge of Ω_i, the constraints (42) and (44) are

$$\frac{l_{i,1}}{6\gamma_{i,1}^f} + \frac{l_{i,2}}{4\gamma_{i,2}^b r_{i,2}} \le 1 \quad \text{and} \quad \frac{l_{i,1}}{6\gamma_{i,1}^b r_{i,1}} + \frac{l_{i,2}}{4\gamma_{i,2}^f} \le 1 \quad \text{if} \quad s = 1, \tag{51}$$

$$\frac{l_{i,1}}{6\gamma_{i,1}^f} + \frac{l_{i,2}}{4\gamma_{i,2}^f} \le 1 \quad \text{and} \quad \frac{l_{i,1}}{6\gamma_{i,1}^b r_{i,1}} + \frac{l_{i,2}}{4\gamma_{i,1}^b r_{i,2}} \le 1 \quad \text{if} \quad s = 2, \tag{52}$$

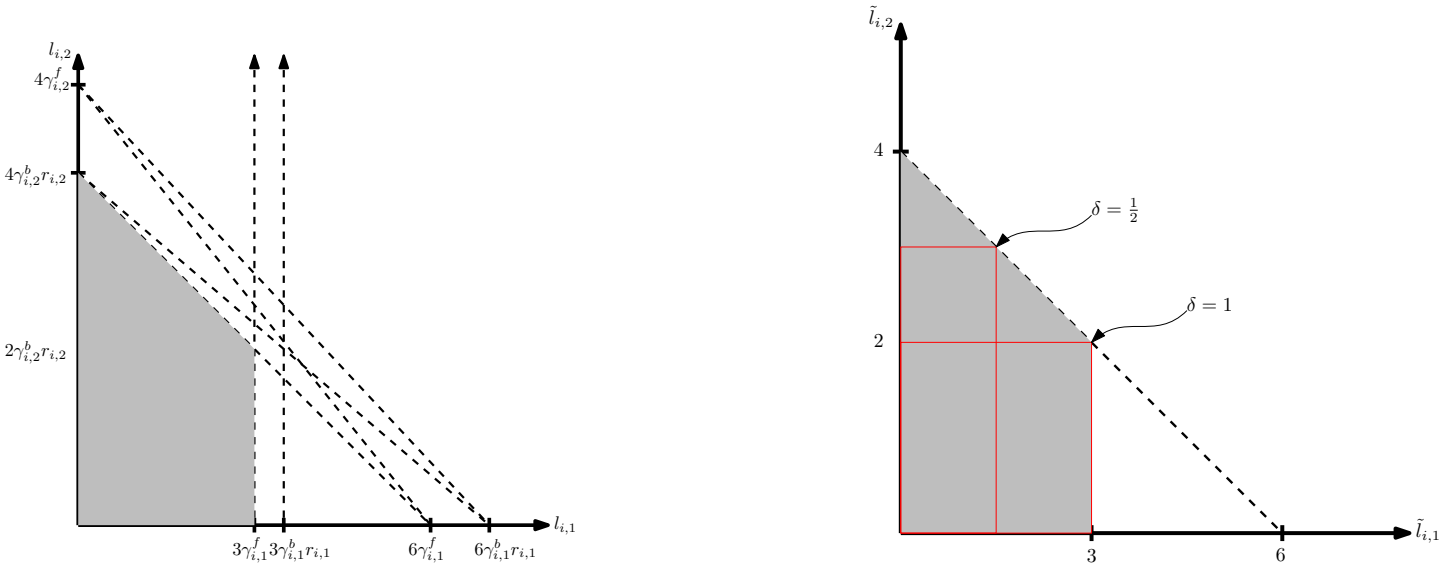
$$\frac{l_{i,1}}{3\gamma_{i,1}^b r_{i,1}} \le 1 \quad \text{and} \quad \frac{l_{i,1}}{3\gamma_{i,1}^f} \le 1 \quad \text{if} \quad s = 3, \tag{53}$$

where $\gamma_{i,k}^f = \frac{d_{i,k}^f}{h_{i,k}}$, $\gamma_{i,k}^b = \frac{d_{i,k}^b}{h_{i,k}}$, and $r_{i,k}$ are given by (17), for $k = 1, 2$.

To illustrate, let us consider an example in Figure 4, with the flow direction $\mathbf{a}$. In this case, Ω_i has one outflow edge $\partial\Omega_{i,m}$. This edge is mapped to edge 3 on the canonical triangle and, consequently, the only constraints on the slope of $U_i^n(\mathbf{x})$ are (53). Next, consider the opposite flow direction $-\mathbf{a}$. The outflow edges of Ω_i are now $\partial\Omega_{i,k}$ and $\partial\Omega_{i,j}$. These edges are mapped to edges 1 and 2 on the canonical triangle, respectively. Therefore, the constraints on the slope of $U_i^n(\mathbf{x})$ are (51) and (52).

Thus, we must check each edge for being an outflow edge and determine which edge of the canonical triangle it maps to. This involves extra coding effort and slows computations. Alternatively, we might choose enforcing (42) and (44) on all three edges of Ω_i, i.e. enforce (51) - (53). This will result in a more restrictive limiter, however this limiter will be easier to implement. Inequalities (51) - (53) can be simplified to

$$\begin{cases} \frac{l_{i,1}}{6\min(\gamma_{i,1}^f, \gamma_{i,1}^b r_{i,1})} + \frac{l_{i,2}}{4\min(\gamma_{i,2}^f, \gamma_{i,2}^b r_{i,2})} & \le 1, \\ \frac{l_{i,1}}{3\min(\gamma_{i,1}^f, \gamma_{i,1}^b r_{i,1})} & \le 1. \end{cases}$$



(a) The full set of inequalities.

(b) Simplified set of inequalities.

FIGURE 5. The gray region is the admissibility region which satisfies all constraints. The region bounded by the rectangle is the one used by the limiter for the chosen δ.

Introducing the modified limiting coefficients

$$\tilde{l}_{i,1} = \frac{l_{i,1}}{\min(\gamma^f_{i,1}, \gamma^b_{i,1} r_{i,1})} \text{ and } \tilde{l}_{i,2} = \frac{l_{i,2}}{\min(\gamma^f_{i,2}, \gamma^b_{i,2} r_{i,2})},$$

the constraints become

$$\begin{cases} \frac{1}{6}\tilde{l}_{i,1} + \frac{1}{4}\tilde{l}_{i,2} & \leq 1, \\ \frac{1}{3}\tilde{l}_{i,1} & \leq 1. \end{cases}$$

We give an illustration of the inequalities (51) - (53) in Figure 5a. For plotting, we chose a particular relation between the geometric constants γ and ratios r, i.e., $\gamma^f_{i,1} \leq \gamma^b_{i,1} r_{i,1}$ and $\gamma^b_{i,2} r_{i,2} \leq \gamma^f_{i,2}$, though in general this is not always the case. The boundaries of the inequalities are plotted as dashed lines. Values satisfying the inequalities will be to the left and below those lines. The same is done for the simplified set of inequalities in Figure 5b. Recall that $l_{i,1}$ or $l_{i,2}$ are non-negative by construction. The case where $l_{i,1}$ or $l_{i,2}$ is equal to zero corresponds to when the forward and backward differences are of opposite sign. This gives the left and bottom boundaries of the stability region. Then, the region from which suitable limiting coefficients can be taken is shown in gray. A simple sufficient condition that satisfies all these constraints is

$$\tilde{l}_{i,1} \leq 3\delta \quad \text{and} \quad \tilde{l}_{i,2} \leq 4 - 2\delta,$$

s	φ_1	φ_2
1	1	$-\sqrt{3}$
2	1	$\sqrt{3}$
3	-2	0

TABLE 1. Values of the basis functions at the midpoints of the canonical edges with edge number s.

for any $\delta \in (0, 1]$. That is, with $r_{i,1} > 0$ and $r_{i,2} > 0$,

$$l_{i,1} \leq 3\delta \min(\gamma^f_{i,1}, \gamma^b_{i,1} r_{i,1}) \quad \text{and} \quad l_{i,2} \leq (4 - 2\delta) \min(\gamma^f_{i,2}, \gamma^b_{i,2} r_{i,2}). \tag{54}$$

This is the region bounded by the rectangles in Figure 5b. Multiplying the first inequality by $\frac{h_{i,1}}{6}|\Delta^f_{i,1}|$ and the second by $\frac{h_{i,2}}{4\sqrt{3}}|\Delta^f_{i,2}|$, by (18) the limited coefficients $\tilde{c}^n_{i,1}$ and $\tilde{c}^n_{i,2}$ must satisfy

$$|\tilde{c}^n_{i,1}| \leq \frac{h_{i,1}}{2}\delta|\Delta^f_{i,1}|\text{min}\left(\frac{d^f_{i,1}}{h_{i,1}}, \frac{d^b_{i,1}}{h_{i,1}} r_{i,1}\right),$$
$$|\tilde{c}^n_{i,2}| \leq \frac{h_{i,2}}{4\sqrt{3}}(4 - 2\delta)|\Delta^f_{i,2}|\text{min}\left(\frac{d^f_{i,2}}{h_{i,2}}, \frac{d^b_{i,2}}{h_{i,2}} r_{i,2}\right).$$

We choose between the current solution coefficient $c^n_{i,k}$ and the largest one allowed, i.e., the upper bounds in the above inequalities. Thus, we have

$$\tilde{c}^n_{i,1} = \frac{h_{i,1}}{2}\delta\Delta^f_{i,1}\text{minmod}\left(\frac{d^f_{i,1}}{h_{i,1}}, \frac{2}{h_{i,1}\delta\Delta^f_{i,1}} c^n_{i,1}, \frac{d^b_{i,1}}{h_{i,1}} r_{i,1}\right),$$
$$\tilde{c}^n_{i,2} = \frac{h_{i,2}}{4\sqrt{3}}(4 - 2\delta)\Delta^f_{i,2}\text{minmod}\left(\frac{d^f_{i,2}}{h_{i,2}}, \frac{4\sqrt{3}}{h_{i,2}(4 - 2\delta)\Delta^f_{i,2}} c^n_{i,2}, \frac{d^b_{i,2}}{h_{i,2}} r_{i,2}\right).$$

Simplifying, this becomes

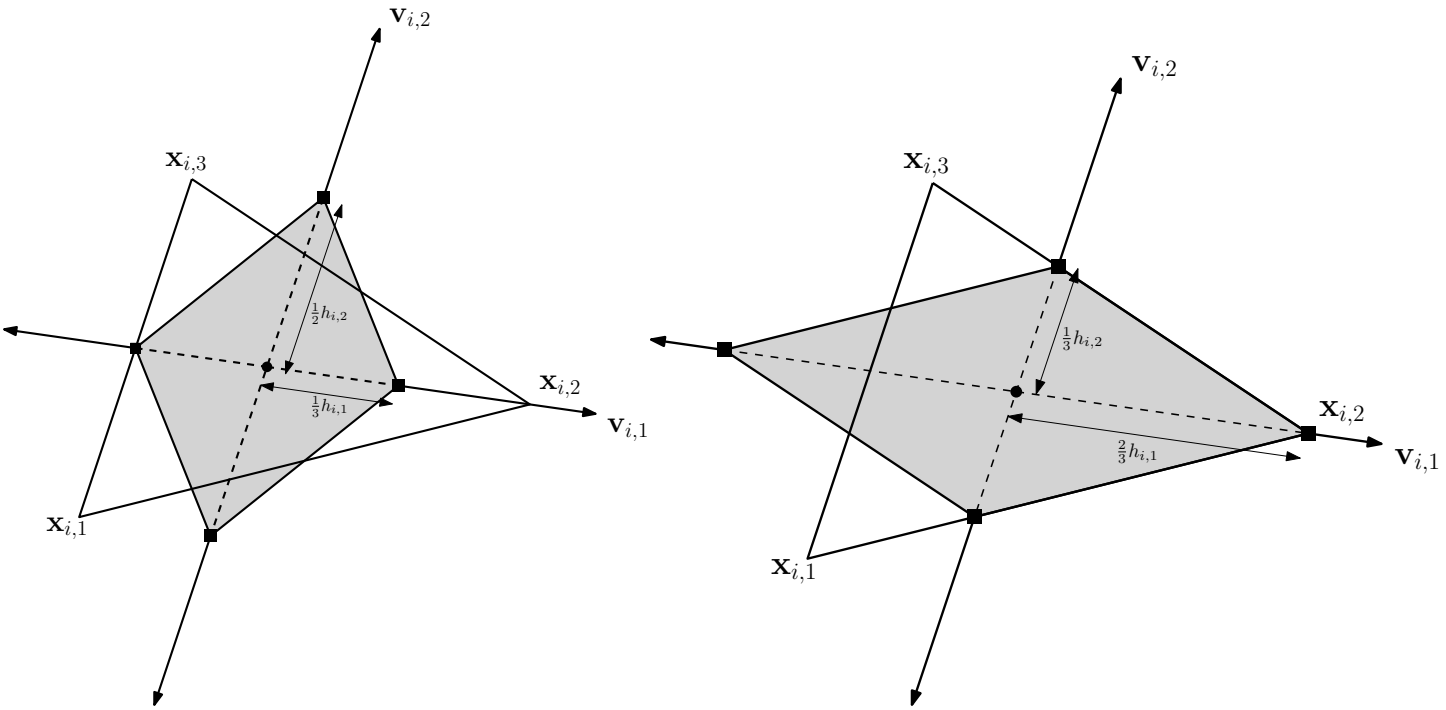
$$\tilde{c}^n_{i,1} = \text{minmod}\left(\delta\frac{U^f_{i,1} - \overline{U}^n_i}{2}, c^n_{i,1}, \delta\frac{\overline{U}^n_i - U^b_{i,1}}{2}\right), \tag{55}$$

$$\tilde{c}^n_{i,2} = \text{minmod}\left((4 - 2\delta)\frac{U^f_{i,2} - \overline{U}^n_i}{4\sqrt{3}}, c^n_{i,2}, (4 - 2\delta)\frac{\overline{U}^n_i - U^b_{i,2}}{4\sqrt{3}}\right). \tag{56}$$

3.6. **Geometric requirements.** By (17), a solution that is linear in x and y will have that $r_{i,1} = r_{i,2} = 1$. In this case, the reconstructed slope must not be reduced for second order spatial accuracy to be preserved, i.e. $1 \leq l_{i,1}, l_{i,2}$. Then, by (54) we have

$$1 \leq 3\delta \min(\gamma^f_{i,1}, \gamma^b_{i,1}), \tag{57}$$

$$1 \leq (4 - 2\delta) \min(\gamma^f_{i,2}, \gamma^b_{i,2}). \tag{58}$$



(a) $\delta = 1$ (b) $\delta = \frac{1}{2}$

FIGURE 6. The forward and backward interpolation points $x^f_{i,k}$, $x^b_{i,k}$ must lie a distance from the centroid of the cell in the direction of $\mathbf{v}_{i,k}$ for $k = 1, 2$ greater than the one indicated, i.e. outside the shaded diamond.

By the definition of the coefficients $\gamma^b_{i,1}$, $\gamma^f_{i,1}$, $\gamma^b_{i,2}$, $\gamma^f_{i,2}$, the above becomes

$$\min(d^f_{i,1}, d^b_{i,1}) \geq \frac{h_{i,1}}{3\delta}, \tag{59}$$

$$\min(d^f_{i,2}, d^b_{i,2}) \geq \frac{h_{i,2}}{4 - 2\delta}. \tag{60}$$

This means that the interpolation points must lie a minimum distance from the cell centroid that depends on δ (Figure 6).

3.7. **Preprocessing.** The vertices of the triangle Ω_i, $(\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \mathbf{x}_{i,3})$, are mapped by (4) to the vertices $(0, 0)$, $(1, 0)$, and $(0, 1)$ of the canonical triangle, respectively. In order for the determinant of the Jacobian of this mapping given in (5) to be positive, the vertices must be ordered counterclockwise. There are three mappings and, consequently, three pairs of limiting directions, that depend on which vertex of the triangle is listed as $\mathbf{x}_{i,1}$ (Figure 7).

The local maximum principle (1) restricts the cell average on Ω_i at time t^{n+1} to a locally defined interval that depends on how the reconstruction neighborhood N_i is chosen. It would be computationally advantageous for N_i to consist of elements that share an edge with the cell Ω_i, as this is the stencil for the DG method. Unfortunately, this may be incompatible with the geometrical requirements (57) and

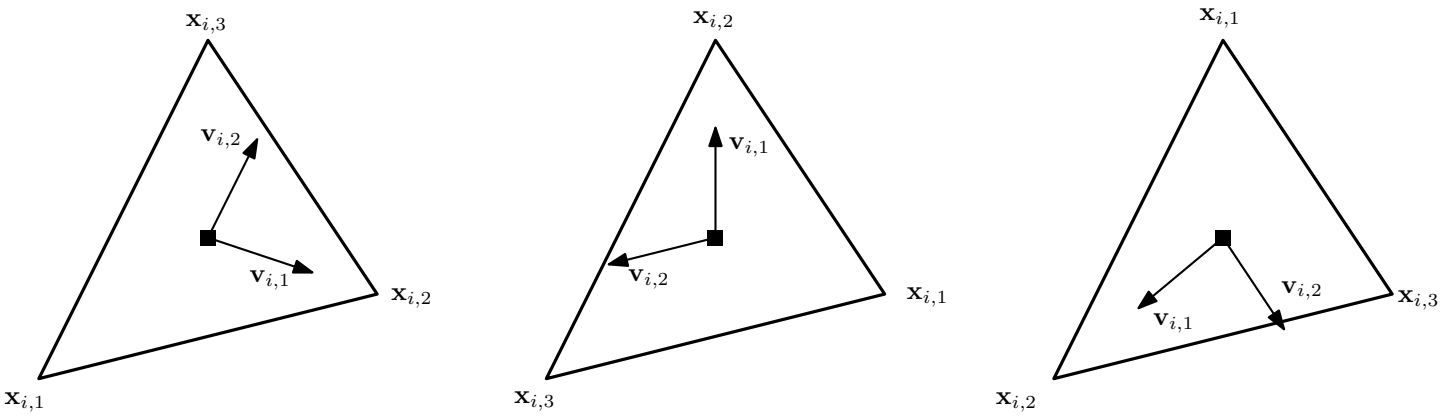


FIGURE 7. Limiting directions of the triangle Ω_i for three vertex numberings.

(58) given in Section 3.6. This is illustrated in Figure 8a where we determine the interpolation points for the limiter (55), (56) with $\delta = 1$ using the procedure described in Section 2. The forward and backwards interpolation points in the direction of $\mathbf{v}_{i,2}$ do not lie far enough away from the centroid of Ω_i because they lie inside the shaded diamond. However, if we enlarge N_i to contain all elements that share a vertex with Ω_i, we obtain interpolation points that satisfy (57) and (58) (Figure 8b). Note that other approaches to find interpolation points are possible, however the present one provides a systematic way to find such points and is simple to code.

These interpolation points are computed during preprocessing of the mesh. We store the pointers to the eight elements involved in limiting and four interpolation coefficients $\beta^f_{i,1}$, $\beta^b_{i,1}$, $\beta^f_{i,2}$, $\beta^b_{i,2}$, (Figure 2). The coordinates of the interpolation points are not needed so they are not stored.

4. Numerical examples

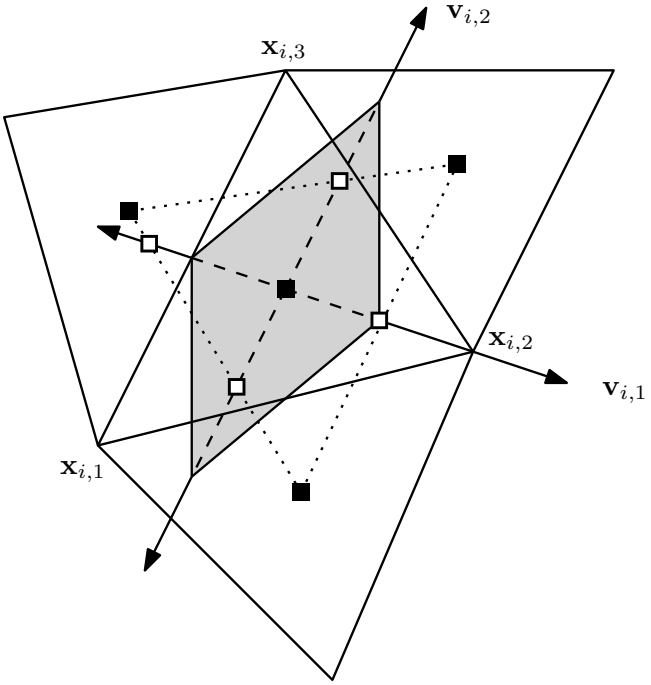
In all examples, we choose $\delta = 1$. The limiter (55), (56) becomes

$$\tilde{c}^n_{i,1} = \text{minmod}\left(\frac{U^f_{i,1} - \overline{U}^n_i}{2}, c^n_{i,1}, \frac{\overline{U}^n_i - U^b_{i,1}}{2}\right), \tag{61}$$

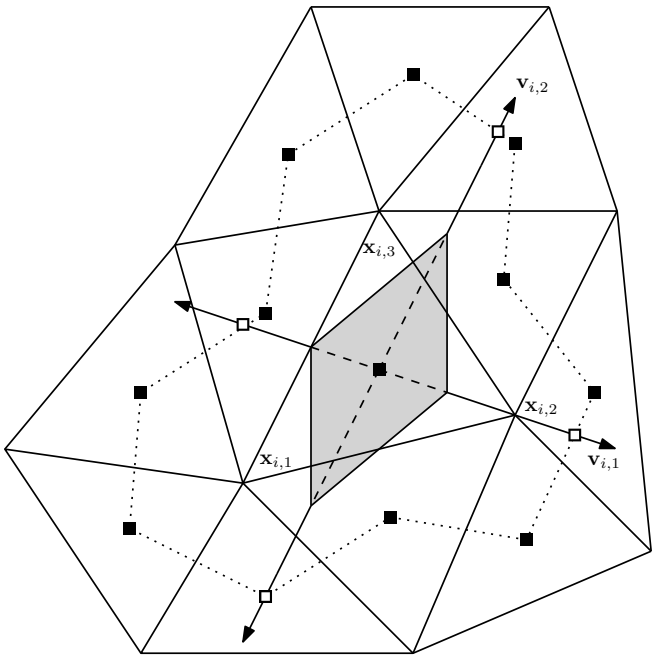
$$\tilde{c}^n_{i,2} = \text{minmod}\left(\frac{U^f_{i,2} - \overline{U}^n_i}{2\sqrt{3}}, c^n_{i,2}, \frac{\overline{U}^n_i - U^b_{i,2}}{2\sqrt{3}}\right). \tag{62}$$

Unless otherwise stated, we solve (2) on the square domain $[-1, 1]^2$ with RK2 time stepping, and limiter (61), (62). The moment limiter is implemented in CUDA C and executed on NVIDIA GPUs with the DG implementation [16] using the optimizations described in [17]. This limiting algorithm is very suitable for GPU acceleration due to the stencil of constant size.

4.1. **Verification of CFL number and global bounds on the solution.** In this example, we verify the time step restriction (49). We use the flux $\mathbf{F}(u) = [u, u]$, along with the the initial condition of a square



(a) Edge neighborhood. The forward and backwards interpolation points in the direction of $\mathbf{v}_{i,2}$ do not satisfy the geometric requirement (58).



(b) Vertex neighborhood. All interpolation points satisfy the geometric requirements (57) and (58).

FIGURE 8. Interpolation points for $\Omega_i = (\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \mathbf{x}_{i,3})$, denoted by hollow square ticks, with the edge (a) and vertex (b) neighborhoods. Solid squares denote cell centroids. The interpolation points must lie outside the shaded region to allow for second order accuracy.

pulse

$$u_0(x,y) = \begin{cases} 1 \text{ if } \max(|x|,|y|) \leq \frac{1}{4} \\ \\ 0 \text{ elsewhere.} \end{cases}$$

The problem is solved until the final time $T = 0.1$ on a structured mesh of 11,522 triangles. The mesh is obtained by first tiling the domain with 76 by 76 squares. Then, we split the squares along their diagonals, connecting the square's upper left and lower right corners, to create triangles. The time step restriction is obtained from the wave speed $||\mathbf{a}|| = \sqrt{2}$ and cell width in the direction of flow, which is $\frac{1}{76}\sqrt{2} \approx 1.860 \cdot 10^{-2}$. The minimum and maximum solution cell averages of the final solution are tabulated in Table 2 for forward Euler and RK2 time steppers using different values of the CFL number. These results verify that time step restriction (49) is tight for the forward Euler method. Additionally, we find that the CFL number can be increased without the the solution averages at the final time exceeding the global bounds of the initial condition. However, the quality of the solution is adversely affected when we exceed the CFL number required for linear stability [14]. We will demonstrate this in the following example.

4.2. **Verification of accuracy.** The CFL number from a linear stability analysis of the scheme (7) is $\frac{3}{13} = \frac{1}{4.333...}$ [14]. We note that this is smaller than the one required by the LMP in (49). In this example,

1/CFL	Minimum	Maximum
2	-3.97e-01	1.14
3	-4.83-02	1.00062
3.5	-4.31e-03	1
4	**-3.39e-19**	**1**

(a) Forward Euler.

1/CFL	Minimum	Maximum
2	-1.19e-19	1
3	-3.31-22	1
3.5	-1.91e-21	1
4	**-4.10e-21**	**1**

(b) RK2.

TABLE 2. Global bounds on the cell averages of the final solution in Example 4.1 using time step restriction (49).

we demonstrate that the smaller of the two CFL numbers should be used. With the flux $\mathbf{F}(u) = [u, 0]$, we solve an advecting pulse problem with the initial condition

$$u_0(x,y) = \begin{cases} \cos^2(2\pi r), & \text{if } r \le \frac{1}{4}, \\ 0, & \text{otherwise,} \end{cases} \tag{63}$$

where $r = \sqrt{(x+\frac{1}{4})^2 + y^2}$. In Table 3, we present the global solution bounds and L_1 errors at $T = 0.5$ with and without a limiter using two different CFL numbers. With a CFL number equal to $\frac{1}{4}$, unlimited solutions are unstable (Table 3c) as predicted by the linear stability analysis. The limiter suppresses the instability and prevents growth of the global bounds of the solution but the accuracy and observed rate of convergence are severely affected (compare the L_1 errors in Tables 3a and 3b). Although the solution is stable, it is not accurate and consequently we must use a smaller CFL number.

The limiter does not introduce a substantial error on this structured grid (compare the L_1 errors in Tables 3b and 3d). Thus, in the following examples we use the CFL number given by the linear stability analysis [14].

4.3. **Limiter** (61)**.** The limiter presented in Section 3 depends on the numbering of the element vertices, i.e. is mapping dependent. Here we construct an example where all elements in the mesh have only one outflow edge, and that edge is always mapped to the third edge ($k = 3$) of Ω_0 (Figure 4). In this case, instead of the general limiter (61)-(62), we can use only limiter (61). Then, only the $c^n_{i,1}$ coefficient is limited and $c^n_{i,2}$ is left untouched.

With the flux $\mathbf{F}(u) = [u, 0]$, we solve an advecting square pulse problem with the initial condition

$$u_0(x,y) = \begin{cases} 1, \text{ if } \max(\left|x+\frac{1}{4}\right|, |y|) \le \frac{1}{4}, \\ 0, \text{ elsewhere,} \end{cases} \tag{64}$$

Number of elements	Minimum	Maximum	L_1 error
200	-1.004307e-37	2.932214e-01	5.1693e-02 (-)
800	-3.317668e-39	6.271256e-01	2.1214e-02 (1.28)
3,200	-1.500449e-62	8.490757e-01	5.5147e-03 (1.94)
12,800	-3.515808e-42	9.444271e-01	1.3530e-03 (2.02)
51,200	**-3.410031e-45**	**9.803368e-01**	**4.5571e-04 (1.56)**
204,800	**-7.484794e-45**	**9.932226e-01**	**3.0248e-04 (0.59)**

(a) Accuracy for $1/\text{CFL} = 4$ with moment limiter.

Number of elements	Minimum	Maximum	L_1 error
200	-4.482126e-38	2.930021e-01	5.1733e-02 (-)
800	-1.974258e-36	6.264496e-01	2.1039e-02 (1.29)
3,200	-3.639373e-73	8.495680e-01	5.4395e-03 (1.95)
12,800	-1.525488e-60	9.446573e-01	1.3091e-03 (2.05)
51,200	-2.857630e-44	9.804285e-01	3.0646e-04 (2.09)
204,800	-4.616411e-42	9.932461e-01	7.2674e-05 (2.07)

(b) Accuracy for $1/\text{CFL} = \frac{13}{3} = 4.333\ldots$ with moment limiter.

Number of elements	Minimum	Maximum	L_1 error
200	-1.069623e-01	5.838654e-01	4.0877e-02 (-)
800	-4.193137e-02	8.903924e-01	1.5198e-02 (1.42)
3,200	-4.965123e-02	9.683763e-01	1.5860e-02 (-)
12,800	-3.270390e+00	3.202645e+00	7.7678e-01 (-)
51,200	-1.001783e+05	1.069657e+05	1.6659e+04 (-)
204,800	-5.694670e+14	5.489725e+14	6.2178e+13 (-)

(c) Accuracy for $1/\text{CFL} = 4$, unlimited.

Number of elements	Minimum	Maximum	L_1 error
200	-7.867535e-02	5.795796e-01	3.1425e-02 (-)
800	-5.092045e-02	8.823980e-01	9.9184e-03 (1.66)
3,200	-2.248452e-02	9.678742e-01	2.7487e-03 (1.85)
12,800	-9.359466e-03	9.920149e-01	7.3307e-04 (1.90)
51,200	-3.713583e-03	9.980151e-01	1.9241e-04 (1.92)
204,800	-1.459199e-03	9.995055e-01	4.9797e-05 (1.95)

(d) Accuracy for $1/\text{CFL} = \frac{13}{3} = 4.333\ldots$, unlimited.

TABLE 3. L_1 errors, convergence rates (in parentheses), and global bounds on the final solution for Example 4.2.

on the same mesh as in Section 4.1 until $T = 0.5$. We pre-process the mesh so that the outflow edges of all elements are mapped to the third edge of Ω_0. In both the initial and final solutions (Figure 9), we observe oscillations in the y-direction. The amplitude of these oscillations diminishes with time (Figures 9a and 9b). In the x direction, the solutions are smoother, with discontinuities spread over two cell widths. Although large overshoots are present in the solution, the means are still confined to the initial range $[0, 1]$. This example demonstrates that the local maximum principle (1) does not guarantee an oscillation-free solution, even for scalar equations. For comparison, we plot the solution with the full limiter (61)-(62) in

Mesh	Elements	Moment	Unlimited
A	1,250	1.9615e-02 (-)	8.3553e-03 (-)
B	5,190	3.7825e-03 (2.37)	1.6512e-03 (2.33)
C	20,552	7.4542e-04 (2.34)	3.7238e-04 (2.14)
D	81,878	1.4196e-04 (2.39)	8.5916e-05 (2.11)

TABLE 4. L_1 errors and convergence rates (in parentheses) for the advecting pulse problem (63) in Example 4.4.

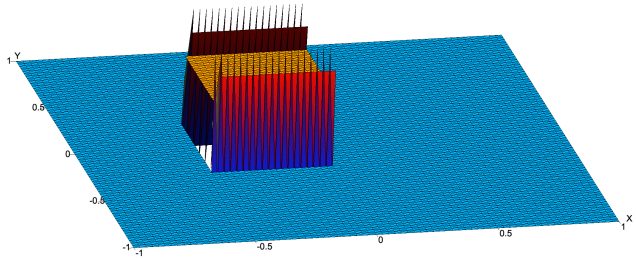
Figures 9e, 9f, 10c, and 10d. We observe that the oscillations in the y-direction are suppressed, although the discontinuities are more diffused and there is a slight smearing effect at the corners of the pulse.

We note however that this is an artificial example as we have a discontinuity that is perpendicular to the edges of the elements, on a mesh of elements with a special mapping to the canonical element. Rotating the initial square pulse by 45 degrees results in a solution with suppressed overshoots, even with limiter (61) (Figures 10a, 10b).

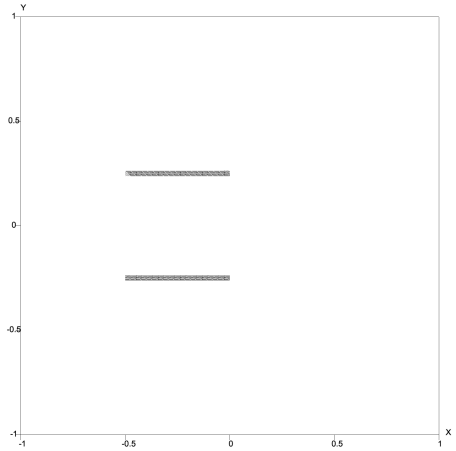
Finally, we solve the problem with a smooth initial condition given by (63). The L_1 error at $T = 0.5$ is 1.4982e-03 for the limiter (61)-(62), and 1.1346e-03 for the limiter (61). This demonstrates that the limiter (61)-(62) does not catastrophically degrade solution quality.

4.4. **Advecting hill.** With the flux $\mathbf{F}(u) = [u, u]$, we solve an advecting hill problem with the initial condition (63), where $r = \sqrt{(x + \frac{1}{4})^2 + (y + \frac{1}{4})^2}$, on a sequence of unstructured meshes A-D. Refined meshes were obtained by remeshing the domain with an increased target number of elements. The final solutions at $T = 0.5$ on meshes A and D with the moment limiter are plotted in Figure 11. The L_1 errors on meshes A-D are reported in Table 4. As expected, the limiter reduces solution accuracy. Nevertheless, we observe the second order rate of convergence in the L_1 norm and linear convergence in the L_∞ norm in approximation of the local maximum (Figure 11c). For the resolved solutions, the moment limiter increases the error by a factor of approximately two (the solution on mesh A is badly resolved due to an insufficient number of elements).

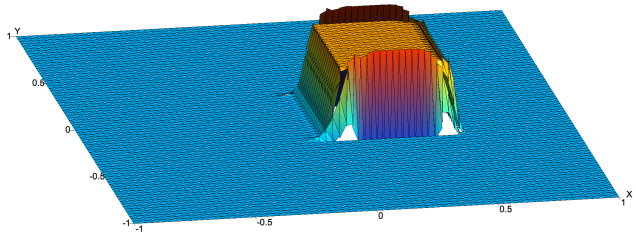
Next, we study how the limiting directions $\mathbf{v}_{i,1}$ and $\mathbf{v}_{i,2}$ (Figure 7) influence the performance of the limiter (61)-(62). Three pairs of $\mathbf{v}_{i,1}$, $\mathbf{v}_{i,2}$ are possible on each Ω_i. This results in three possible limiting stencils and sets of parameter γ. For each pair of limiting directions in Figure 7, we computed the minimum and maximum γ out of the four involved in the limiting process, named $\gamma_{\min}$ and $\gamma_{\max}$, respectively. By maximizing γ, we mean that the configuration with the largest $\gamma_{\min}$ was chosen. Conversely, by minimizing γ, we mean that the configuration with the smallest $\gamma_{\max}$ was chosen. The errors in solutions obtained with different limiting directions are given in Table 5. It appears that choosing reconstruction points closest



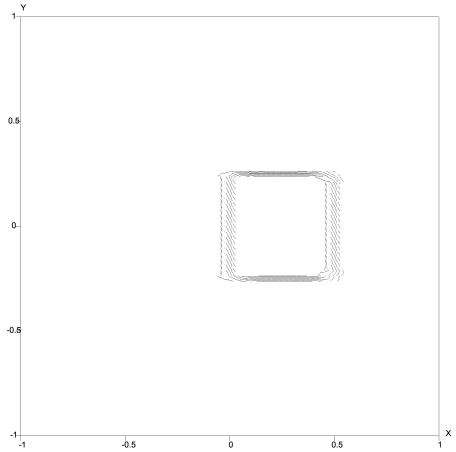
(a) The initial condition using limiter (61).



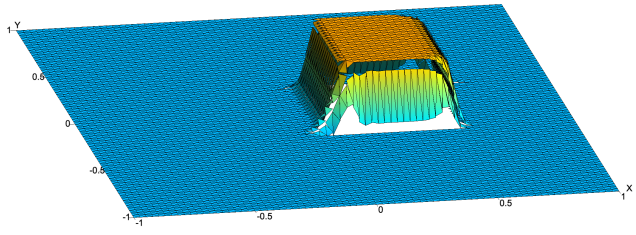
(b) Isolines of the initial condition using limiter (61).



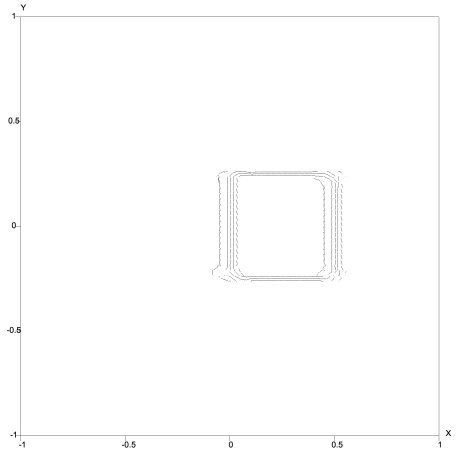
(c) The final solution at $T = 0.5$ using (61).



(d) Isolines of the final solution at $T = 0.5$ using limiter (61).

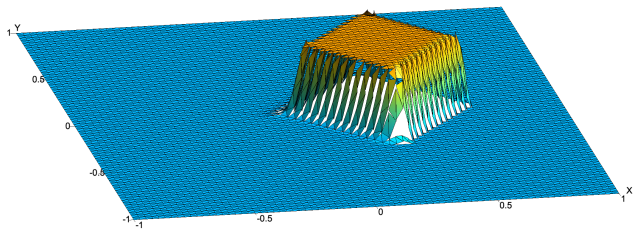


(e) The final solution at $T = 0.5$ using limiter (61)-(62).

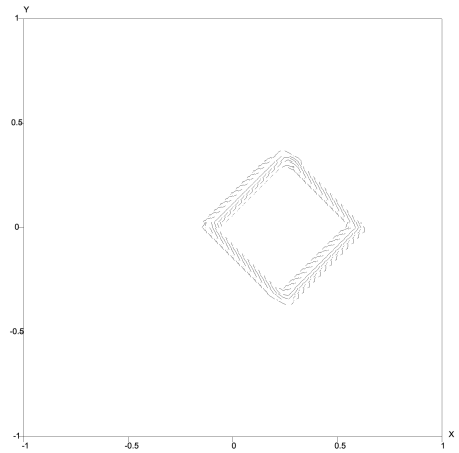


(f) Isolines of the final solution at $T = 0.5$ using limiter (61)-(62).

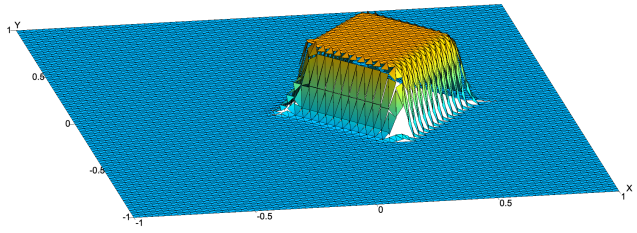
FIGURE 9. Advecting square pulse (64) in Example 4.3.



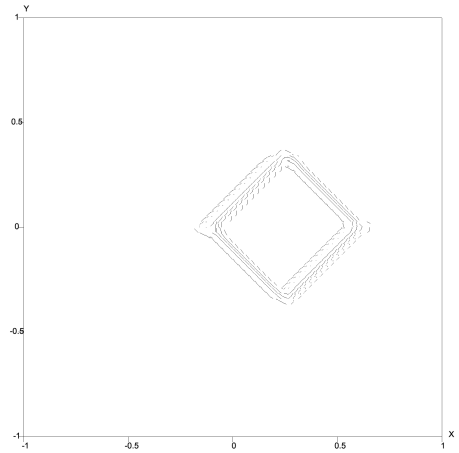
(a) The final solution at $T = 0.5$ using limiter (61).



(b) The final solution at $T = 0.5$ using limiter (61).



(c) The final solution at $T = 0.5$ using limiter (61)-(62).



(d) Isolines of the final solution at $T = 0.5$ using limiter (61)-(62).

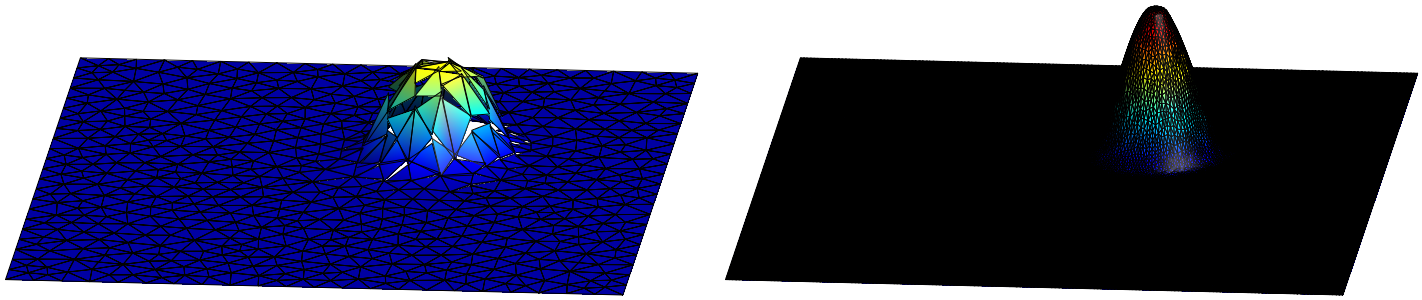
FIGURE 10. Advecting rotated square pulse in Example 4.3.

Mesh	Minimized γ	Mesh generator γ	Maximized γ
A	1.9042e-02	1.9615e-02	1.9175e-02
B	3.7692e-03	3.7825e-03	3.9160e-03
C	7.3582e-04	7.4542e-04	7.6800e-04
D	1.3915e-04	1.4196e-04	1.4369e-04

TABLE 5. L_1 errors for different choices of parameters γ in Example 4.4.

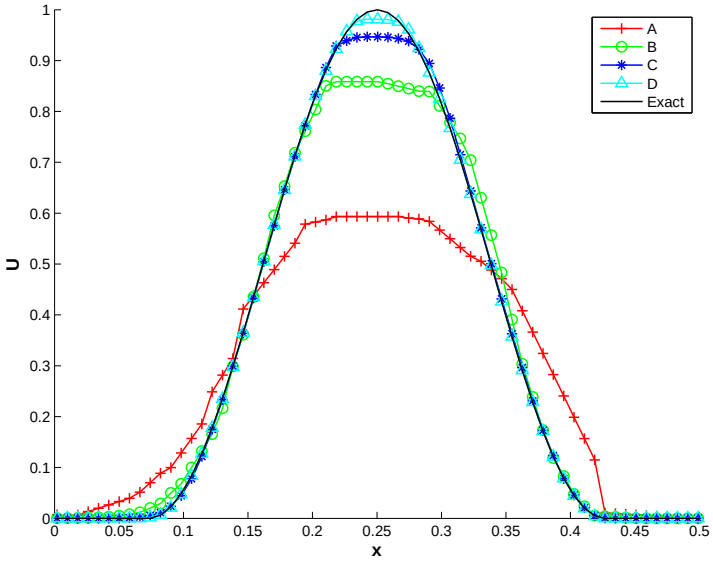
to the cell centroid is beneficial, but not substantially so. Thus, the vertex ordering given by the mesh generator can be used directly, without additional preprocessing of the mesh.

4.5. **Rotating shapes.** With the flux $\mathbf{F}(u) = [-2\pi yu, 2\pi xu]$, we solve a rotating shapes problem. The exact solution is a rotation of the initial data about the origin. The initial condition comprises a hill and



(a) Raised solution A.

(b) Raised solution D.



(c) Profile of solution along the line $y = x$.

FIGURE 11. Advecting hill (63) in Example 4.4 at $T = 0.5$ with the moment limiter.

a square pulse (Figure 12) defined by

$$u_0(x,y) = \begin{cases} \cos^2(2\pi r) \text{ if } r \leq 0.25, \\ 1 \text{ if } \max(|x-0.35|, |y|) \leq 0.25, \\ 0 \text{ elsewhere}, \end{cases}$$

where $r = \sqrt{(x+0.5)^2 + y^2}$. This problem is solved on an unstructured mesh of 12,792 triangles using the minimum cell width (50) as the measure of cell size. The isolines and profile along $y = 0$ of the final solution are displayed in Figure 13. This solution is of comparable quality to the moment limiter described in [13], solved on a structured mesh of $80 \times 80 = 6{,}400$ quadrilaterals.

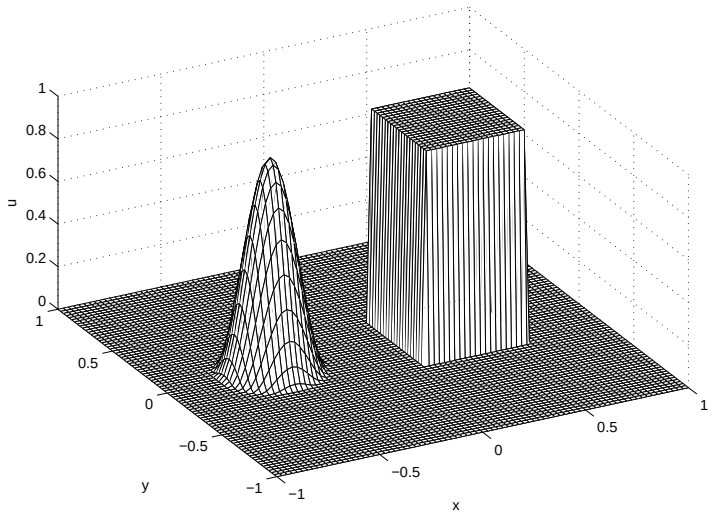
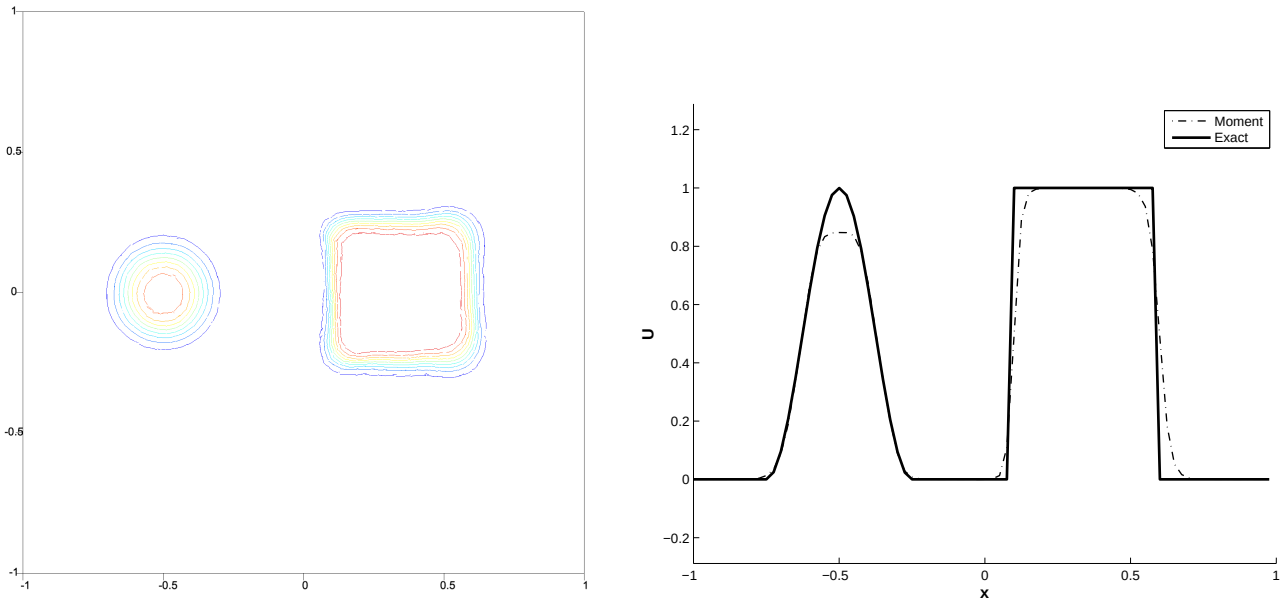


FIGURE 12. Initial condition in Example 4.5.



(a) Isolines with moment limiter.

(b) Profile of solution on the line $y = 0$.

FIGURE 13. Rotating shapes at $T = 1$.

4.6. **Double Mach reflection problem.** We consider the two-dimensional Euler equations

$$\frac{d}{dt}\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ E\end{pmatrix}+\frac{d}{dx}\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (E+p)u\end{pmatrix}+\frac{d}{dy}\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ (E+p)v\end{pmatrix}=0$$

with the equation of state

$$p=(\gamma-1)\left(E-\frac{\rho}{2}(u^2+v^2)\right),$$

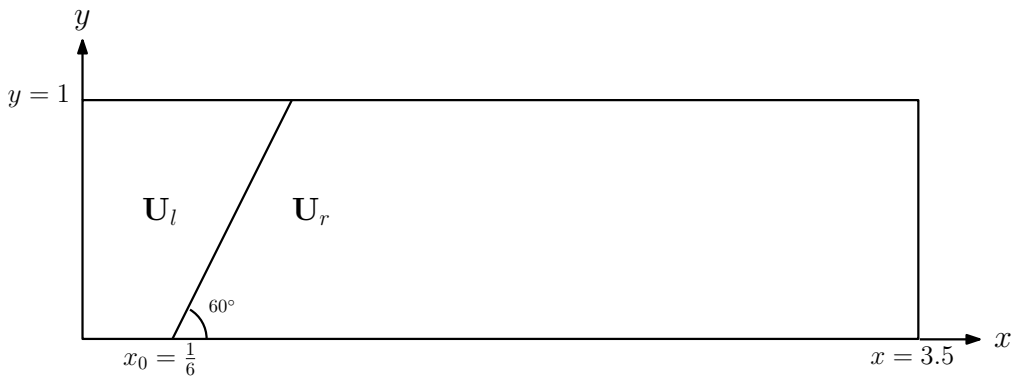


FIGURE 14. Set-up of the double Mach test problem in Example 4.6.

where $\gamma = 1.4$ is the adiabatic constant for air. We solve the double Mach reflection problem using the set up described in [18, 16] and illustrated in Figure 14. The computational domain is $[0, 3.5] \times [0, 1]$ with a Mach 10 shock impinging with an angle of 60° on a reflecting boundary. The states to the left $\mathbf{U}_l$ and right $\mathbf{U}_r$ of the shock given in Table 6. The problem is solved on an unstructured mesh of 271,458 triangles until a final time of $T = 0.2$. The moment limiter is extended to systems by limiting each conserved variable. We observe that the shocks are well resolved, and the slipline emanating from the primary triple point is tight. These are results are again comparable to those obtained with a moment limiter on Cartesian grids [13].

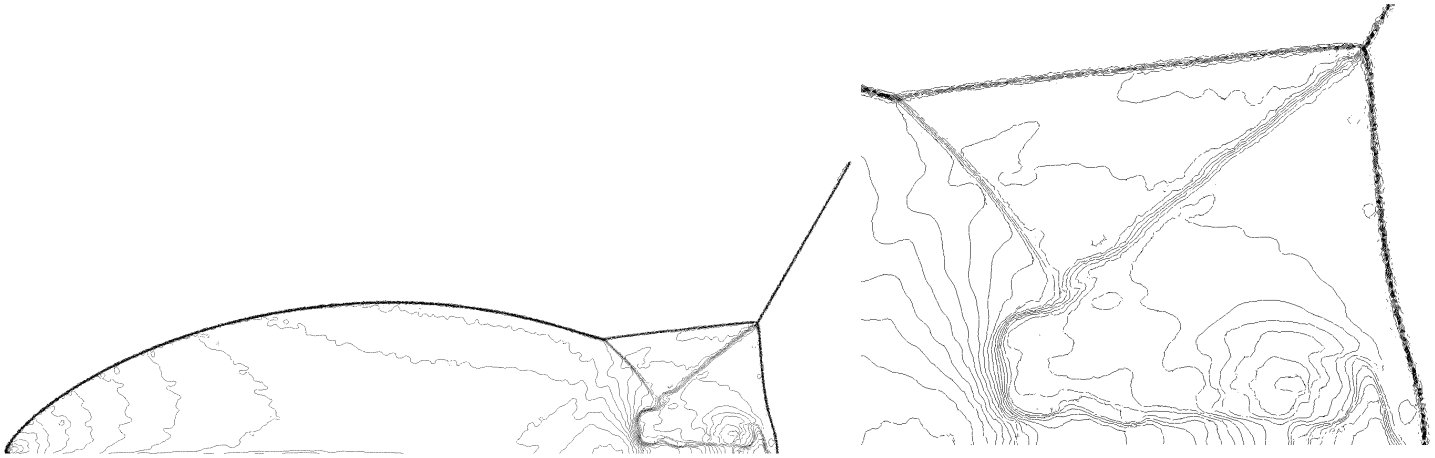
We compare the GPU runtime performance of the moment limiter to the so-called Barth-Jespersen limiter. The Barth-Jespersen limiter uses either the edge neighborhood or vertex neighborhood to determine a local interval by which to bound the numerical solution at the edge midpoints. It was shown in [19] that the limiter using the edge neighborhood is first order accurate but fast and the limiter using the vertex neighborhood is second order accurate but slow. The total time spent executing the three limiting algorithms on an NVIDIA Titan X Pascal is provided in Table 7. We observe that the moment limiter is the fastest of the three limiters and takes approximately 8.6 percent of the total DG-GPU solver run time. The Barth-Jespersen limiter using the edge neighborhood is slightly slower than the moment limiter, but it is only first order accurate. The Barth-Jespersen limiter using the vertex neighborhood is second order accurate, but it executes three times slower per time step than the moment limiter and takes 22 percent of the total solver run time, which is non-negligible. We also note that both Barth-Jespersen limiters have a more restrictive CFL number of $\frac{1}{6}$ in comparison to the moment limiter's CFL number of $\frac{3}{13}$, which explains the increased number of time steps for the Barth-Jespersen limiters in Table 7.

	ρ	s	p
$\mathbf{U}_l$	8	8.25	116.5
$\mathbf{U}_r$	1.4	0	1

TABLE 6. Density, normal speed, and pressure to the left and right of the shock in Example 4.6.

Limiter	Total solver run time (s)	Limiter run time (s)	Time steps	Time (ms)/ step
Moment	159	13.8 (8.6 %)	6,941	1.9 (-)
Barth-Jespersen (Edge)	191	18.2 (9.5 %)	8,305	2.1 (1.1x)
Barth-Jespersen (Vertex)	231	51.4 (22 %)	8,624	5.9 (3.1x)

TABLE 7. Run time comparisons for moment and Barth-Jespersen limiters. The number in brackets in the 'Limiter run time (s)' column is the percentage of the total solver run time that the limiting algorithm takes to execute. The number in brackets in the 'Time (ms)/ step' column is the speed up factor of the moment limiter compared to the Barth-Jespersen limiters per time step.



(a) Isovalues of the density.

(b) Zoom of the isovalues in neighborhood of the slipline.

FIGURE 15. Double Mach reflection problem.

5. Conclusion

We have proposed a new second order limiter that operates directly on solution coefficients by finding suitable directions in which the one-dimensional minmod operation can be employed. This is in contrast to existing limiters that sample solution values on the edges of triangles and require that they remain in some locally defined range. We derive a family of possible limiters and pick one that is easy to implement. We show on a number of numerical examples that the limiter produces second order accurate, stable numerical solutions, and performs comparably to a moment limiter on Cartesian grids.

This limiter is appealing because it has a stencil of constant size, it is easy to implement, much of its overhead is moved to the preprocessing stage, e.g., determination of the limiter stencil and interpolation coefficients, and it does not require computing the solution values at quadrature points.

Analysis for nonlinear fluxes is provided in the Appendix, though the resulting stability constraints are more involved. Traditionally, limiters developed for linear problems are applied to nonlinear systems. Thus, we do the same here.

Extending this limiter to nonconforming meshes for use in adaptive simulations appears to be possible. An element's limiting directions will stay the same when its neighbors are refined or coarsened, though the elements used in the reconstruction of forward and backward differences may have to be updated. This will not require projecting the neighboring solutions onto coarser or finer elements as is sometimes done. Another possibility for future work is the extension of this limiter to higher order bases, though the analysis is more involved as there are a higher number of solution coefficients that must be limited. An extension to three-dimensional computations would also be of practical interest. The advantages of a stencil of constant size become all the more important as the number of neighbors will grow in three dimensions. Finally, we plan on investigating the applicability of this limiter to computations involved in cut cell element geometries [7, 20].

6. Acknowledgment

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada grant 341373-07, and an Alexander Graham Bell PGS-D grant. We gratefully acknowledge the support of the NVIDIA Corporation with the donation of hardware used for this research.

Appendix A. Nonlinear Fluxes

For nonlinear fluxes, the RK-DG method requires third order accuracy to evaluate surface integrals in (9). With Gauss-Legendre numerical integration, we will need two quadrature points. We will modify our analysis to account for this. Most of the derivation follows the linear case.

Using the divergence theorem with (11), we obtain

$$\overline{U}_i^{n+1} = \overline{U}_i^n - \Delta t \sum_{j \in N_i^e} \frac{1}{2} \frac{|\partial \Omega_{i,j}|}{|\Omega_i|} \sum_{q=1,2} \left[\mathbf{F}^*(U_i^n(\mathbf{x}_{i,j,q}), U_j^n(\mathbf{x}_{i,j,q})) - \mathbf{F}^*(\overline{U}_i^n, \overline{U}_i^n) \right] \cdot \mathbf{n}_{i,j}.$$

Then, adding and subtracting $\mathbf{F}^*(U_i^n(\mathbf{x}_{i,j,q}), \overline{U}_i) \cdot \mathbf{n}_{i,j}$ in the inner sum, we have

$$\overline{U}_i^{n+1} = \overline{U}_i^n - \Delta t \sum_{j \in N_i^e} \sum_{q=1,2} \frac{1}{2} \frac{|\partial \Omega_{i,j}|}{|\Omega_i|} \Big[F_{i,j}(U_i^n(\mathbf{x}_{i,j,q}), \overline{U}_i^n) - F_{i,j}(\overline{U}_i^n, \overline{U}_i^n) + F_{i,j}(U_i^n(\mathbf{x}_{i,j,q}), U_j^n(\mathbf{x}_{i,j,q})) - F_{i,j}(U_i^n(\mathbf{x}_{i,j,q}), \overline{U}_i^n) \Big],$$

where $F_{i,j}(U_1, U_2) = \mathbf{F}(U_1, U_2) \cdot \mathbf{n}_{i,j}$. For $c^*_{i,j,q}$ belonging to the interval defined by $\overline{U}^n_i$ and $U^n_i(\mathbf{x}_{i,j,q})$, and $c^{**}_{i,j,q}$ belonging to the interval defined by $\overline{U}^n_i$ and $U^n_j(\mathbf{x}_{i,j,q})$, we have by the mean value theorem

$$\overline{U}^{n+1}_i = \overline{U}^n_i - \Delta t \sum_{j \in N^e_i} \sum_{q=1,2} \frac{1}{2} \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} \left[\frac{\partial F_{i,j}}{\partial U_1}(c^*_{i,j,q}, \overline{U}^n_i)(U^n_i(\mathbf{x}_{i,j,q}) - \overline{U}^n_i) + \frac{\partial F_{i,j}}{\partial U_2}(U^n_i(\mathbf{x}_{i,j,q}), c^{**}_{i,j,q})(U^n_j(\mathbf{x}_{i,j,q}) - \overline{U}^n_i) \right],$$

where $\frac{\partial F_{i,j}}{\partial U_1}$ and $\frac{\partial F_{i,j}}{\partial U_2}$ are the partial derivatives with respect to the first and second variables. We introduce the following coefficients

$$v^+_{i,j,q} = \Delta t \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} \frac{\partial F_{i,j}}{\partial U_1}(c^*_{i,j,q}, \overline{U}^n_i) \quad \text{and} \quad v^-_{j,i,q} = -\Delta t \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} \frac{\partial F_{i,j}}{\partial U_2}(U^n_i(\mathbf{x}_{i,j,q}), c^{**}_{i,j,q}), \tag{65}$$

that are non-negative by the assumed monotonicity of the flux. We now have

$$\overline{U}^{n+1}_i = \overline{U}^n_i + \sum_{j \in N^e_i} \sum_{q=1,2} \frac{1}{2} \left[v^-_{j,i,q}(U^n_j(\mathbf{x}_{i,j,q}) - \overline{U}^n_i) - v^+_{i,j,q}(U^n_i(\mathbf{x}_{i,j,q}) - \overline{U}^n_i) \right]. \tag{66}$$

After limiting, the solution average at time t^{n+1} is written

$$\overline{U}^{n+1}_i = \overline{U}^n_i + \sum_{j \in N^e_i} \sum_{q=1,2} \frac{1}{2} \left[v^-_{j,i,q}(\tilde{U}^n_j(\mathbf{x}_{i,j,q}) - \overline{U}^n_i) - v^+_{i,j,q}(\tilde{U}^n_i(\mathbf{x}_{i,j,q}) - \overline{U}^n_i) \right], \tag{67}$$

where the values $v^-_{j,i,q}$ and $v^+_{j,i,q}$ have been updated using (65), $\tilde{U}^n_i(\mathbf{x}_{i,j,q})$, and $\tilde{U}^n_j(\mathbf{x}_{i,j,q})$, and $\tilde{U}^n_i(\mathbf{x}_{i,j,q})$, and $\tilde{U}^n_j(\mathbf{x}_{i,j,q})$ are the limited numerical solutions on Ω_i and Ω_j, respectively. Following the linear case, we aim to rewrite the sums in (67) in the form

$$\sum_{q=1,2} \frac{1}{2} v^-_{j,i,q}(\tilde{U}^n_j(\mathbf{x}_{i,j,q}) - \overline{U}^n_i) = v^-_{j,i} \left[f_{j,i}(\overline{U}^n_j - \overline{U}^n_i) + f_{j,i,1}(U_{j,i,1} - \overline{U}^n_i) + f_{j,i,2}(U_{j,i,2} - \overline{U}^n_i) \right] \tag{68}$$

for the inflow term and

$$\sum_{q=1,2} \frac{1}{2} v^+_{i,j,q}(\tilde{U}^n_i(\mathbf{x}_{i,j,q}) - \overline{U}^n_i) = -v^+_{i,j} \left[g_{i,j,1}(U_{i,j,1} - \overline{U}^n_i) + g_{i,j,2}(U_{i,j,2} - \overline{U}^n_i) \right] \tag{69}$$

for the outflow term, with some non-negative multipliers $v^-_{j,i}$, $v^+_{i,j}$, $f_{j,i}$, $f_{j,i,1}$, $f_{j,i,2}$, $g_{i,j,1}$, and $g_{i,j,2}$.

A.1. **Inflow term.** First we consider (68). Analogous to (29), we replace the limited solution values with the limited forward differences to obtain

$$\begin{aligned} \sum_{q=1,2} \frac{1}{2} v^-_{j,i,q}(\tilde{U}^n_j(\mathbf{x}_{i,j,q}) - \overline{U}^n_i) = \frac{v^-_{j,i,1} + v^-_{j,i,2}}{2}(\overline{U}^n_j - \overline{U}^n_i) + \frac{v^-_{j,i,1}\varphi_1(\mathbf{r}_{j,i,1}) + v^-_{j,i,2}\varphi_1(\mathbf{r}_{j,i,2})}{2} l_{j,1} \frac{h_{j,1}}{6} \frac{U^f_{j,1} - \overline{U}^n_j}{d^f_{j,1}} \\ + \frac{v^-_{j,i,1}\varphi_2(\mathbf{r}_{j,i,1}) + v^-_{j,i,2}\varphi_2(\mathbf{r}_{j,i,2})}{2} l_{j,2} \frac{h_{j,2}}{4\sqrt{3}} \frac{U^f_{j,2} - \overline{U}^n_j}{d^f_{j,2}}. \end{aligned}$$

Let $v^-_{j,i} = \frac{v^-_{j,i,1}+v^-_{j,i,2}}{2}$, $v^{-,1}_{j,i} = \frac{v^-_{j,i,1}\varphi_1(\mathbf{r}_{j,i,1})+v^-_{j,i,2}\varphi_1(\mathbf{r}_{j,i,2})}{2}$, and $v^{-,2}_{j,i} = \frac{v^-_{j,i,1}\varphi_2(\mathbf{r}_{j,i,1})+v^-_{j,i,2}\varphi_2(\mathbf{r}_{j,i,2})}{2}$. If $v^-_{j,i}$ is nonzero, the above becomes

$$\sum_{q=1,2}\frac{1}{2}v^-_{j,i,q}(\tilde{U}^n_j(\mathbf{x}_{i,j,q})-\overline{U}^n_i) = v^-_{j,i}\left[(\overline{U}^n_j-\overline{U}^n_i)+l_{j,1}\frac{v^{-,1}_{j,i}}{v^-_{j,i}}\frac{h_{j,1}}{6}\frac{U^f_{j,1}-\overline{U}^n_j}{d^f_{j,1}}+l_{j,2}\frac{v^{-,2}_{j,i}}{v^-_{j,i}}\frac{h_{j,2}}{4\sqrt{3}}\frac{U^f_{j,2}-\overline{U}^n_j}{d^f_{j,2}}\right]. \tag{70}$$

We replace the forward difference with the backward difference if the sign of $v^{-,1}_{j,i}$ or $v^{-,2}_{j,i}$ is negative. We use the following shorthand notation

$$\alpha^-_{j,i,1} = \begin{cases} \frac{1}{6}\frac{h_{j,1}}{d^f_{j,1}}\frac{v^{-,1}_{j,i}}{v^-_{j,i}} \text{ if } v^{-,1}_{j,i}\geq 0 \\ \frac{1}{6}\frac{h_{j,1}}{d^b_{j,1}r_{j,1}}\frac{|v^{-,1}_{j,i}|}{v^-_{j,i}} \text{ otherwise.}\end{cases} \quad \text{and} \quad U^-_{j,i,1} = \begin{cases} U^f_{j,1} \text{ if } v^{-,1}_{j,i}\geq 0 \\ U^b_{j,1} \text{ otherwise.}\end{cases} \tag{71}$$

$$\alpha^-_{j,i,2} = \begin{cases} \frac{1}{4\sqrt{3}}\frac{h_{j,2}}{d^f_{j,2}}\frac{v^{-,2}_{j,i}}{v^-_{j,i}} \text{ if } v^{-,2}_{j,i}\geq 0 \\ \frac{1}{4\sqrt{3}}\frac{h_{j,2}}{d^b_{j,2}r_{j,2}}\frac{|v^{-,2}_{j,i}|}{v^-_{j,i}} \text{ otherwise,}\end{cases} \quad \text{and} \quad U^-_{j,i,2} = \begin{cases} U^f_{j,2} \text{ if } v^{-,2}_{j,i}\geq 0 \\ U^b_{j,2} \text{ otherwise.}\end{cases} \tag{72}$$

Then (70) becomes

$$\sum_{q=1,2}\frac{1}{2}v^-_{j,i,q}(\tilde{U}^n_j(\mathbf{x}_{i,j,q})-\overline{U}^n_i) = v^-_{j,i}\left[(\overline{U}^n_j-\overline{U}^n_i)+l_{j,1}\alpha^-_{j,i,1}(U^-_{j,i,1}-\overline{U}^n_j)+l_{j,2}\alpha^-_{j,i,2}(U^-_{j,i,2}-\overline{U}^n_j)\right].$$

Putting the above in the form (68), we have

$$\begin{aligned}\sum_{q=1,2}\frac{1}{2}v^-_{j,i,q}(\tilde{U}^n_j(\mathbf{x}_{i,j,q})-\overline{U}^n_i) = v^-_{j,i}\big[&(1-l_{j,1}\alpha^-_{j,i,1}-l_{j,2}\alpha^-_{j,i,2})(\overline{U}^n_j-\overline{U}^n_i) \\ &+l_{j,1}\alpha^-_{j,i,1}(U^-_{j,i,1}-\overline{U}^n_i)+l_{j,2}\alpha^-_{j,i,2}(U^-_{j,i,2}-\overline{U}^n_i)\big],\end{aligned} \tag{73}$$

with $f_{j,i} = 1-l_{j,1}\alpha^-_{j,i,1}-l_{j,2}\alpha^-_{j,i,2}$, $f_{j,i,1} = l_{j,1}\alpha^-_{j,i,1}$, and $f_{j,i,2} = l_{j,2}\alpha^-_{j,i,2}$.

Sum and non-negativity. The multipliers $f_{j,i,1}$ and $f_{j,i,2}$ are non-negative by (71) and (72). Requiring the coefficient $f_{j,i}$ to be non-negative gives the following condition

$$f_{j,i} = 1-l_{j,1}\alpha^-_{j,i,1}-l_{j,2}\alpha^-_{j,i,2}\geq 0 \quad \forall j\in N^e_i. \tag{74}$$

By definition, the sum of the coefficients is

$$f_{j,i}+f_{j,i,1}+f_{j,i,2} = 1. \tag{75}$$

A.2. **Outflow term.** We now consider (69). Let $v_{i,j}^{+} = \frac{v_{i,j,1}^{+}+v_{i,j,2}^{+}}{2}$, $v_{i,j}^{+,1} = \frac{v_{i,j,1}^{+}\varphi_1(\mathbf{r}_{i,j,1})+v_{i,j,2}^{+}\varphi_1(\mathbf{r}_{i,j,2})}{2}$, and $v_{i,j}^{+,2} = \frac{v_{i,j,1}^{+}\varphi_2(\mathbf{r}_{i,j,1})+v_{i,j,2}^{+}\varphi_2(\mathbf{r}_{i,j,2})}{2}$. We replace the limited solution coefficients with the limited forward differences and if $v_{i,j}^{+}$ is nonzero, we obtain

$$\sum_{q=1,2} \frac{1}{2} v_{i,j,q}^{+}(\tilde{U}_i^n(\mathbf{x}_{i,j,q}) - \overline{U}_i^n) = v_{i,j}^{+} \left(\frac{v_{i,j}^{+,1}}{v_{i,j}^{+}} l_{i,1} \frac{h_{i,1}}{6} \frac{U_{i,1}^f - \overline{U}_i^n}{d_{i,1}^f} + \frac{v_{i,j}^{+,2}}{v_{i,j}^{+}} l_{i,2} \frac{h_{i,2}}{4\sqrt{3}} \frac{U_{i,2}^f - \overline{U}_i^n}{d_{i,2}^f} \right).$$

We introduce the following variables for convenience

$$\alpha_{i,j,1}^{+} = \begin{cases} \frac{1}{6}\frac{h_{i,1}}{d_{i,1}^f}\frac{|v_{i,j}^{+,1}|}{v_{i,j}^{+}} \text{ if } v_{i,j}^{+,1} \le 0 \\ \frac{1}{6}\frac{h_{i,1}}{d_{i,1}^b r_{i,1}}\frac{v_{i,j}^{+,1}}{v_{i,j}^{+}} \text{ otherwise.} \end{cases} \quad \text{and} \quad U_{i,j,1}^{+} = \begin{cases} U_{i,1}^f \text{ if } v_{i,j}^{+,1} \le 0 \\ U_{i,1}^b \text{ otherwise,} \end{cases} \tag{76}$$

$$\alpha_{i,j,2}^{+} = \begin{cases} \frac{1}{4\sqrt{3}}\frac{h_{i,2}}{d_{i,2}^f}\frac{|v_{i,j}^{+,2}|}{v_{i,j}^{+}} \text{ if } v_{i,j}^{+,2} \le 0 \\ \frac{1}{4\sqrt{3}}\frac{h_{i,2}}{d_{i,2}^b r_{i,2}}\frac{v_{i,j}^{+,2}}{v_{i,j}^{+}} \text{ otherwise.} \end{cases} \quad \text{and} \quad U_{i,j,2}^{-} = \begin{cases} U_{i,2}^f \text{ if } v_{i,j}^{+,2} \le 0 \\ U_{i,2}^b \text{ otherwise.} \end{cases} \tag{77}$$

In the form (69), the above expansion is

$$\sum_{q=1,2} \frac{1}{2} v_{i,j,q}^{+}(\tilde{U}_i^n(\mathbf{x}_{i,j,q}) - \overline{U}_i^n) = -v_{i,j}^{+} \left[l_{i,1}\alpha_{i,j,1}^{+}(U_{i,j,1}^{+} - \overline{U}_i^n) + l_{i,2}\alpha_{i,j,2}^{+}(U_{i,j,2}^{+} - \overline{U}_i^n) \right]. \tag{78}$$

with $g_{i,j,1} = l_{i,1}\alpha_{i,j,1}^{+}$ and $g_{i,j,2} = l_{i,2}\alpha_{i,j,2}^{+}$.

Sum and non-negativity. The multipliers $g_{i,j,1}$ and $g_{i,j,2}$ are non-negative by (76) and (77). The sum of the coefficients is given by

$$g_{i,j,1} + g_{i,j,2} = l_{i,1}\alpha_{i,j,1}^{+} + l_{i,2}\alpha_{i,j,2}^{+}. \tag{79}$$

A.3. **Putting it all together.** The inflow terms and outflow terms have been expanded into sums of the form (68) and (69) in (73) and (78), respectively. Substituting these sums into the scheme (67) gives

$$\begin{aligned} \overline{U}_i^{n+1} = \overline{U}_i^n + \sum_{j \in N_i^e} v_{j,i}^{-} \left[f_{j,i}(\overline{U}_j^n - \overline{U}_i^n) + f_{j,i,1}(U_{j,i,1}^{-} - \overline{U}_i^n) + f_{j,i,2}(U_{j,i,2}^{-} - \overline{U}_i^n) \right] \\ + v_{i,j}^{+} \left[g_{i,j,1}(U_{i,j,1}^{+} - \overline{U}_i^n) + g_{i,j,2}(U_{i,j,2}^{+} - \overline{U}_i^n) \right]. \end{aligned}$$

This is of the form (19). We require that the sum of the coefficients in front of the differences above is less than or equal to one. Using (75) and (79), we write this requirement as

$$\sum_{j\in N_i^e} v_{j,i}^- \left[f_{j,i} + f_{j,i,1} + f_{j,i,2}\right] + v_{i,j}^+ \left[g_{i,j,1} + g_{i,j,2}\right] = \sum_{j\in N_i^e} v_{j,i}^- + v_{i,j}^+ \left(l_{i,1}\alpha_{i,j,1}^+ + l_{i,2}\alpha_{i,j,2}^+\right) \leq 1. \tag{80}$$

For (80) to be satisfied, we enforce on each edge of Ω_i

$$l_{i,1}\alpha_{i,j,1}^+ + l_{i,2}\alpha_{i,j,2}^+ \leq 1 \quad \forall j \in N_i^e, \tag{81}$$

and the time step constraint

$$\sum_{j\in N_i^e} \left(v_{j,i}^- + v_{i,j}^+\right) \leq 1, \quad \forall \Omega_i. \tag{82}$$

For non-negativity of the expansion coefficient $f_{j,i}$ we must enforce the constraint (74) on all the edges of Ω_i

$$l_{j,1}\alpha_{j,i,1}^- + l_{j,2}\alpha_{j,i,2}^- \leq 1, \quad \forall j \in N_i^e. \tag{83}$$

A.4. **Time step restriction.** Using the definition of $v_{j,i}^-$ and $v_{i,j}^+$ in (65), (82) becomes

$$\Delta t \sum_{j\in N_i^e} \sum_{q=1,2} \frac{1}{2} \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} \left(\frac{\partial F_{i,j}}{\partial U_1}(c_{i,j,q}^*, \overline{U}_i^n) - \frac{\partial F_{i,j}}{\partial U_2}(U_i^n(\mathbf{x}_{i,j,q}), c_{i,j,q}^{**}) \right) \leq 1. \tag{84}$$

Due to the differentiability of the numerical flux, there exists a λ_i such that $\frac{\partial F_{i,j}}{\partial U_1}(c_{i,j,q}^*, \overline{U}_i^n) \leq \lambda_i$ and $-\frac{\partial F_{i,j}}{\partial U_2}(U_i^n(\mathbf{x}_{i,j,q}), c_{i,j,q}^{**}) \leq \lambda_i$ hold. The time step restriction in (84) now becomes

$$2\Delta t \lambda_i \sum_{j\in N_i^e} \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} \leq 1. \tag{85}$$

Note that the radius of the circle inscribed in Ω_i is

$$h_i = 2\frac{|\Omega_i|}{|\partial\Omega_i|}, \tag{86}$$

where $|\partial\Omega_i|$ and $|\Omega_i|$ are the perimeter and area of Ω_i, respectively. The nonlinear time step restriction in terms of the inscribed circle is

$$\Delta t \leq \frac{1}{4}\frac{h_i}{\lambda_i}. \tag{87}$$

References

[1] P. K. Sweby, "High resolution schemes using flux limiters for hyperbolic conservation laws," *SIAM Journal on Numerical Analysis*, vol. 21, no. 5, pp. 995–1011, 1984.

[2] A. Harten, "High resolution schemes for hyperbolic conservation laws," *Journal of Computational Physics*, vol. 49, no. 3, pp. 357–393, 1983.

[3] A. Harten, "On a class of high resolution total-variation-stable finite-difference schemes," *SIAM Journal on Numerical Analysis*, vol. 21, no. 1, pp. 1–23, 1984.

[4] E. Tadmor, "Convenient total variation diminishing conditions for nonlinear difference schemes," *SIAM Journal on Numerical Analysis*, vol. 25, no. 5, pp. 1002–1014, 1988.

[5] J. B. Goodman and R. J. LeVeque, "On the accuracy of stable schemes for 2D scalar conservation laws," *Mathematics of Computation*, pp. 15–21, 1985.

[6] T. Barth and D. Jespersen, "The design and application of upwind schemes on unstructured meshes.," *AIAA paper*, pp. 89–0366, 1989.

[7] S. May and M. Berger, "Two-dimensional slope limiters for finite volume schemes on non-coordinate-aligned meshes," *SIAM Journal on Scientific Computing*, vol. 35, no. 5, pp. A2163–A2187, 2013.

[8] T. Buffard and S. Clain, "Monoslope and multislope MUSCL methods for unstructured meshes," *Journal of Computational Physics*, vol. 229, no. 10, pp. 3745–3776, 2010.

[9] C. Le Touze, A. Murrone, and H. Guillard, "Multislope MUSCL method for general unstructured meshes," *Journal of Computational Physics*, vol. 284, pp. 389–418, 2015.

[10] H. Hoteit, P. Ackerer, R. Mosé, J. Erhel, and B. Philippe, "New two-dimensional slope limiters for discontinuous Galerkin methods on arbitrary meshes," *International Journal for Numerical Methods in Engineering*, vol. 61, no. 14, pp. 2566–2593, 2004.

[11] B. Cockburn, S. Hou, and C.-W. Shu, "The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case," *Mathematics of Computation*, vol. 54, no. 190, pp. 545–581, 1990.

[12] R. Biswas, K. D. Devine, and J. E. Flaherty, "Parallel, adaptive finite element methods for conservation laws," *Applied Numerical Mathematics*, vol. 14, no. 1-3, pp. 255–283, 1994.

[13] L. Krivodonova, "Limiters for high-order discontinuous Galerkin methods," *Journal of Computational Physics*, vol. 226, no. 1, pp. 879–896, 2007.

[14] N. Chalmers and L. Krivodonova, "A robust CFL condition for the discontinuous Galerkin method on triangular meshes," *Draft at* `http://www.math.uwaterloo.ca/~lgk/A_Characteristic_Based_CFL_Number.pdf`.

[15] S. Gottlieb, D. Ketcheson, and C.-W. Shu, *Strong Stability Preserving Runge-Kutta and Multistep Time Discretizations*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2011.

[16] M. Fuhry, A. Giuliani, and L. Krivodonova, "Discontinuous Galerkin methods on graphics processing units for nonlinear hyperbolic conservation laws," *International Journal for Numerical Methods in Fluids*, vol. 76, no. 12, pp. 982–1003, 2014.

[17] A. Giuliani and L. Krivodonova, "Face coloring in unstructured CFD codes," *Parallel Computing*, vol. 63, pp. 17–37, 2017.

[18] P. Woodward and P. Colella, "The numerical simulation of two-dimensional fluid flow with strong shocks," *Journal of Computational Physics*, vol. 54, no. 1, pp. 115–173, 1984.

[19] A. Giuliani and L. Krivodonova, "Analysis of slope limiters on unstructured triangular meshes," *Journal of Computational Physics*, 2018. https://doi.org/10.1016/j.jcp.2018.07.031.

[20] R. Qin and L. Krivodonova, "A discontinuous Galerkin method for solutions of the Euler equations on Cartesian grids with embedded geometries," *Journal of Computational Science*, vol. 4, no. 1, pp. 24–35, 2013.