Analysis of slope limiters on unstructured triangular meshes

Andrew Giuliani[a], Lilia Krivodonova[a,*]

[a]*Department of Applied Mathematics, University of Waterloo*

Abstract

We analyze the stability and accuracy of second order limiters for the discontinuous Galerkin method on unstructured triangular meshes. We derive conditions for a limiter such that the numerical solution preserves second order accuracy and satisfies the local maximum principle. This leads to a new measure of cell size that is approximately twice as large as the radius of the inscribed circle. It is shown with numerical experiments that the resulting bound on the time step is tight. Finally, we consider various combinations of limiting points and limiting neighborhoods and present numerical experiments comparing the accuracy, stability, and efficiency of the corresponding limiters.

1. Introduction

Hyperbolic conservation laws are a class of partial differential equations that model wave propagation. Weak solutions of such equations admit discontinuities, which can lead to nonphysical oscillations when high order numerical methods are employed. A popular technique to stabilize the growth of these oscillations for methods that are formally second order accurate is slope limiting. The gradient is computed directly by differentiating a polynomial solution, e.g. in Galerkin methods, or reconstructed using neighboring solution means, e.g. finite volume (FV) methods. A limiting algorithm will modify, or limit, this gradient so that the solution at suitable points belongs to a specified, local range [1].

For one-dimensional problems, slope limiters that ensure a total variation diminishing (TVD) property are frequently used [2, 3, 4, 5]. In two dimensions, enforcing a TVD property can lead to at most first order schemes [6]. A different requirement on the numerical solution is enforcement of a local maximum principle, studied in [7] on two-dimensional structured grids for steady state computations. This idea is used in [8] to reconstruct non-oscillatory gradients on unstructured meshes of triangles. This limiter is quite popular due to its ease of implementation and computational simplicity. It consists in writing the numerical solution as a sum of the cell mean and slope. The slope is then reduced by a scalar between 0 and 1 such that the numerical solution at predetermined points lies in a locally defined interval. Some

*Corresponding author

Email address: lgk@uwaterloo.ca (Lilia Krivodonova)

limiters modify the x and y components of the gradient separately, e.g. [9], by solving a small linear program on each element. Slope limiters can operate on solution values at the edge midpoints [10], at the neighboring cell centroids [9], or cell vertices as in [11, 12, 13]. In contrast to the above methods, classified as monoslope methods, multislope methods have also been studied whereby the solution is reconstructed and limited independently at each face of the element [14, 15].

Much literature on limiters has been devoted to finite volume methods. When transitioning to the discontinuous Galerkin (DG) method, often the same limiters are applied. However, second order limiters for the discontinuous Galerkin method have been presented in, e.g. [16] for one-dimensional problems and [17] for multidimensional problems. The limiter proposed in [18] requires precomputation of several mesh-dependent geometric parameters on each cell, which increases computational complexity. This explains the popularity of coupling the DG method with the so-called Barth-Jespersen limiter [8]: no geometric data needs to be precomputed, and the limiter does not require a stencil larger than that of the DG method. Another second order limiter for the DG method on triangles was presented in [19] and requires solving an optimization problem.

Classical limiters operate only on the linear approximations to the solution. Limiters that work on higher than second order accurate approximations are needed and a significant effort has been placed into finding such limiters. In [20, 21], the idea of moment limiters was proposed, whereby the numerical solution's dth derivative is limited using the $(d-1)$th derivatives on neighboring cells. Generalizations of the moment limiter to unstructured meshes were studied in [22, 23]. Different approaches to high order limiting were described in [24, 25].

In this work, we analyze the Barth-Jespersen limiter [8] on two-dimensional unstructured meshes of triangles, applied to linear and nonlinear problems using the DG method. This limiter has been addressed in [26] for finite volume methods, but not for the DG method. Despite its popularity, we argue that in its simplest form, it is not a well performing limiter for the DG method.

The simplest implementation of the Barth-Jespersen limiter uses the edge neighborhood and edge midpoints as limiting points. With these choices, we show that unstructured meshes are unlikely to yield second order accurate numerical solutions, defeating the purpose of high resolution numerical methods. For these meshes, we show that the way a refinement study is conducted will influence the observed rate of convergence of the solution. For example, refinement obtained by tiling the initial mesh or remeshing the domain at a reduced cell size can yield first order convergence. One may observe second order accuracy in

the L_1 norm with nested refinement, but first order accuracy is still observed in the L_∞ norm. We prove that a remedy of this problem is to choose an alternative limiting neighborhood, such that the limiting points lie in the admissibility region that we define.

In our analysis of second order limiters applied to DG, we address two issues: stability and accuracy. For stability, we have proven that the numerical solution for scalar equations will satisfy a local maximum principle that ensures L_∞ stability, provided a suitable time step restriction is enforced. This new time step restriction follows from the stability analysis, and uses a new measure of cell size, which is the cell width in the direction of flow. We show with numerical experiments that this time step restriction is tight. The new measure is approximately double the radius of the inscribed circle, typically used with maximum principle limiters and the DG method. As a result, the maximum allowable time step doubles and the amount of computational work halves.

From our analysis, we find the range to which the cell means of the solution at the next time step will belong, provided the above time step restriction is enforced. This range is determined by the solution averages on nearby elements, i.e. on the neighbors used in the limiting procedure. There is freedom in defining this neighborhood, e.g. we can choose the elements that share edges or we can choose elements that share vertices with the element being limited [27]. These neighborhoods are the most natural ones, though others are possible, e.g. the entire mesh. We find that smaller neighborhoods introduce too much numerical diffusion. In particular, limiting with the edge neighborhood is too diffusive. On the other hand, if the neighborhood has a large and variable size, e.g. vertex neighborhood on an unstructured grid, this can yield almost a threefold increase in the time spent executing the limiting subroutines. This has implications for limiters that use vertex-type neighborhoods [11, 13, 24, 28].

The other aspect of the limiting algorithm is the choice of points at which the numerical solution is checked for overshoots, i.e. the algorithm's limiting points. In this work, we study the one- and two-point Gauss-Legendre quadrature nodes as limiting points. Checking for oscillations at quadrature points comes naturally in the DG implementation. This is because the basis functions at these points are often precomputed, therefore solution values can be obtained efficiently. Other choices are theoretically possible though seldom done in practice. We have proven that one- and two-point limiting are sufficient for the stability of linear and nonlinear problems, respectively. Two-point limiting may lead to first order accuracy and catastrophically diffusive solutions on edge neighborhoods. Numerical experiments verify that two-point limiting is more diffusive than one-point limiting on all neighborhoods, though the difference is small

for the vertex neighborhood. While the one-point limiter with nonlinear fluxes will not guarantee that the minimum and maximum of the solution are maintained, for all problems that we considered, the growth in the means was small. Finally, we find that the number of limiting points does not affect code run time as drastically as the size of the neighborhood. In the numerical experiments section (Section 8), we discuss which combination of limiting points and neighborhoods should be used.

2. The discontinuous Galerkin method

In two spatial dimensions, hyperbolic conservation laws are partial differential equations (PDEs) of the form

$$\mathbf{u}_t + \nabla \cdot \mathbf{F}(\mathbf{u}) = 0, \tag{1}$$

with the solution $\mathbf{u}(\mathbf{x}, t) = (u_1, u_2, ..., u_M)^\intercal$ defined on $\Omega \times [0, T]$ such that $\mathbf{x} = (x, y)$, $\mathbf{x} \in \Omega \subset \mathbb{R}^2$, T is the final time and $\mathbf{F}(\mathbf{u}) = (\mathbf{F}_1(\mathbf{u}), \mathbf{F}_2(\mathbf{u}))$ is the flux function. Additionally, the initial condition along with appropriate boundary conditions are prescribed.

The discontinuous Galerkin method can be formulated by first dividing the domain Ω into an unstructured mesh of triangles such that $\Omega = \bigcup_i \Omega_i$. We define $S(\Omega_i)$ to be the space of linear polynomials on Ω_i, and $\{\phi_{i,k}\}_{k=0,1,2}$ to be the set of orthonormal basis functions on $S(\Omega_i)$. The weak form of the conservation law is obtained by multiplying equation (1) by a test function $v \in H^1(\Omega_i)$ and integrating on element Ω_i. After applying the divergence theorem, we obtain

$$\int_{\Omega_i} \mathbf{u}_t v d\mathbf{x} - \int_{\Omega_i} \mathbf{F}(\mathbf{u}) \cdot \nabla v d\mathbf{x} + \int_{\partial\Omega_i} v \mathbf{F}(\mathbf{u}) \cdot \mathbf{n} dl = 0, \ \forall v \in H^1(\Omega_i), \tag{2}$$

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

The exact solution on element Ω_i is approximated by $\mathbf{U}_i$, which is a linear combination of the basis functions $\phi_{i,k}$, i.e. $\mathbf{U}_i = \sum_{k=0}^{2} \mathbf{c}_{i,k}\phi_{i,k}$, where $\mathbf{c}_{i,k} = [c_{i,k}^1, c_{i,k}^2, \ldots, c_{i,k}^m, \ldots, c_{i,k}^M]^\intercal$ are referred to as the modal degrees of freedom. 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 $\phi_{i,k}$, equation (2) now becomes

$$\frac{d}{dt}\mathbf{c}_{i,k} = \int_{\Omega_i} \mathbf{F}(\mathbf{U}_i) \cdot \nabla\phi_{i,k} \, d\mathbf{x} - \sum_{j \in N_i^e, j \neq i} \int_{\partial\Omega_{i,j}} \phi_{i,k} \mathbf{F}^*(\mathbf{U}_i, \mathbf{U}_j) \cdot \mathbf{n}_{i,j} \, dl, \quad k = 0, 1, 2, \tag{3}$$

where $\partial\Omega_{i,j}$ is the edge shared by Ω_i and Ω_j, N_i^e is the set of Ω_i and elements sharing an edge with Ω_i, and $\mathbf{n}_{i,j}$ is the outward pointing unit normal on that edge. We use numerical quadrature to evaluate the integrals in (3). The system of equations (3) can be solved in time using a standard ODE solver, e.g. a Runge-Kutta (RK) method. In the presence of discontinuities in the exact solution, the numerical solution can develop nonphysical oscillations that may lead to numerical instability. To suppress oscillations and stabilize the solution a limiter will be used.

3. Limiting algorithm

With the following limiting algorithm, we seek to enforce the local maximum principle. The numerical solution satisfies the local maximum principle if

$$\min_{j\in N_i} \overline{U}_j^n \leq \overline{U}_i^{n+1} \leq \max_{j\in N_i} \overline{U}_j^n, \tag{4}$$

where N_i is a set containing the index of Ω_i and the indices of elements in the neighborhood of Ω_i, and $\overline{U}_i^n$ is the cell average.

We previously defined the numerical solution in terms of basis functions and degrees of freedom. Here, we rewrite it in terms of the cell average and slope at time step n:

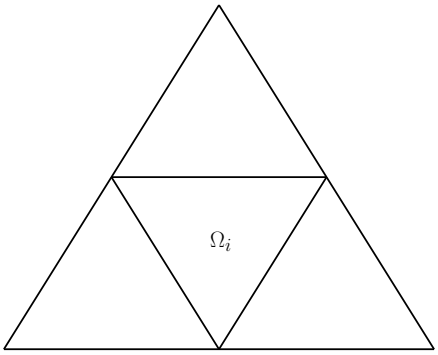
$$U_i^n(\mathbf{x}) = \overline{U}_i^n + \nabla U_i^n \cdot (\mathbf{x} - \mathbf{x}_i), \tag{5}$$

where $\mathbf{x}_i$ is the centroid of Ω_i, $\overline{U}_i^n$ is the cell average, and ∇U_i^n is the solution gradient. The limiting procedure applied to the numerical solution on Ω_i multiplies the gradient by a coefficient α_i, with the aim to enforce the maximum principle (4) on the means at the next time step. The limited solution $\tilde{U}_i^n(\mathbf{x})$ is of the form

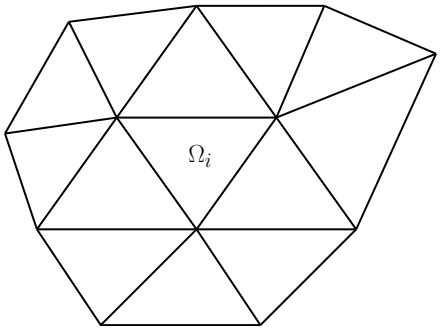
$$\tilde{U}_i^n(\mathbf{x}) = \overline{U}_i^n + \alpha_i \nabla U_i^n \cdot (\mathbf{x} - \mathbf{x}_i). \tag{6}$$

Limiting is done by comparing the values of $U_i(\mathbf{x})$ to the solution averages on neighboring elements, where the points $\mathbf{x}$ can be quadrature points, element vertices, edge midpoints, or other. We refer to these points as limiting points. If the solution at the limiting points falls outside of the range defined by its neighbors, its slope is reduced by α_i.

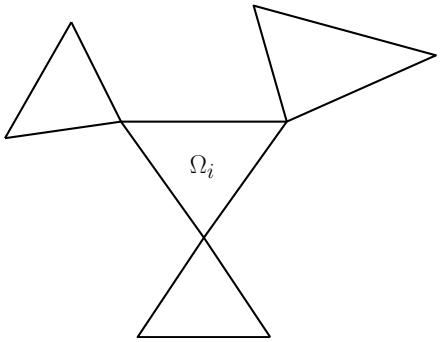
We collect the indices of the elements used in limiting the slope on Ω_i in a set. As with limiting



(a) Edge-neighborhood N_i^e: Ω_i and all elements that share an edge with Ω_i.



(b) Vertex-neighborhood N_i^v: Ω_i and all elements that share a vertex with Ω_i.



(c) Reduced neighborhood N_i^r: Ω_i and three vertex neighbors.

Figure 1: Edge, vertex, and reduced neighborhoods of Ω_i.

points, there is freedom in choosing a suitable neighborhood of Ω_i. For example, the edge neighborhood is comprised of Ω_i itself and all the elements that share an edge with it, we refer to the set of these indices as N_i^e. The vertex neighborhood is comprised of Ω_i itself and all elements that share a vertex with it, we refer to the set of these indices as N_i^v. We can also choose a reduced subset of N_i^v, which we refer to as N_i^r. These neighborhoods are illustrated in Figure 1.

We execute the following algorithm to compute α_i:

1. Compute the minimum and maximum cell means on the elements in N_i:

$$m_i^n = \min_{j \in N_i} \overline{U}_j^n \quad \text{and} \quad M_i^n = \max_{j \in N_i} \overline{U}_j^n. \tag{7}$$

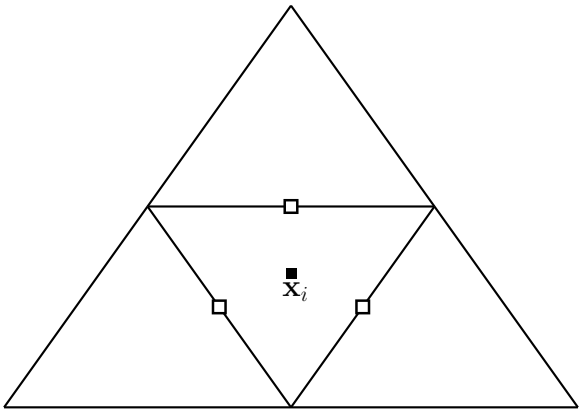
2. Compute the coefficient $y_i(\mathbf{x}_l)$ at each limiting point $\mathbf{x}_l$

$$y_i(\mathbf{x}_l) = \begin{cases} \frac{M_i^n - \overline{U}_i^n}{U_i^n(\mathbf{x}_l) - \overline{U}_i^n}, & \text{if } U_i^n(\mathbf{x}_l) - \overline{U}_i^n > 0, \\ \frac{m_i^n - \overline{U}_i^n}{U_i^n(\mathbf{x}_l) - \overline{U}_i^n}, & \text{if } U_i^n(\mathbf{x}_l) - \overline{U}_i^n < 0, \\ 1, & \text{otherwise.} \end{cases}$$

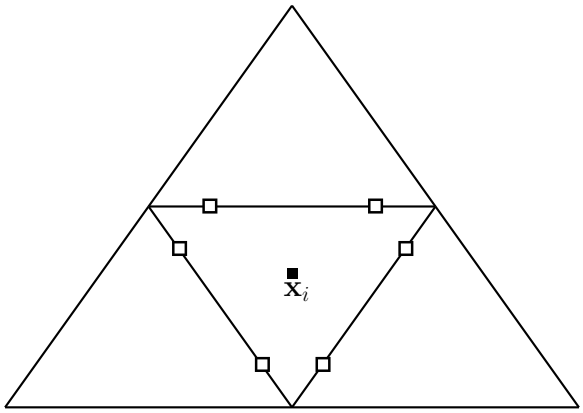
3. Find the smallest $y_i(\mathbf{x}_l)$ on Ω_i

$$y_i = \min_l y_i(\mathbf{x}_l).$$

4. If $y_i \in (0, 1)$, then the solution is outside the locally defined range, $[m_i^n, M_i^n]$, for at least one limiting point. Scaling the gradient by $\alpha_i = y_i$ brings that value into the prescribed range. If $y_i > 1$, then



(a) One-point Gauss-Legendre quadrature rule.



(b) Two-point Gauss-Legendre quadrature rule.

Figure 2: Nodes of quadrature rules for edge integrals.

the solution at the limiting points lies in the range, i.e. the current slope is acceptable and should not be modified, so $\alpha_i = 1$. Combining the above into one formula, we have

$$\alpha_i = \min(y_i, 1).$$

5. The limited numerical solution $\tilde{U}_i^n$ is now given by (6).

After limiting, the polynomial $\tilde{U}_i^n(\mathbf{x})$ is rewritten in terms of the basis functions and the simulation continues.

4. Time integration

We propagate (3) in time using an explicit two-stage second order Runge-Kutta (RK) method, known as Heun's method. For a system of ODEs of the form

$$\frac{d}{dt}\mathbf{c} = L(\mathbf{c}),$$

the time stepping scheme, with a limiter, is given by Algorithm 1.

Algorithm 1 SSP-RK2 algorithm.

$\mathbf{c}^{(1)} = \mathbf{c}^n + \Delta t L(\mathbf{c}^n)$
Limit $\mathbf{c}^{(1)}$
$\mathbf{c}^{(2)} = \mathbf{c}^{(1)} + \Delta t L(\mathbf{c}^{(1)})$
$\mathbf{c}^{n+1} = \frac{1}{2}\mathbf{c}^n + \frac{1}{2}\mathbf{c}^{(2)}$
Limit $\mathbf{c}^{n+1}$

In the algorithm above, we limit the intermediate RK stage and the solution at level t^{n+1}. The stability results we prove in the next section concern one forward Euler time step, which is only first order accurate in time. The presented analysis extends to a special subset of RK methods, called Strong Stability Preserving (SSP) schemes. This is because such methods can be written as a convex combination of forward Euler steps [29]. Since each forward Euler step does not introduce new extrema, a convex combination of them will not either. Note that Heun's method is SSP.

5. Stability

We now prove stability of the DG scheme coupled with the limiter (6) under a suitable time step constraint for linear and nonlinear equations. From (3), with the test function $\phi_{i,0} = |\Omega_i|^{-\frac{1}{2}}$, where $|\Omega_i|$ is the area of the cell, we obtain the ordinary differential equation for propagation of the mode corresponding to the constant basis function, $c_{i,0}$,

$$\frac{d}{dt}c_{i,0} = -\frac{1}{\sqrt{|\Omega_i|}} \sum_{j \in N_i^e, j \neq i} \int_{\partial\Omega_{i,j}} \mathbf{F}^*(U_i(\mathbf{x}), U_j(\mathbf{x})) \cdot \mathbf{n} \; dl.$$

Multiplying both sides of the equation by $\phi_{i,0}$, recalling the orthonormal property of the basis, and using $\overline{U}_i = c_{i,0}\phi_{i,0}$, we have

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

We apply one forward Euler time step to the equation above, and the scheme for the cell average on Ω_i becomes

$$\overline{U}_i^{n+1} = \overline{U}_i^n - \frac{\Delta t}{|\Omega_i|} \sum_{j \in N_i^e, j \neq i} \int_{\partial\Omega_{i,j}} \mathbf{F}^*(U_i^n(\mathbf{x}), U_j^n(\mathbf{x})) \cdot \mathbf{n} \; dl. \tag{8}$$

In the case of nonlinear fluxes $\mathbf{F}(u)$, the DG method needs to integrate the boundary integral with third order accuracy [17]. An efficient choice of approximation is the two-point Gauss-Legendre quadrature rule, with $\mathbf{x}_{i,j,q}$ being the qth quadrature point on $\partial\Omega_{i,j}$. Replacing the boundary integral in (8) with the quadrature rule gives

$$\overline{U}_i^{n+1} = \overline{U}_i^n - \Delta t \sum_{j \in N_i^e, j \neq i} \frac{1}{2} \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} \sum_{q=1,2} \mathbf{F}^*(U_i^n(\mathbf{x}_{i,j,q}), U_j^n(\mathbf{x}_{i,j,q})) \cdot \mathbf{n}_{i,j}, \tag{9}$$

where $|\partial\Omega_{i,j}|$ is the length of $\partial\Omega_{i,j}$. For a linear flux, this becomes

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

where $\mathbf{x}_{i,j}$ is the midpoint of the edge shared by Ω_i and Ω_j. Before presenting the main result, we state the following proposition, the proof of which is provided in Appendix A.

Proposition 1. *For a quadrature point* $\mathbf{x}$*, there exists a multiplier* $0 \le r \le 2$ *and another quadrature point* $\mathbf{x}'$ *on a different edge, such that*

$$U_i(\mathbf{x}) - \overline{U}_i = r(\overline{U}_i - U_i(\mathbf{x}')).$$

For schemes (9) and (10), we have the following maximum principle result.

Theorem 1. *Let* $m_i' = \min_{k\in N_i^e} m_k^n$ *and* $M_i' = \max_{k\in N_i^e} M_k^n$*, where* m_k^n *and* M_k^n *are given by* (7)*. If* $m_i^n \le U_i^n(\mathbf{x}) \le M_i^n$ *for all quadrature points* $\mathbf{x}_{i,j}$ *in* (10) *or* $\mathbf{x}_{i,j,q}$ *in* (9)*, and* Δt *is subject to the CFL constraint*

$$\Delta t \le \frac{1}{6}\min_i \frac{h_{c,i}}{\lambda_i}, \tag{11}$$

where $h_{c,i}$ *is the radius of the inscribed circle of* Ω_i*, and* λ_i *is the magnitude of the wave speed on* Ω_i*, then*

$$\overline{U}_i^{n+1} \in [m_i', M_i']. \tag{12}$$

That is, the schemes (9) *and* (10) *satisfy a local maximum principle.*

Proof. The proof consists of three steps. First, we write the solution mean at t^{n+1}, $\overline{U}_i^{n+1}$, in the following form

$$\overline{U}_i^{n+1} = d_i\overline{U}_i^n + \sum d_j U_j^n(\mathbf{x}), \tag{13}$$

where the sum is over all edge quadrature points $\mathbf{x}$, and $U_j^n(\mathbf{x})$ are understood to be the solution values from either inside or outside the element Ω_i. Next, we show that under the CFL constraint (11), the coefficients d_j are non-negative, and their sum is equal to 1, i.e. they have

1. sum property:

$$d_i + \sum d_j = 1, \tag{14}$$

2. non-negativity property:

$$d_j \geq 0. \tag{15}$$

This means that $\overline{U}_i^{n+1}$ in (13) is a convex combination of solution values at t^n. Upon application of the limiter (6), these values will be bounded, i.e. we have

3. limiting property:

$$U_j^n(\mathbf{x}) \in \left[\min_{k\in N_j} \overline{U}_k^n, \max_{k\in N_j} \overline{U}_k^n\right] = [m_j^n, M_j^n],$$

where $\mathbf{x}$ is understood to be an edge quadrature point. Finally, if the conditions in properties 1, 2, and 3 are satisfied, then the bounds (12) on $\overline{U}_i^{n+1}$ directly follow. We now prove the theorem for linear problems, i.e. (1) with linear fluxes.

Linear problems. For linear problems we use the upwind numerical flux, which is given by

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

where $N_i^{e,-}$ and $N_i^{e,+}$ are the sets of inflow and outflow neighbors, respectively, i.e. $N_i^{e,\pm} = \{j : j \in N_i^e, j \neq i, \text{ such that } \pm\,\mathbf{a}\cdot\mathbf{n}_{i,j} > 0\}$. Therefore, scheme (10) becomes

$$\overline{U}_i^{n+1} = \overline{U}_i^n + \Delta t \sum_{j\in N_i^{e,-}} |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} U_j^n(\mathbf{x}_{i,j}) - \Delta t \sum_{j\in N_i^{e,+}} |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} U_i^n(\mathbf{x}_{i,j}). \tag{16}$$

By the divergence theorem, we have the following relation

$$\sum_{j\in N_i^e, j\neq i} |\partial\Omega_{i,j}| \mathbf{a}\cdot\mathbf{n}_{i,j} = 0. \tag{17}$$

Using (17) in (16), we have

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

Applying Proposition 1 to the outflow terms in the previous equation, we obtain

$$\overline{U}_i^{n+1} = \overline{U}_i^n + \Delta t \sum_{j \in N_i^{e,-}} |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|}(U_j^n(\mathbf{x}_{i,j}) - \overline{U}_i^n) - \Delta t \sum_{j \in N_i^{e,+}} |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} r_{i,j}(\overline{U}_i^n - U_i^n(\mathbf{x}_{i,j'})),$$

where $r_{i,j}$ is the scaling coefficient r on edge $\partial\Omega_{i,j}$. Grouping terms allows us to write the above equation in the form (13):

$$\begin{aligned}\overline{U}_i^{n+1} = \left(1 - \Delta t \sum_{j \in N_i^{e,-}} |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} - \Delta t \sum_{j \in N_i^{e,+}} |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} r_{i,j}\right) \overline{U}_i^n + \Delta t \sum_{j \in N_i^{e,-}} |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} U_j^n(\mathbf{x}_{i,j}) \\ + \Delta t \sum_{j \in N_i^{e,+}} |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} r_{i,j} U_i^n(\mathbf{x}_{i,j'}). \quad (18)\end{aligned}$$

We will now prove Properties 1 and 2.

Sum. The sum constraint is automatically satisfied because

$$\begin{aligned} d_i + \sum d_j &= \left(1 - \Delta t \sum_{j \in N_i^{e,+}} r_{i,j} |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} - \Delta t \sum_{j \in N_i^{e,-}} |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|}\right) \\ &\quad + \Delta t \sum_{j \in N_i^{e,+}} r_{i,j} |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} + \Delta t \sum_{j \in N_i^{e,-}} |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} \\ &= 1. \end{aligned}$$

Non-negativity. First, note that the coefficients in (18)

$$\Delta t |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} \quad \text{and} \quad \Delta t |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} r_{i,j}$$

corresponding to d_j in (13) are always non-negative. Next, we will choose a local stable time step Δt_i such that d_i is non-negative as well, i.e.

$$d_i = 1 - \Delta t_i \sum_{j \in N_i^{e,-}} |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} - \Delta t_i \sum_{j \in N_i^{e,+}} |\mathbf{a}\cdot\mathbf{n}_{i,j}| \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} r_{i,j} \geq 0. \quad (19)$$

Observing that $|\mathbf{a}\cdot\mathbf{n}_{i,j}| \leq ||\mathbf{a}||$, $r_{i,j} \leq 2$, extending the sums from $N_i^{e,\pm}$ to N_i^e, and rearranging the terms,

we obtain the following upper bound on the sum terms in (19)

$$\Delta t_i \sum_{j\in N_i^{e,-}} |\mathbf{a}\cdot\mathbf{n}_{i,j}|\frac{|\partial\Omega_{i,j}|}{|\Omega_i|} + \Delta t_i \sum_{j\in N_i^{e,+}} |\mathbf{a}\cdot\mathbf{n}_{i,j}|\frac{|\partial\Omega_{i,j}|}{|\Omega_i|} r_{i,j} \leq 3\Delta t_i||\mathbf{a}||\frac{\sum_{j\in N_i^e, j\neq i}|\partial\Omega_{i,j}|}{|\Omega_i|}.$$

Coefficient d_i will be non-negative if

$$3\Delta t_i||\mathbf{a}||\frac{\sum_{j\in N_i^e, j\neq i}|\partial\Omega_{i,j}|}{|\Omega_i|} \leq 1.$$

Solving for Δt_i yields the sufficient condition for the non-negativity property (15)

$$\Delta t_i \leq \frac{1}{6}\frac{h_{c,i}}{||\mathbf{a}||},$$

where

$$h_{c,i} = 2\frac{|\Omega_i|}{|\partial\Omega_i|},$$

$|\partial\Omega_i|$ is the perimeter of Ω_i, and $h_{c,i}$ is the radius of the circle inscribed in Ω_i. Then the non-negativity constraint on the entire mesh is

$$\Delta t \leq \frac{1}{6}\min_i \frac{h_{c,i}}{||\mathbf{a}||}. \tag{20}$$

Finally, property 3 is guaranteed by limiter (6). Thus (12) is true, and the linear scheme (10) is L_∞ non-increasing with time in the means.

Nonlinear problems. We consider the scheme (9), and use the notation $F_{i,j}(U_1, U_2) = \mathbf{F}^*(U_1, U_2)\cdot\mathbf{n}_{i,j}$. Similar to the linear case, we use the divergence theorem to obtain

$$\sum_{j\in N_i^e, j\neq i} |\partial\Omega_{i,j}| F_{i,j}(\overline{U}_i^n, \overline{U}_i^n) = 0. \tag{21}$$

Using (21) in (9), we obtain

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

Adding and subtracting $F_{i,j}(\overline{U}_i^n, U_j^n(\mathbf{x}_{i,j,q}))$ in the inner sum, we have

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

Using the mean value theorem, we obtain

$$\overline{U}_i^{n+1} = \overline{U}_i^n - \frac{\Delta t}{|\Omega_i|} \sum_{j \in N_i^e, j \neq i} \frac{1}{2} |\partial\Omega_{i,j}| \sum_{q=1,2} \frac{\partial F_{i,j}}{\partial U_1}(\xi_1, U_j^n(\mathbf{x}_{i,j,q}))(U_i^n(\mathbf{x}_{i,j,q}) - \overline{U}_i^n) + \frac{\partial F_{i,j}}{\partial U_2}(\overline{U}_i^n, \xi_2)(U_j^n(\mathbf{x}_{i,j,q}) - \overline{U}_i^n)$$

with $\frac{\partial F_{i,j}}{\partial U_1}$ and $\frac{\partial F_{i,j}}{\partial U_2}$ as the partial derivatives with respect to the first and second arguments of $F_{i,j}$, respectively, ξ_1 between $\overline{U}_i^n$ and $U_i^n(\mathbf{x}_{i,j,q})$ and ξ_2 between $\overline{U}_i^n$ and $U_j^n(\mathbf{x}_{i,j,q})$. Introducing $v_{i,j,q}^1 = \Delta t_i \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} \frac{\partial F_{i,j}}{\partial U_1}(\xi_1, U_j^n(\mathbf{x}_{i,j,q}$ and $v_{i,j,q}^2 = \Delta t_i \frac{|\partial\Omega_{i,j}|}{|\Omega_i|} \frac{\partial F_{i,j}}{\partial U_2}(\overline{U}_i^n, \xi_2)$, we have

$$\overline{U}_i^{n+1} = \overline{U}_i^n + \sum_{j \in N_i^e, j \neq i} \frac{1}{2} \sum_{q=1,2} v_{i,j,q}^1(\overline{U}_i^n - U_i^n(\mathbf{x}_{i,j,q})) - v_{i,j,q}^2(U_j^n(\mathbf{x}_{i,j,q}) - \overline{U}_i^n).$$

182 By the monotonicity property of the numerical flux

$$v_{i,j,q}^1 \geq 0 \text{ and } -v_{i,j,q}^2 \geq 0. \tag{22}$$

As in the case of a linear flux, we apply Proposition 1 to the $\overline{U}_i^n - U_i^n(\mathbf{x}_{i,j,q})$ term, i.e.

$$\overline{U}_i^{n+1} = \overline{U}_i^n + \sum_{j \in N_i^e, j \neq i} \frac{1}{2} \sum_{q=1,2} v_{i,j,q}^1 r_{i,j,q}(U_i^n(\mathbf{x}_{i,j',q}) - \overline{U}_i^n) - v_{i,j,q}^2(U_j^n(\mathbf{x}_{i,j,q}) - \overline{U}_i^n),$$

where $r_{i,j,q}$ is the scaling coefficient r on edge $\partial\Omega_{i,j}$ at the qth quadrature point. Grouping terms yields

$$\overline{U}_i^{n+1} = \left(1 - \sum_{j \in N_i^e, j \neq i} \sum_{q=1,2} \frac{1}{2}(v_{i,j,q}^1 r_{i,j,q} - v_{i,j,q}^2)\right) \overline{U}_i^n + \sum_{j \in N_i^e, j \neq i} \frac{1}{2} \sum_{q=1,2} \left[v_{i,j,q}^1 r_{i,j,q} U_i^n(\mathbf{x}_{i,j',q}) - v_{i,j,q}^2 U_j^n(\mathbf{x}_{i,j,q})\right]. \tag{23}$$

183 This is of the form (13). We will now prove properties 1 and 2.

Sum. The sum constraint is automatically satisfied because

$$\begin{aligned} d_i + \sum d_j &= 1 - \sum_{j \in N_i^e, j \neq i} \sum_{q=1,2} \frac{1}{2}(v_{i,j,q}^1 r_{i,j,q} - v_{i,j,q}^2) \\ &\quad + \sum_{j \in N_i^e, j \neq i} \{\frac{1}{2} \sum_{q=1,2} v_{i,j,q}^1 r_{i,j,q} - \frac{1}{2} \sum_{q=1,2} v_{i,j,q}^2\} \\ &= 1. \end{aligned}$$

Non-negativity. First, note that the coefficients in (23) corresponding to the d_j coefficients in (13) are always non-negative by (22). Next, we will choose a Δt_i such that d_i is non-negative as well, i.e.

$$d_i = 1 - \sum_{j \in N_i^e, j \neq i} \sum_{q=1,2} \frac{1}{2}(v_{i,j,q}^1 r_{i,j,q} - v_{i,j,q}^2) \geq 0.$$

Due to the differentiability of the numerical flux, there exists a λ_i such that $\frac{\partial F_{i,j}}{\partial U_1}(\xi_1, U_j^n(\mathbf{x}_{i,j,q})) \leq \lambda_i$ and $-\frac{\partial F_{i,j}}{\partial U_2}(\overline{U}_i^n, \xi_2) \leq \lambda_i$ hold. Similar to the linear case, a sufficient condition on the local time step Δt_i is

$$6\Delta t_i \frac{\lambda_i}{h_{c,i}} \leq 1.$$

A time step suitable for all elements is determined by minimizing the ratio $\frac{h_{c,i}}{\lambda_i}$, i.e.

$$\Delta t \leq \frac{1}{6} \min_i \frac{h_{c,i}}{\lambda_i}.$$

Finally, Property 3 is enforced with limiter (6). Thus (12) is true and the nonlinear scheme (9) is L_∞ non-increasing. □

Remark.

Here we derive a less restrictive CFL condition for linear problems. Consider the coefficient

$$d_i = 1 - \Delta t \sum_{j \in N_i^{e,-}} |\mathbf{a} \cdot \mathbf{n}_{i,j}| \frac{|\partial \Omega_{i,j}|}{|\Omega_i|} - \Delta t \sum_{j \in N_i^{e,+}} |\mathbf{a} \cdot \mathbf{n}_{i,j}| \frac{|\partial \Omega_{i,j}|}{|\Omega_i|} r_{i,j}. \tag{24}$$

Because $0 \leq r_{i,j} \leq 2$, it follows that d_i is bounded below by

$$1 - \Delta t \sum_{j \in N_i^{e,-}} |\mathbf{a} \cdot \mathbf{n}_{i,j}| \frac{|\partial \Omega_{i,j}|}{|\Omega_i|} - 2\Delta t \sum_{j \in N_i^{e,+}} |\mathbf{a} \cdot \mathbf{n}_{i,j}| \frac{|\partial \Omega_{i,j}|}{|\Omega_i|} \leq d_i.$$

188 The non-negativity of d_i is guaranteed if

$$0 \leq 1 - \Delta t \sum_{j \in N_i^{e,-}} |\mathbf{a} \cdot \mathbf{n}_{i,j}| \frac{|\partial \Omega_{i,j}|}{|\Omega_i|} - 2\Delta t \sum_{j \in N_i^{e,+}} |\mathbf{a} \cdot \mathbf{n}_{i,j}| \frac{|\partial \Omega_{i,j}|}{|\Omega_i|} \leq d_i. \tag{25}$$

From (17), we have the identity

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

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

$$\sum_{j \in N_i^{e,-}} |\partial \Omega_{i,j}| |\mathbf{a} \cdot \mathbf{n}_{i,j}| = \sum_{j \in N_i^{e,+}} |\partial \Omega_{i,j}| |\mathbf{a} \cdot \mathbf{n}_{i,j}|. \tag{26}$$

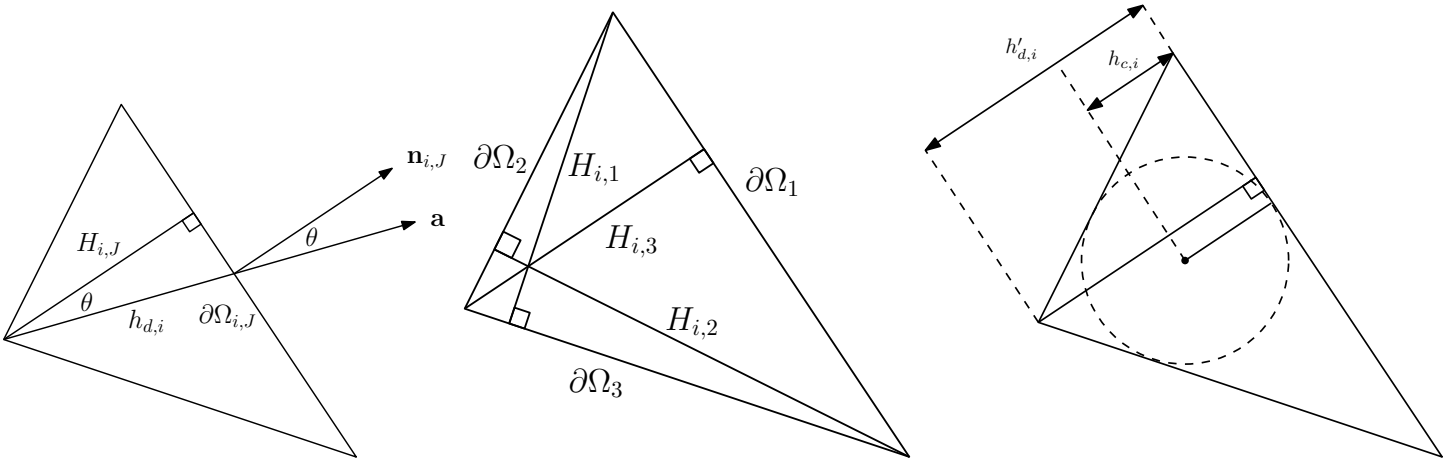
For linear problems, three situations are possible. There can be two inflow edges and one outflow edge, or one inflow edge and two outflow edges. In these two situations, there is a single inflow or a single outflow edge. We refer to that edge as $\partial \Omega_{i,J}$. Finally, when the direction of flow is parallel to an edge, i.e. is one inflow and one outflow edge. In this case, $\partial \Omega_{i,J}$ can refer to either the inflow or outflow edge. In terms of $\partial \Omega_{i,J}$, identity (26) now becomes

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

190 Using the above in (25), we obtain

$$0 \leq 1 - 3\Delta t |\mathbf{a} \cdot \mathbf{n}_{i,J}| \frac{|\partial \Omega_{i,J}|}{|\Omega_i|}. \tag{27}$$

191 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 $\partial \Omega_{i,J}$ as
192 shown in Figure 3a. Further, a simple geometric consideration reveals that $||\mathbf{a}|| H_{i,J} = h_{d,i} |\mathbf{a} \cdot \mathbf{n}_J|$, where
193 $h_{d,i}$ is the width of the cell in the direction of $\mathbf{a}$ as in Figure 3a. The non-negativity constraint on the



(a) Cell size in the direction of flow $h_{d,i}$. (b) Cell size for systems $h'_{d,i} = \min(H_{i,1}, H_{i,2}, H_{i,3})$. (c) Comparison of minimum cell width $h'_{d,i}$ and radius of the inscribed circle $h_{c,i}$.

Figure 3: Measures of cell size for time step restriction (28).

entire mesh is then given by

$$\Delta t \leq \frac{1}{6} \min_i \frac{h_{d,i}}{||\mathbf{a}||}. \tag{28}$$

By geometrical considerations, we note that this $h_{d,i}$ is larger than the radius of the inscribed circle $h_{c,i}$ (Figure 3c). Therefore, this CFL condition (28) is less restrictive.

For systems of equations, in general, there is not a single direction along which information is propagated. For simplicity, we propose to take the minimum possible cell width, i.e.,

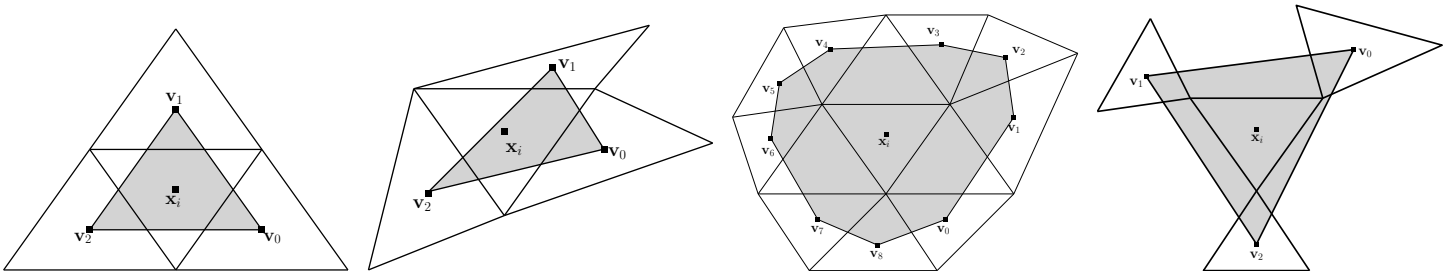
$$h'_{d,i} = \min(H_{i,1}, H_{i,2}, H_{i,3}), \tag{29}$$

where $H_{i,1}$, $H_{i,2}$, and $H_{i,3}$ are the cell widths perpendicular to the three edges of the element, $\partial\Omega_1$, $\partial\Omega_2$, and $\partial\Omega_3$, as shown in Figure 3b.

We have shown that at the next time step the solution means will satisfy a local maximum principle that depends on the chosen limiting neighborhood. That is, $\overline{U}_i^{n+1}$ will lie in the interval $[m', M']$, where m' and M' depend on the elements involved in limiting by (7). In the next section, we discuss how the choice of limiting points and neighborhoods affects the accuracy of the numerical solution.

6. Solution accuracy and admissibility region

In order to preserve second order accuracy, the limiting algorithm (6) must not modify linear data. We call the set from which one can choose limiting points such that this condition is not violated the *admissibility region*. First, we give a definition of this region, and then prove in Theorem 2 that points



(a) Edge-neighborhood, equilateral triangles.

(b) Edge-neighborhood, deformed triangles.

(c) Vertex-neighborhood.

(d) Reduced-neighborhood.

Figure 4: Admissibility regions for Ω_i and various limiting neighborhoods.

from this region satisfy the desired property.

Definition 1. *The limiter's admissibility region is defined as the convex hull of the centroids of the elements in N_i, where N_i is the neighborhood of Ω_i involved in the limiter* (6)*. Geometrically, this region is a convex polygon whose vertices are labeled $\mathbf{v}_k$ and ordered counterclockwise about their barycenter (Figure 4). Any point $\mathbf{x}$ in the region can be written as*

$$\mathbf{x} = \sum_k \gamma_k \mathbf{v}_k,$$

$$\sum_k \gamma_k = 1 \text{ and } \gamma_k \geq 0.$$

By definition, the points $\mathbf{x}$ in the convex hull satisfy the following conditions

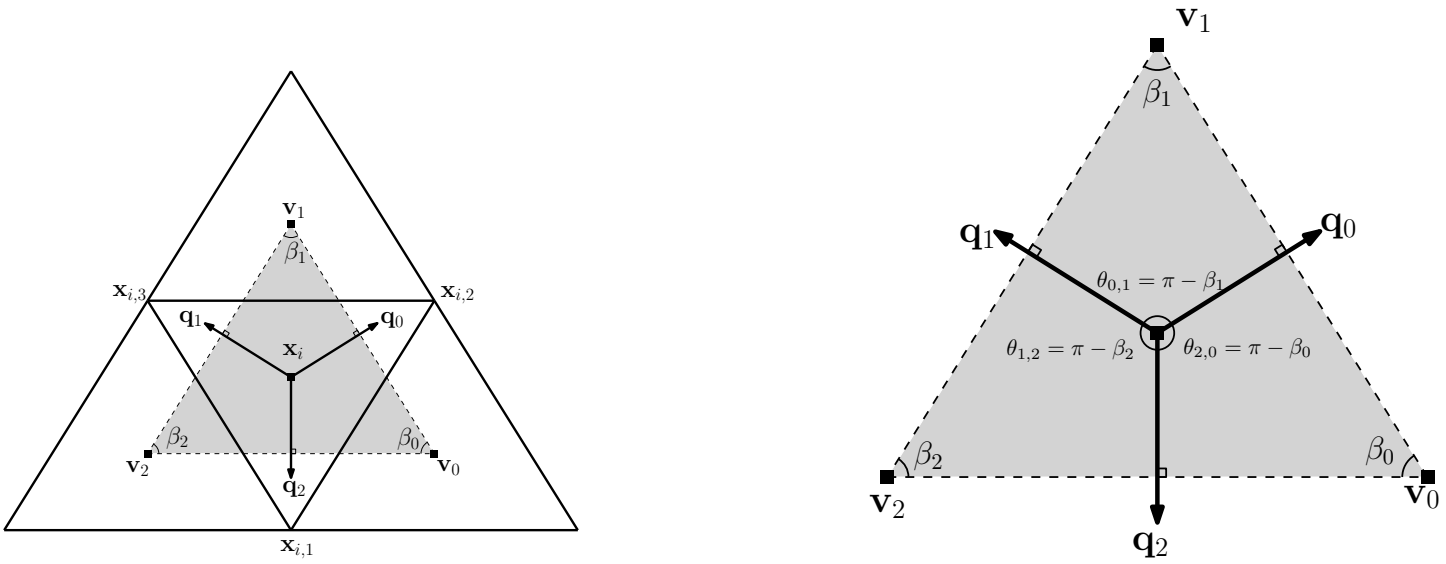
$$(\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{q}_k \leq (\mathbf{v}_k - \mathbf{x}_i) \cdot \mathbf{q}_k, \tag{30}$$

$$(\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{q}_k \leq (\mathbf{v}_{k+1} - \mathbf{x}_i) \cdot \mathbf{q}_k, \tag{31}$$

for all indices k, where $\mathbf{q}_k$ are outward pointing unit vectors such that $\mathbf{q}_k \cdot (\mathbf{v}_{k+1} - \mathbf{v}_k) = 0$, i.e. they are vectors perpendicular to the boundaries of the convex hull. Additionally, the pairs $\mathbf{q}_k$ and $\mathbf{q}_{k+1}$ are linearly independent.

We display examples of admissibility regions in Figure 4. These regions depend on the neighborhood involved in computing the local minimum and maximum $[m_i^n, M_i^n]$ in the limiting procedure in Section 3. For the edge neighborhood the region is simply the triangle formed by connecting the centroids of the elements that share an edge with Ω_i, as shown in Figures 4a and 4b. For the vertex neighborhood, the shape is more complex, as shown in Figure 4c.

The admissibility region of the vertex neighborhood usually contains all the limiting points. An excep-



(a) Edge neighborhood, with vectors $\mathbf{q}_k$.

(b) Admissibility region of Ω_i in Figure 5a.

Figure 5: Illustration for Theorem 2, where $\Omega_i = (\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \mathbf{x}_{i,3})$, and the admissibility region is shaded.

tion would be neighborhoods of elements that are located on the boundary of the computational domain. The number of elements in the vertex neighborhood is variable in unstructured meshes. This leads to memory inefficiencies in numerical codes as the stencil for the limiter varies from element to element. This motivates defining the reduced neighborhood. We iterate through all possible combinations of three elements in the vertex neighborhood and choose the first subset such that all limiting points are contained in its convex hull (Figure 4d).

Theorem 2. *(i) If the limiting points lie in the admissibility region then linear data will not be modified by the limiter* (6)*. (ii) If a limiting point lies outside the admissibility region, then there exists a gradient that will be modified by the limiter* (6)*.*

Proof. We first prove part (i) of the theorem. Let us consider an admissibility region, which is a polygon with vertices $\mathbf{v}_k$ (Figure 5a). We denote by β_k the angle formed by the edges of the polygon at vertex $\mathbf{v}_k$. Since the region is convex, we have that $0 < \beta_k < \pi$. We denote by $\theta_{k,k+1}$ the angle between $\mathbf{q}_k$ and $\mathbf{q}_{k+1}$. A simple geometric consideration reveals that $\theta_{k,k+1} = \pi - \beta_k$ (Figure 5b) and, consequently, $0 < \theta_{k,k+1} < \pi$.

We consider a vector $\mathbf{g} = \nabla U_i$. There exists an index K such that $\mathbf{g}$ lies between $\mathbf{q}_K$ and $\mathbf{q}_{K+1}$ (Figure 5b). Since $0 < \theta_{K,K+1} < \pi$, we can express $\mathbf{g} = c_1\mathbf{q}_K + c_2\mathbf{q}_{K+1}$ such that $c_1, c_2 \geq 0$. Assume the limiting point $\mathbf{x} \in \Omega_i$ is in the admissibility region. Therefore, it satisfies (30) and (31), with index $k = K + 1$ and

$k = K$, respectively, i.e.

$$(\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{q}_K \leq (\mathbf{v}_{K+1} - \mathbf{x}_i) \cdot \mathbf{q}_K,$$
$$(\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{q}_{K+1} \leq (\mathbf{v}_{K+1} - \mathbf{x}_i) \cdot \mathbf{q}_{K+1}.$$

Multiplying the first inequality by c_1, and the second by c_2, then summing, we have

$$(\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{g} \leq (\mathbf{v}_{K+1} - \mathbf{x}_i) \cdot \mathbf{g}.$$

Adding the cell average $\overline{U}_i$, we have by (5),

$$U_i(\mathbf{x}) \leq U_i(\mathbf{v}_{K+1}).$$

Therefore,

$$U_i(\mathbf{x}) \leq M_i,$$

where M_i is given in (7). The same reasoning can be applied to $\mathbf{g} = -\nabla U_i$, which gives

$$-U_i(\mathbf{x}) \leq -m_i.$$

Therefore,

$$m_i \leq U_i(\mathbf{x}),$$

where m_i is given by (7). Therefore $m_i \leq U_i(\mathbf{x}) \leq M_i$, $\mathbf{x} \in \Omega_i$, and by the limiter algorithm (6), the slope is not limited.

We now prove part (ii) of the theorem by constructing a solution that will be limited by algorithm (6) if a limiting point $\mathbf{x} \in \Omega_i$ lies outside of the admissibility region. In this case, at least one inequality (30) or (31), e.g. (30) with index K, will not hold:

$$(\mathbf{v}_K - \mathbf{x}_i) \cdot \mathbf{q}_K < (\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{q}_K.$$

However, each of the vertices $\mathbf{v}_k$ of the admissibility region belongs to the region itself, therefore by (30)

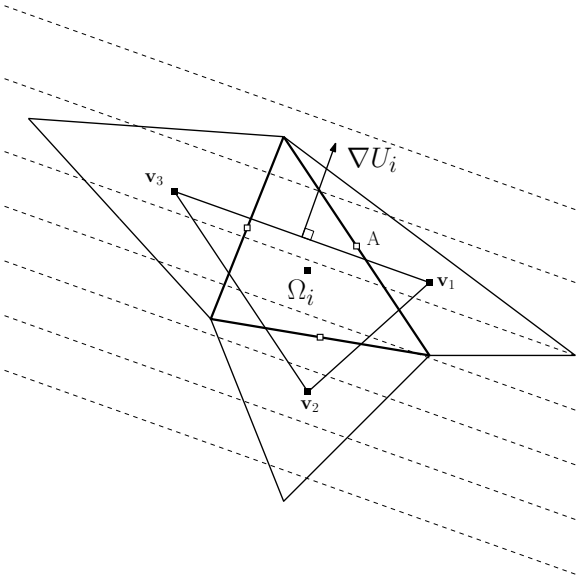


Figure 6: Limiting point A is outside the admissibility region of Ω_i, described by $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$.

and above, we have

$$(\mathbf{v}_k - \mathbf{x}_i) \cdot \mathbf{q}_K \leq (\mathbf{v}_K - \mathbf{x}_i) \cdot \mathbf{q}_K < (\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{q}_K. \tag{32}$$

Let us assume that the global solution is a plane, whose gradient is $\mathbf{q}_K$. We add to (32) the solution average $\overline{U}_i$ on Ω_i. Using (5), we obtain

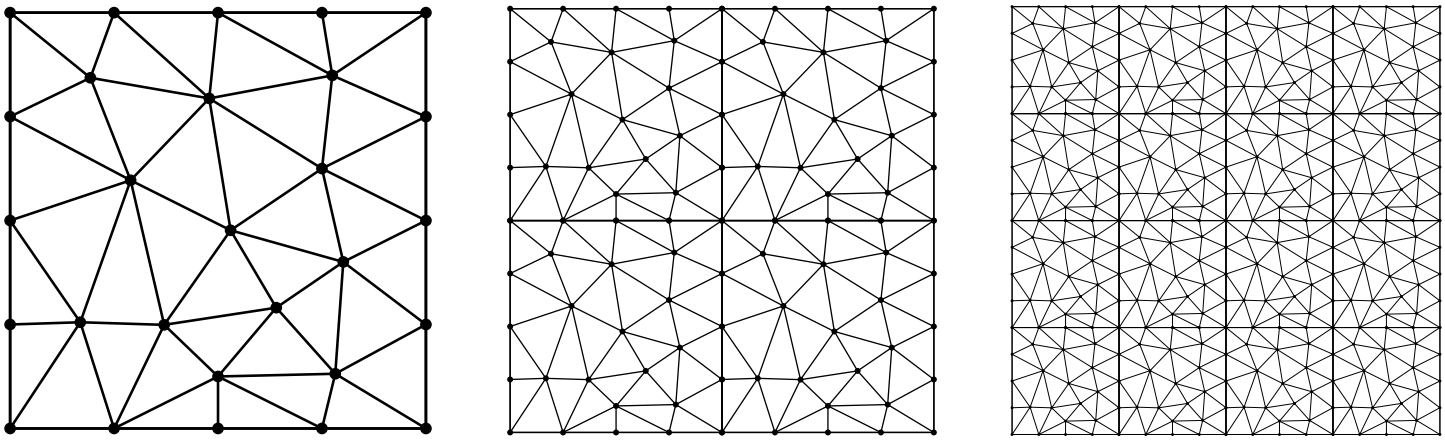
$$U_i(\mathbf{v}_k) < U_i(\mathbf{x}), \quad \forall k. \tag{33}$$

Additionally, because the vertices of the admissibility region correspond to neighboring element centroids, we have $U_i(\mathbf{v}_k) = \overline{U}_k$. Taking the maximum over k in (33) gives

$$M_i < U_i(\mathbf{x}).$$

Since U_i evaluated at $\mathbf{x}$ exceeds its allowed range, the slope of U_i will be limited by the limiting algorithm (6). □

Remark: Part (ii) of this theorem has a simple geometric interpretation that is illustrated in Figure 6. U_i is a linear function whose isolines are parallel lines. Since the isoline passing through the limiting point, A, lies higher than the centroids on the neighboring elements, the value of U_i at A exceeds the value at the neighboring centroids and the numerical solution on Ω_i will be limited.



(a) Ω^0. (b) Ω^1. (c) Ω^2.

Figure 7: Mesh sequence obtained through tiling.

7. Refinement studies

In the following discussion, we argue that limiting with the edge neighborhood should not be used with the discontinuous Galerkin method. With this limiter, we are not guaranteed that an element's limiting points will all lie in the admissibility region. This will lead to a reduced rate of convergence on smooth solutions. The observed rate of convergence under mesh refinement will depend on a number of factors: the quality of the initial mesh, the particular numerical solution, and the method of refinement. To illustrate this point, we construct two sequences of meshes and conduct numerical simulations that demonstrate different convergence behaviors. Here we analyze only the midpoint limiter, the two-point limiter will perform even worse.

All numerical simulations were done using the DG code described in [30] written for NVIDIA graphics processing units (GPUs), using the code optimizations described in [31].

7.1. Tiled refinement

We start with the initial mesh Ω_0 of a square domain Ω. Then, Ω^0 is scaled by a factor of $\frac{1}{2}$, and tiled over Ω to obtain the next mesh in the sequence Ω_1; that is, Ω^1 is composed of four scaled copies of Ω^0. We continue in a similar fashion, i.e. Ω_2 contains 16 scaled copies of Ω_0. We show a sample initial mesh, and two subsequent meshes obtained through tiling in Figure 7. The initial mesh is arbitrary with the only restriction that vertex placement on opposing boundaries is identical. This is needed in order to avoid nonconforming elements on the boundary of adjacent tiles. To simplify this discussion, we assume that elements on tile boundaries are not limited.

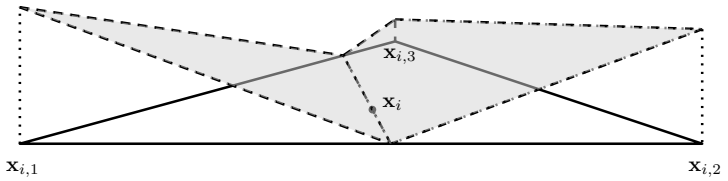


Figure 8: Shaded surfaces are described by the integrand $|a(x - x_i) + b(y - y_i)|$ in (34).

To demonstrate a loss of accuracy under limiting, we examine the limiting operation applied to a linear function $u(x, y)$. On Ω_i, this function can be written as

$$u_i(x, y) = a(x - x_i) + b(y - y_i) + \overline{u}_i,$$

where $\overline{u}_i$ is the average over Ω_i, x_i, y_i are the coordinates of the cell centroid. After applying the limiter to $u_i(x, y)$, we obtain the limited function

$$\tilde{U}_i(x, y) = \alpha_i a(x - x_i) + \alpha_i b(y - y_i) + \overline{u}_i,$$

where $\alpha_i \in [0, 1]$ is the limiting coefficient. This coefficient will not change upon translation or scaling of the mesh, provided the numerical solution is linear.

The L_1 norm of the error introduced due to limiting is

$$\begin{aligned} E_1(\Omega^0) &= \sum_i \int_{\Omega_i} |u_i(x, y) - \tilde{U}_i(x, y)| dxdy \\ &= \sum_i (1 - \alpha_i) \int_{\Omega_i} |a(x - x_i) + b(y - y_i)| dxdy. \end{aligned} \tag{34}$$

Each integral in the sum has a geometrical interpretation of the volume of two polyhedra since $a(x - x_i) + b(y - y_i)$ is zero along a line passing through the centroid of Ω_i. This is illustrated in Figure 8, where the shaded planes are the surfaces described by the integrand.

Shrinking the mesh by a factor of two, $\mathbf{x}' = \frac{1}{2}\mathbf{x}$, shrinks the volume of the polyhedra and, therefore, the error by a factor of eight. We have on the scaled mesh, Ω', the L_1 error

$$E_1(\Omega') = \frac{1}{8} E_1(\Omega^0).$$

Additionally, translating the mesh, $\mathbf{x}' = \mathbf{x} + \mathbf{d}$, does not change the error. On the translated mesh, Ω', the L_1 error is

$$\begin{aligned} E_1(\Omega') &= \sum_i (1-\alpha_i) \int_{\Omega_i} |a((x+d_x) - (x_i+d_x)) + b((y+d_y) - (y_i+d_y))| dxdy \\ &= \sum_i (1-\alpha_i) \int_{\Omega_i} |a(x-x_i) + b(y-y_i)| dxdy \\ &= E_1(\Omega^0). \end{aligned}$$

To summarize, scaling the Ω^0 by a factor of two reduces the error by a factor of eight and translation does not affect the error. Thus, the L_1 error on Ω^1, which consists of four scaled and translated copies of Ω_0 (Figure 7) is

$$E_1(\Omega^1) = \frac{4}{8} E_1(\Omega^0) = \frac{1}{2} E_1(\Omega^0).$$

This implies that the nth mesh in the tiled sequence has the error

$$E_1(\Omega^n) = \left(\frac{1}{2}\right)^n E_1(\Omega^0),$$

which indicates at most first order convergence in the general case.

7.2. Nested refinement

We now define a mesh sequence for which the same limiter, i.e. edge midpoints as limiting points coupled with the edge neighborhood, will yield a second order approximation of the initial data. We start by considering a mesh consisting of one element, Ω_i (Figure 9a). It is refined by splitting into four children, Ω_j, Ω_k, Ω_l, and Ω_m (Figure 9b). We can show that on the center child element of Ω_i, in this case Ω_j, linear data will not be limited. To show this, note that the limiting points of Ω_j, $\boldsymbol{\xi}$, are the midpoints of its edges:

$$\begin{aligned} \boldsymbol{\xi}_{j,1} &= \frac{1}{2}(\mathbf{x}_{j,3} + \mathbf{x}_{j,1}), \\ \boldsymbol{\xi}_{j,2} &= \frac{1}{2}(\mathbf{x}_{j,1} + \mathbf{x}_{j,2}), \\ \boldsymbol{\xi}_{j,3} &= \frac{1}{2}(\mathbf{x}_{j,2} + \mathbf{x}_{j,3}), \end{aligned} \tag{35}$$

where $\mathbf{x}_{j,1}$, $\mathbf{x}_{j,2}$, and $\mathbf{x}_{j,3}$ are the vertices of Ω_j. Further, the vertices of Ω_j are the midpoints of Ω_i's edges:

$$\begin{aligned}\mathbf{x}_{j,1} &= \frac{1}{2}(\mathbf{x}_{i,1} + \mathbf{x}_{i,2}),\\ \mathbf{x}_{j,2} &= \frac{1}{2}(\mathbf{x}_{i,2} + \mathbf{x}_{i,3}),\\ \mathbf{x}_{j,3} &= \frac{1}{2}(\mathbf{x}_{i,3} + \mathbf{x}_{i,1}),\end{aligned} \tag{36}$$

where $\mathbf{x}_{i,1}$, $\mathbf{x}_{i,2}$, and $\mathbf{x}_{i,3}$ are the vertices of Ω_i. Combining (35) and (36), we have

$$\begin{pmatrix}\boldsymbol{\xi}_{j,1}\\ \boldsymbol{\xi}_{j,2}\\ \boldsymbol{\xi}_{j,3}\end{pmatrix} = \begin{pmatrix}\frac{1}{2} & \frac{1}{4} & \frac{1}{4}\\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4}\\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2}\end{pmatrix}\begin{pmatrix}\mathbf{x}_{i,1}\\ \mathbf{x}_{i,2}\\ \mathbf{x}_{i,3}\end{pmatrix}. \tag{37}$$

We also write the centroids $\mathbf{x}_m$, $\mathbf{x}_k$, and $\mathbf{x}_l$ in terms of $\mathbf{x}_{i,1}$, $\mathbf{x}_{i,2}$, and $\mathbf{x}_{i,3}$:

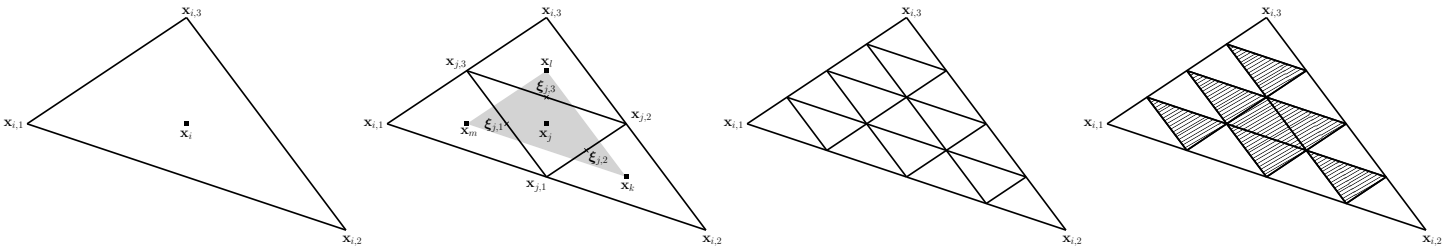
$$\begin{pmatrix}\mathbf{x}_m\\ \mathbf{x}_k\\ \mathbf{x}_l\end{pmatrix} = \begin{pmatrix}\frac{2}{3} & \frac{1}{6} & \frac{1}{6}\\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6}\\ \frac{1}{6} & \frac{1}{6} & \frac{2}{3}\end{pmatrix}\begin{pmatrix}\mathbf{x}_{i,1}\\ \mathbf{x}_{i,2}\\ \mathbf{x}_{i,3}\end{pmatrix}. \tag{38}$$

Combining (37) and (38), we obtain

$$\begin{pmatrix}\boldsymbol{\xi}_{j,1}\\ \boldsymbol{\xi}_{j,2}\\ \boldsymbol{\xi}_{j,3}\end{pmatrix} = \begin{pmatrix}\frac{2}{3} & \frac{1}{6} & \frac{1}{6}\\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6}\\ \frac{1}{6} & \frac{1}{6} & \frac{2}{3}\end{pmatrix}\begin{pmatrix}\mathbf{x}_m\\ \mathbf{x}_k\\ \mathbf{x}_l\end{pmatrix}.$$

Therefore, by Definition 1 the limiting points of Ω_j belong to its admissibility region. Thus linear data on Ω_j will not be limited by (6). An example is given in Figure 9b, where the limiting points lie inside the shaded admissibility region. In Figure 9c, the third mesh in the sequence, Ω^2, is shown. A simple geometric consideration reveals that elements that do not share an edge with the boundary of the original element will not be limited. This is because they are the center element of Ω^1, scaled by a factor of $\frac{1}{2}$, translated and rotated. Therefore, elements on which linear data is limited can only appear on the boundaries of the initial mesh elements in Ω^0 (Figure 9d). In the nth mesh of this sequence, there are $3(2^n - 1)$, $n > 0$, elements on the boundary.

We now construct a nested refinement sequence starting with an arbitrary initial triangulation, Ω^0,



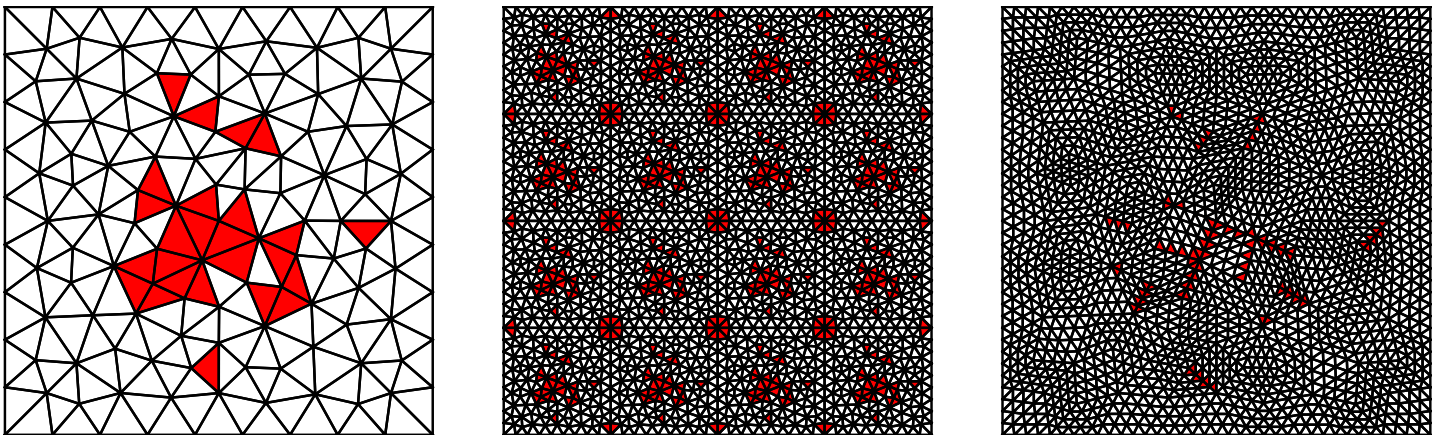
(a) Ω^0, one element Ω_i.

(b) Ω^1, four child elements Ω_j, Ω_k, Ω_l, and Ω_m. Grey zone is the admissibility region for Ω_j.

(c) Ω^2, sixteen child elements.

(d) Linear data on hatched elements in Ω^2 will not be limited by (6).

Figure 9: Mesh sequence obtained through nested refinement, with Ω^0 in (a) as the starting mesh.



(a) Initial mesh.

(b) Third mesh in tiling sequence.

(c) Third mesh in nested refinement sequence.

Figure 10: Sample meshes from tiling and nested refinement sequences. The elements on which linear data is limited are shaded.

with N_0 triangles, of a square domain Ω, e.g. the initial mesh in Section 7.1. To obtain the first mesh in the sequence, Ω^1, we refine each cell as described above. Completing this procedure n times gives the nth mesh in the sequence, Ω^n, which has $4^n N_0$ elements. The number of elements that can possibly be limited in Ω^n is $N_0 \cdot 3(2^n - 1)$, for $n > 0$. The upper bound on the number of elements on which the function will be approximated to first order grows linearly with h decreasing while the number of elements in the mesh grows quadratically, where h is a measure of cell size. This yields an effective second order rate of convergence in an integral norm of an approximation of the initial data, because the limited elements form a smaller and smaller proportion of the total number of elements in the mesh. This is easy to see by noting that the error on limited elements is $\mathcal{O}(h)$, the area of an element is $\mathcal{O}(h^2)$, and the number of limited elements is $\mathcal{O}(h^{-1})$. The product of these estimates gives $\mathcal{O}(h^2)$.

8. Numerical experiments

Unless otherwise specified, the problem is solved on the domain $[-1, 1]^2$, using the upwind or Lax-Friedrichs numerical flux on linear and nonlinear problems, respectively. We use the Heun's method to integrate in time unless otherwise stated and specify the particular time step restriction used in each example.

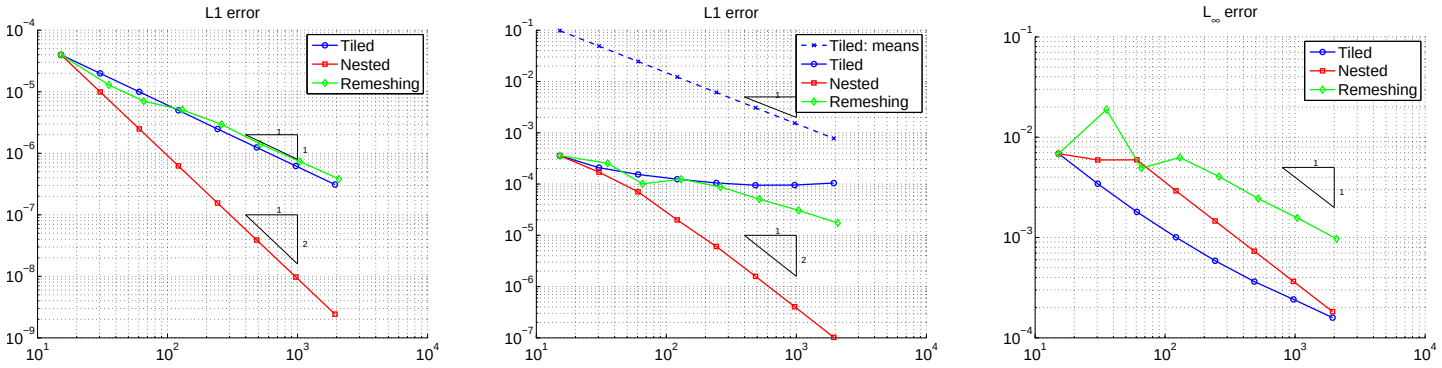
8.1. Accuracy verification - linear exact solution

We solve (1) with the flux $\mathbf{F}(u) = [u, u]$, on the tiled and nested mesh sequences (Figure 10) described in Section 7 with the initial and boundary condition chosen such that the exact solution is

$$u(x, y, t) = t - \frac{1}{2}x - \frac{1}{2}y. \tag{39}$$

We use the limiter based on the edge midpoints coupled with the edge neighborhood. First we project and limit the initial condition on the mesh sequences and report the error resulting from the application of the limiter in Figure 11a. Since there is no error due to projection of the initial condition into the finite element space, the observed error is entirely due to one application of the limiter. We observe the first and second order rate of convergence as discussed in Sections 7.1 and 7.2 for the tiled and nested mesh sequences, respectively. We also construct another sequence by remeshing the domain, whereby each mesh in the sequence has a comparable number of elements to nested and tiled refinement. The L_1 error on the remeshed sequence behaves similarly to tiled refinement, i.e. with first order accuracy.

Next, we examine the global error due to the cumulative effect of the interaction between the limiter and the DG method at the final time $T = 1$. In these experiments, we use the time step restriction based on the radius of the inscribed circle $h_{c,i}$ in (20) (Figures 11b and 11c). The limiter severely affects the solutions on the tiled sequence. The maximum error initially decreases with a rate of one, and then the rate tapers off to approximately 0.6. However, in the L_1 norm, convergence appears to stall. The error even increases at the last two solutions in the sequence. Note that the solution means are still converging with first order accuracy (Figure 11b); in the plot, we refer to this error with 'Tiled: means'. For nested refinement, we observe quadratic convergence in the L_1 norm and linear convergence in the L_∞ norm. The reason for the first order convergence in the L_∞ norm is that some elements are limited. On those elements, the accuracy is only first order. However, we observe second order convergence in the L_1 norm because the number of limited elements is small relative to the total number of elements in the mesh (for



(a) L_1 error after application of the limiter on the initial projection.

(b) L_1 error at the final time $T = 1$.

(c) L_∞ error at the final time $T = 1$.

Figure 11: Convergence history of initial condition (39) after projection and limiting, and final solution at $T = 1$ of (1) in Section 8.1. We plot the error versus (number of elements)$^{\frac{1}{2}}$.

example Figure 10c). Even though the analysis in Section 7.2 is only valid for the first application of the limiter, its conclusion seems to hold for multiple applications during a simulation.

8.2. Accuracy verification - nonlinear exact solution

We consider (1) with the flux $\mathbf{F}(u) = [u, u]$, and the following initial condition

$$u(x, y, 0) = \sin(\pi x) \sin(\pi y), \tag{40}$$

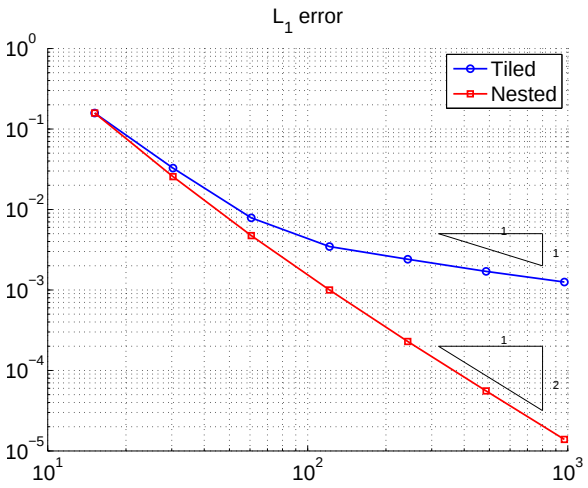
along with periodic boundary conditions in the x and y directions. We use a time step restriction based on the radius of the inscribed circle, $h_{c,i}$, in (20). We report the global error in the numerical solution at $T = 1$ on the nested and tiled mesh sequences (Figure 12). We note that convergence behavior is similar to that in Section 8.1 (Figure 11).

8.3. Validation of CFL number for linear fluxes

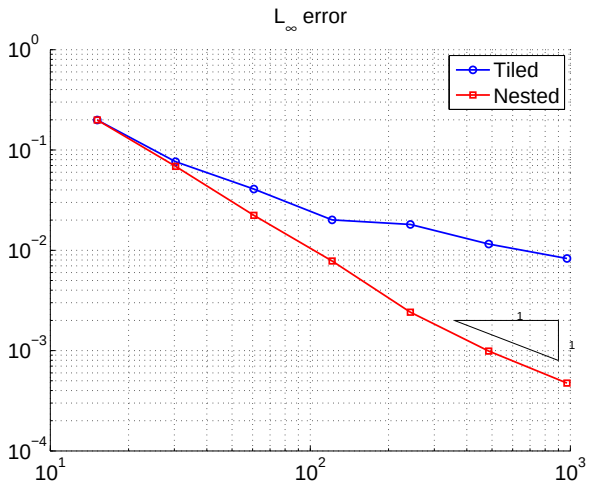
These experiments verify that the less restrictive CFL condition (28) based on cell width in the direction of flow $h_{d,i}$ is tight. We solve (1) with flux $\mathbf{F}(u) = [u, u]$ and a square pulse as the initial condition

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

We limit at the edge midpoints and use the vertex neighborhood. The domain is divided into a 40 by 40 grid of squares. Then the squares are split into triangles by connecting the upper left and lower right corners



(a) L_1 error at the final time $T = 1$.



(b) L_∞ error at the final time $T = 1$.

Figure 12: Convergence history of the final solution of (1) in Section 8.2. We plot the error versus (number of elements)$^{\frac{1}{2}}$.

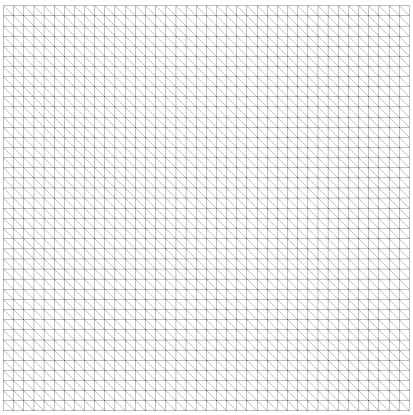


Figure 13: Structured mesh on which the linear CFL (28) based on $h_{d,i}$ is verified to be tight.

of each square (Figure 13). The width of the cells in the direction of the flow is $h_{d,i} = \frac{1}{40}\sqrt{2} \approx 3.535 \cdot 10^{-2}$ and the wave speed is $||\mathbf{a}|| = \sqrt{2}$. We solve the problem until the final time $T = 0.1$ with forward Euler and RK2 time stepping. The smallest and largest of the cell-wise solution averages at the final time are reported in Table 1 for various values of the CFL number.

We observe that the time step restriction (28) is valid and tight for the forward Euler method. For RK2, we notice in Table 1b that the CFL number can be increased without the solution violating the local maximum principle. A possible reason is the larger absolute stability region of the RK2 family of time integrators.

8.4. CFL experiments for a nonlinear flux

This experiment demonstrates that both measures of cell size, radius of the inscribed circle $h_{c,i}$ in (20) and minimum cell height $h'_{d,i}$ in (29), yield solutions that appear to satisfy the maximum principle for

1/CFL	Minimum	Maximum
3	-1.84e-01	1.23
4	-3.11-03	1.00085
5	-1.15e-05	1
5.95	-2.69e-07	1
6	**-6.56e-18**	**1**

(a) Forward Euler.

1/CFL	Minimum	Maximum
3	-9.50e-18	1.000336
4	-6.15e-18	1
5	-3.93e-18	1
5.95	-3.72e-18	1
6	**-3.62e-18**	**1**

(b) RK2-SSP.

Table 1: Minimum and maximum cell average for Example 8.3 using time step restriction (28) based on the width of the cell in the direction of flow, $h_{d,i}$, for various CFL numbers.

nonlinear problems. We consider problem (1) with flux $\mathbf{F}(u) = \left[\frac{1}{2}u^2, \frac{1}{2}u^2\right]$ (the two-dimensional Burgers' equation). The initial condition is a square pulse of side length $\frac{1}{2}$, centered at the origin and rotated by $\frac{\pi}{4}$ (Figure 14a). The exact solution at $T = \frac{\sqrt{2}}{2}$ is given by

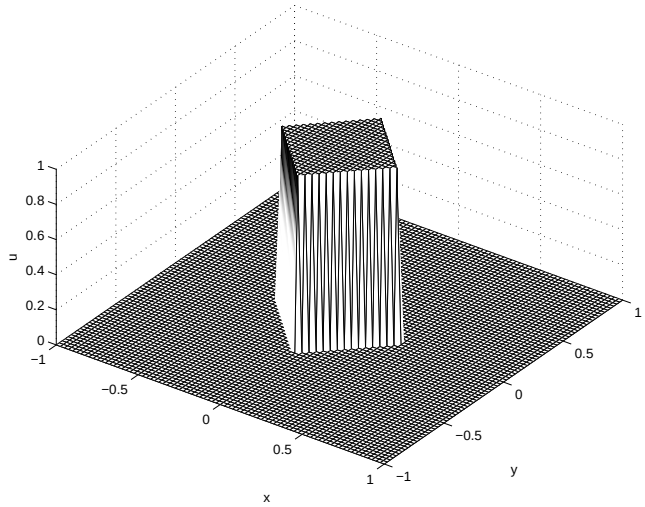
$$u\left(x', y', \frac{\sqrt{2}}{2}\right) = \begin{cases} x' + \frac{1}{4}, \text{ if } -\frac{1}{4} \leq x' \leq \frac{3}{4} \text{ and } -\frac{1}{4} \leq y' \leq \frac{1}{4}, \\ 0, \text{ otherwise,} \end{cases}$$

where $x' = \frac{\sqrt{2}}{2}x + \frac{\sqrt{2}}{2}y$ and $y' = -\frac{\sqrt{2}}{2}x + \frac{\sqrt{2}}{2}y$ (Figure 14b). We use the vertex neighborhood and the nodes of the two-point Gauss-Legendre quadrature rule as limiting points. The first mesh is generated by dividing the domain into a 10 by 10 grid of squares. Then the square elements are split into triangles by connecting the upper left and lower right corners of each square. Subsequent meshes we test on are obtained through nested refinement. We report the minimum and maximum cell means for both measures of cell size in Table 2. It appears that both time step restrictions result in solutions that are L_∞ non-increasing. However, the minimum cell height is substantially larger than the inscribed radius, which reduces the number of time steps by more than half.

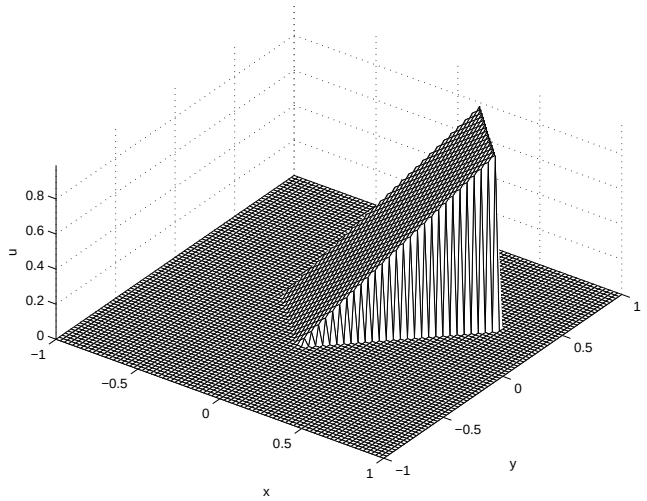
We also solve this problem using the vertex neighborhood and one-point limiting. According to the analysis in Theorem 1, the solution is not guaranteed to stay within the local bounds. We find this to be the case, however the growth in the means is on the order of approximately 10^{-6}. This indicates that while the proof is correct, practically the violation of the local maximum principle is small.

8.5. Rotating objects

This example demonstrates the comparative performance of the proposed limiters, i.e., one- or two-point limiting points with edge, vertex or reduced neighborhoods. We solve (1) with the flux $\mathbf{F}(u) = [-2\pi yu, 2\pi xu]$ on the square domain $[-1, 1]^2$. The exact solution is a rotation of the initial data about



(a) Exact initial condition.



(b) Exact solution at $T = \frac{\sqrt{2}}{2}$.

Figure 14: Exact solution at initial and final time for Example 8.4.

Number of elements	Time steps	Minimum	Maximum
200	85	5.22e-11	0.693
800	199	2.59e-18	0.873
3200	406	9.1e-36	0.934
12800	817	1.05e-69	0.962
51200	1637	4.14e-137	0.980

(a) Radius of the inscribed circle (20).

Number of elements	Time steps	Minimum	Maximum
200	35	5.53e-11	0.691
800	82	2.55e-18	0.871
3200	168	1.01e-35	0.932
12800	338	9.51e-70	0.962
51200	678	3.31e-137	0.980

(b) Minimum cell height (29).

Table 2: Verification of the time step restriction based on and radius of the inscribed circle $h_{c,i}$ in (20) and minimum cell height $h'_{d,i}$ in (29).

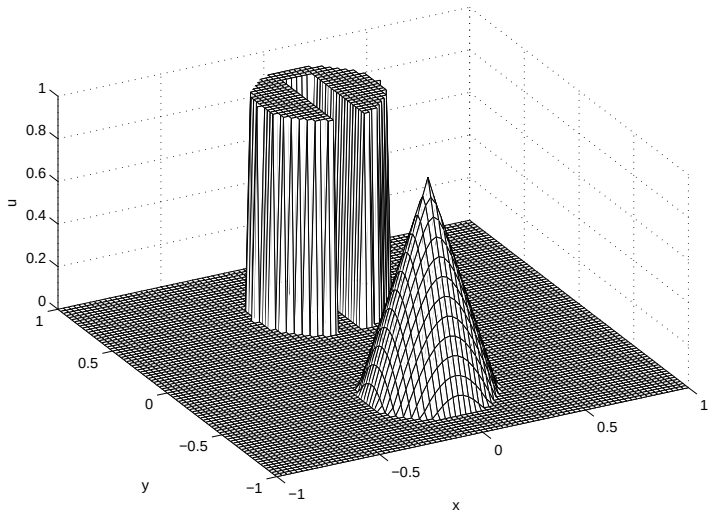


Figure 15: Initial condition for rotating objects test problem in Example 8.5.

the origin. The solution comprises a slotted cylinder and a cone (Figure 15). Each object is defined on a disc of radius $r_0 = 0.3$, the center of which is (x_0, y_0). The height of the objects are written in terms of $r(x,y) = \frac{1}{r_0}\sqrt{(x-x_0)^2 + (y-y_0)^2}$. Outside of the discs, the initial solution values are zero. The center of the slotted cylinder is $(x_0, y_0) = (0, 0.5)$ and its height is defined as

$$h(x,y) = \begin{cases} 1 \text{ if } |x - x_0| \geq 0.05 \text{ or } y \geq 0.7 \\ \\ 0 \text{ otherwise,} \end{cases} \qquad \text{for } r(x,y) \leq 1.$$

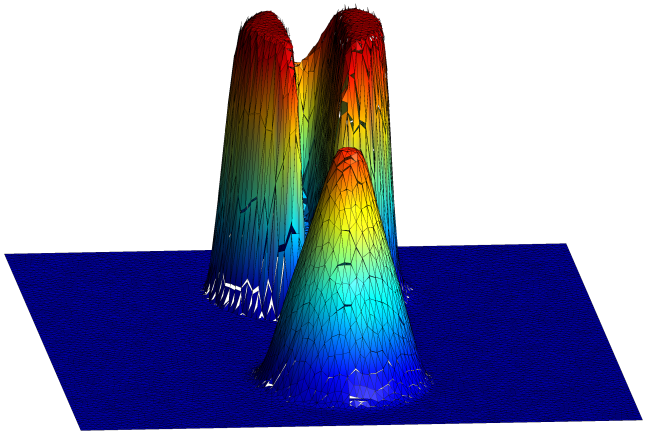
The center of the cone is $(x_0, y_0) = (0, -0.5)$ and its height is defined as

$$h(x,y) = 1 - r(x,y) \text{ for } r(x,y) \leq 1.$$

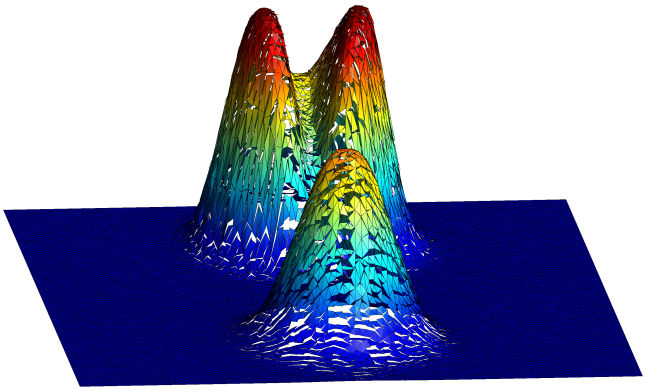
We use the time step restriction based on the minimum cell height $h'_{d,i}$ (29), and an unstructured mesh of 16,870 triangles. The surface integral in (3) is evaluated using the two-point quadrature rule. As a result, one-point limiting does not guarantee a solution that is L_∞ non-increasing. The solutions at $T = 1$ are plotted in Figures 16 - 19. The two-point, edge neighborhood limiter is clearly the most diffusive and the one-point, vertex neighborhood limiter is the least. The two-point, edge neighborhood limiter provides a noticeably worse solution than the other limiters, Figure 19. The one- and two-point vertex limiters yield the least diffusive results, and perform similarly.

In conclusion, we observe that neighborhoods with fewer elements are more diffusive, e.g. edge and reduced neighborhoods and larger neighborhoods are less diffusive, e.g. vertex neighborhood. Larger

neighborhoods result in larger intervals to which the numerical solution at the limiting points is constrained. Finally as expected, two-point limiting is more diffusive than one-point limiting.

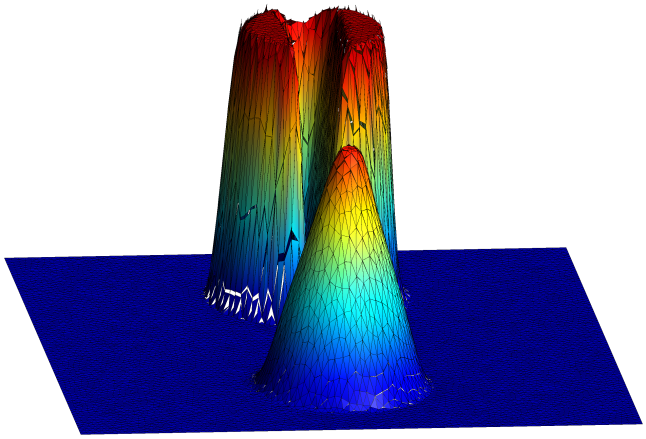


(a) One-point limiting, edge neighborhood.

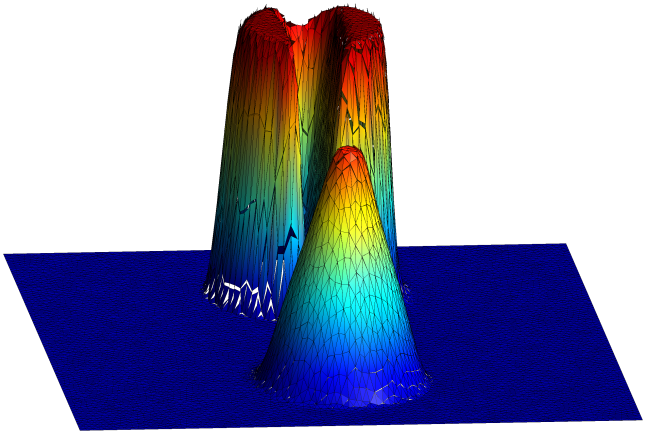


(b) Two-point limiting, edge neighborhood.

Figure 16: Raised solution in Example 8.5 for the edge neighborhoods.

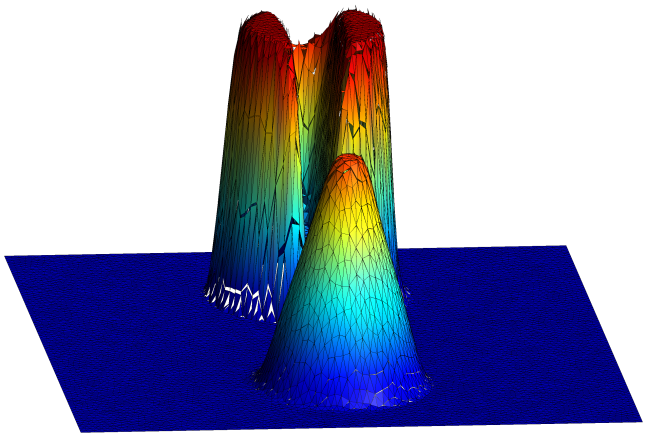


(a) One-point limiting, vertex neighborhood.

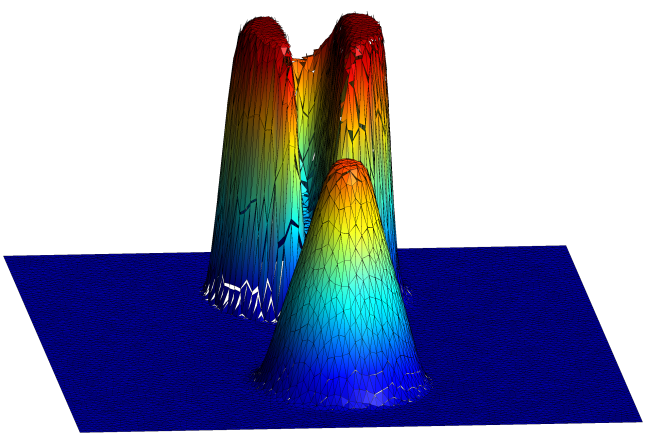


(b) Two-point limiting, vertex neighborhood.

Figure 17: Raised solution in Example 8.5 for the vertex neighborhoods.

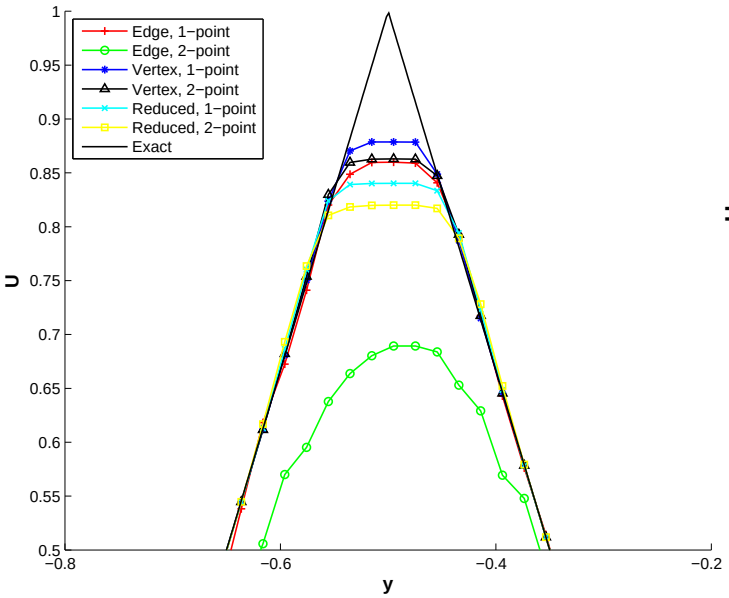


(a) One-point limiting, reduced neighborhood.

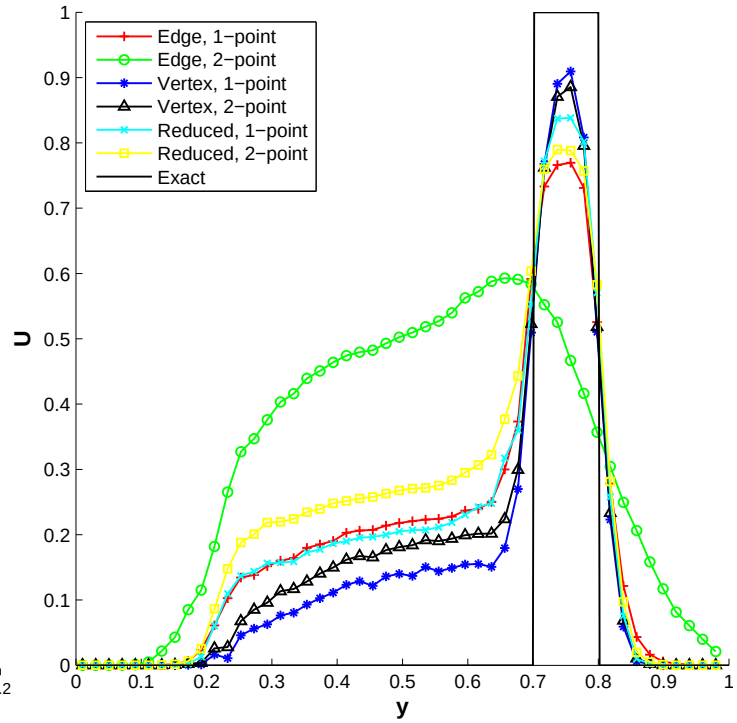


(b) Two-point limiting, reduced neighborhood.

Figure 18: Raised solution in Example 8.5 for the reduced neighborhoods.



(a) Profile at the cone.



(b) Profile at the slotted cylinder.

Figure 19: Cross sections of solution for Example 8.5 along $x = 0$ at $T = 1$.

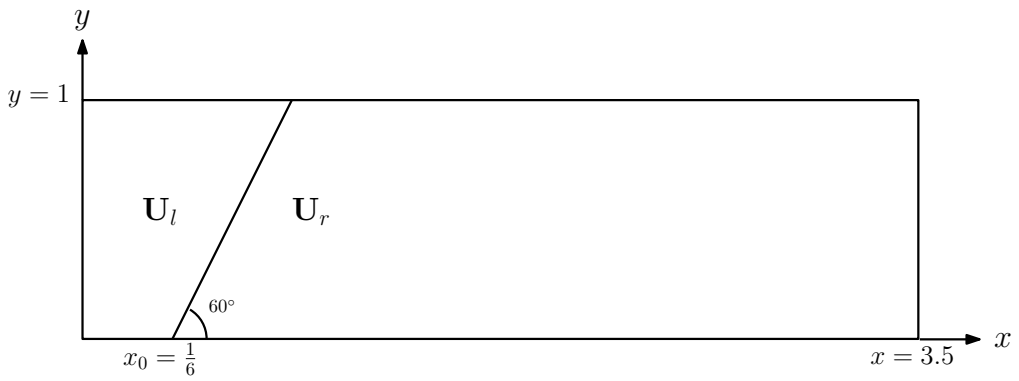


Figure 20: Set-up of the double Mach test problem in Example 8.6.

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

Table 3: Density, normal speed, and pressure to the left and right of the shock in Example 8.6.

8.6. Transient shock - double Mach reflection

Consider the two-dimensional Euler equations

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ E\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (E+p)u\end{pmatrix}+\frac{\partial}{\partial y}\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ (E+p)v\end{pmatrix}=0$$

with the equation of state

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

where $\gamma = 1.4$, the adiabatic constant for air. We solve the double Mach reflection problem using the set up described in [32, 30]. 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 set-up is shown in Figure 20, with the states to the left $\mathbf{U}_l$ and right $\mathbf{U}_r$ of the shock given in Table 3. The problem is solved on an unstructured mesh of 271,458 triangles until a final time of $T = 0.2$. We extend the limiter to systems of equations by limiting each component separately, i.e., we limit the conserved variables.

The contour plots for density using three different limiters are shown in Figures 21, 22, and 23. Although computationally simple, limiting using the edge neighborhood does not yield a solution of good quality. One-point limiting smears the slipline (contact discontinuity) emanating from the primary triple point in Figure 21a. Two-point limiting in Figure 21c is clearly too diffusive: the contact and reflected shock are

Neighborhood	limiting points	Run time (s)	Time steps	Time (ms) /step
vertex	1	51.4	8,624	5.9 (-)
reduced	1	18.7	8,409	2.2 (2.68x)
vertex	2	55.5	8,371	6.6 (-)
reduced	2	27.7	8,464	3.2 (2.06x)

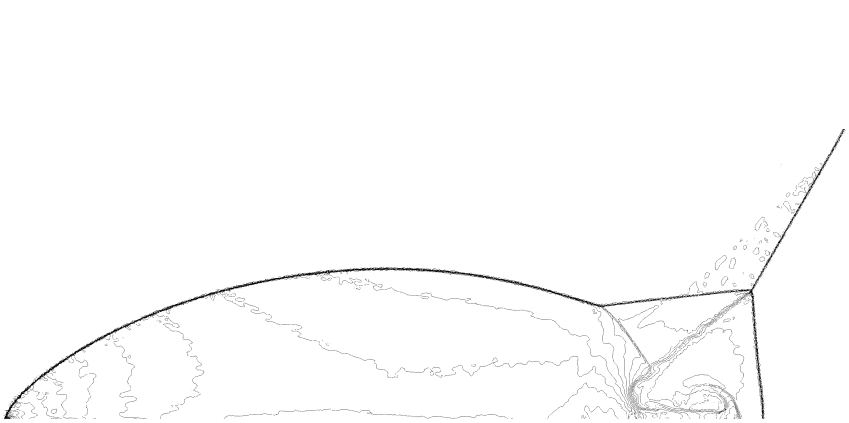
Table 4: Comparison of run time for limiters. The number in brackets is the speed up factor of the limiters using the reduced neighborhood relative to those using the vertex neighborhood, with the same number of limiting points.

smeared significantly and the rightward moving jet is not resolved at all.

One- and two-point limiting with the reduced neighborhood in Figures 22a and 22c performs just as poorly as one-point limiting with the edge neighborhood. Enlarging the limiting stencil to the vertex neighborhood significantly improves the solution in Figures in Figures 23a and 23c. The shocks and slipline are tighter and the jet is better resolved. We observe more vortices due to Rayleigh-Taylor instabilities. Both one- and two-point limiting appear to be numerically stable, though two-point limiting is more diffusive.

In Table 4, we report the time spent executing subroutines for limiters using the vertex and reduced neighborhoods in the DG-GPU code [30, 31] on an NVIDIA Titan X Pascal. The number of quadrature points does not seem to affect run time of the limiter subroutine as much as stencil size. We note that the run time is gravely affected when using a variable stencil size. For this test problem, the run time of the one-point, vertex neighborhood limiter subroutines took 5.9 ms, and the one-point, reduced neighborhood limiter subroutines was 2.2 ms; this is a 2.68x reduction in run time. On the mesh in this example, the vertex neighborhood size varies from 7 to 17, whereas the size of the reduced neighborhood is simply 3. The substantial increase in runtime of the limiter algorithm can be explained by the following. First, the amount of memory loads required to execute any vertex neighborhood based limiters is at least double that required for the reduced neighborhood. Further, due to size variability of the vertex neighborhood, thread divergence in the GPU code will limit the attainable parallelism at runtime. For a discussion of thread divergence in unstructured CFD codes on GPUs, see [31].

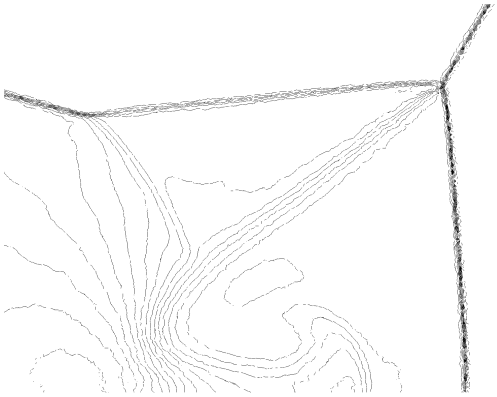
The conclusion to draw from this example is that the quality of the limited solution is a trade-off between computational work and numerical diffusion. For more computational work, one can reduce the amount of numerical diffusion introduced by the limiter by using the vertex neighborhood. The least computationally intensive limiter using the edge neighborhood can excessively smooth the solution. For this problem, we consider the two-point, vertex limiter as the best trade-off between solution quality and run time. Two-point limiting is preferred because the basis functions are already precomputed at these



(a) One-point limiting, edge neighborhood.

(b) One-point limiting, edge neighborhood, zoom.

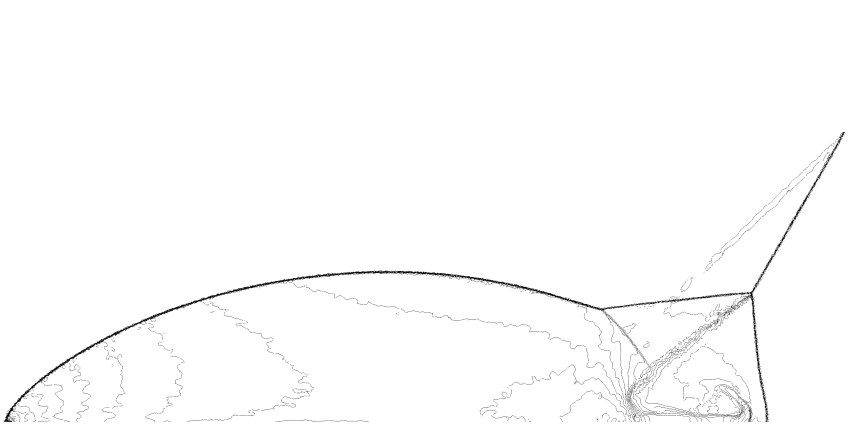
(c) Two-point limiting, edge neighborhood.



(d) Two-point limiting, edge neighborhood, zoom.

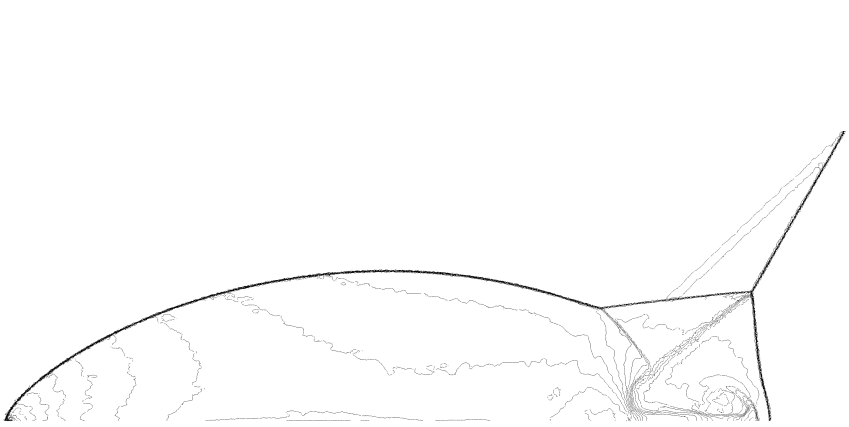
Figure 21: Double Mach reflection problem using the edge neighborhood.

quadrature points. Using two-point limiting is not substantially slower than using one-point limiting, though this may depend on the implementation.

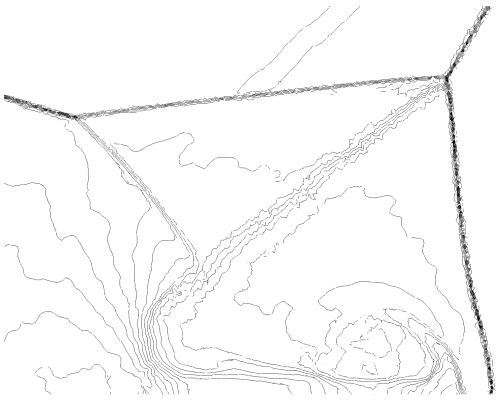


(a) One-point limiting, reduced neighborhood, zoom.

(b) One-point limiting, reduced neighborhood.

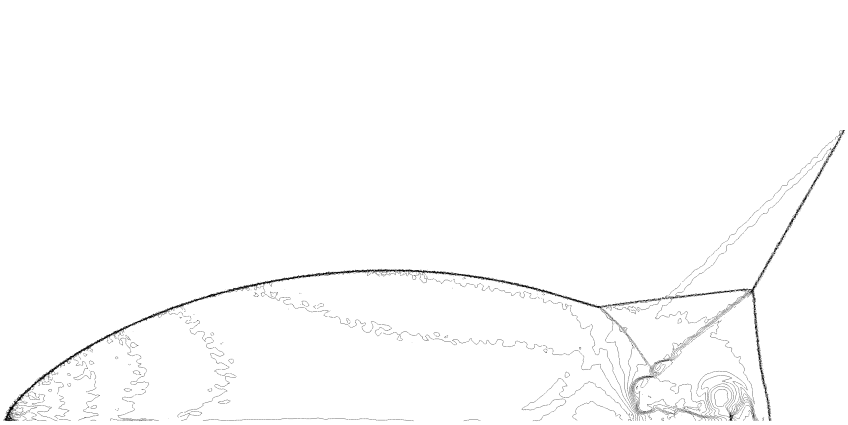


(c) Two-point limiting, reduced neighborhood.

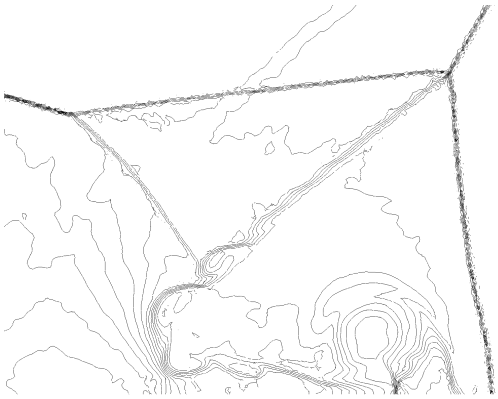


(d) Two-point limiting, reduced neighborhood, zoom.

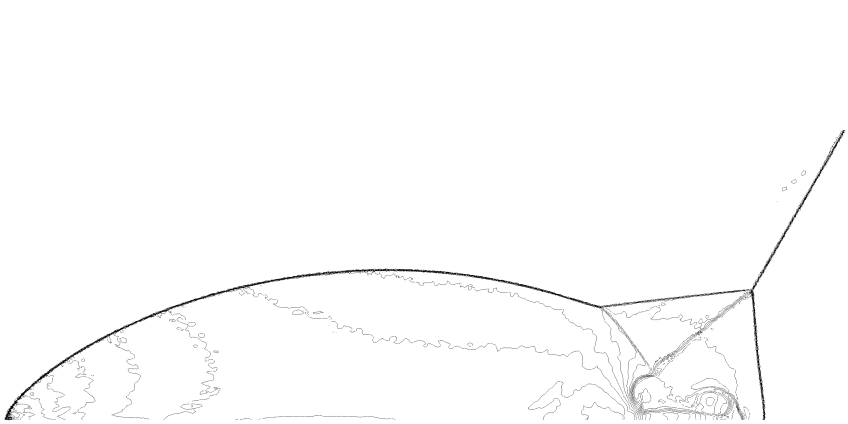
Figure 22: Double Mach reflection problem using the reduced neighborhood.



(a) One-point limiting, vertex neighborhood.



(b) One-point limiting, vertex neighborhood, zoom.



(c) Two-point limiting, vertex neighborhood.

(d) Two-point limiting, vertex neighborhood, zoom.

Figure 23: Double Mach reflection problem using the vertex neighborhood.

9. Conclusion

We have studied aspects of second order limiters on unstructured meshes of triangles. These limiters were first introduced and analyzed for finite volume methods but are often applied to DG methods. The important difference between finite volume and DG methods is the manner by which the surface integral is evaluated. FV methods use midpoint quadrature for nonlinear problems, while DG uses two-point quadrature, thus the FV analysis is not directly transferable to the DG method. Since the the surface integral advances the solution means in time, the values at the quadrature points need to be controlled for overshoots if we are to enforce the local maximum principle on the means. Numerical experiments indicate that limiting at the midpoint for nonlinear equations leads to the violation of the maximum principle by a small amount. For example, the solution in Section 8.4 grew only by 10^{-6}. While solutions with one-point limiting might look noisy, they do not have unbounded growth, even for long time integration. This indicates that strong stability may be more of theoretical interest rather than of practical purpose, unless an application cannot tolerate any deviation from the initial means.

We find that maintaining accuracy is more difficult than preventing uncontrolled growth of the solution. Our findings indicate that despite ease of implementation and convenience, the edge neighborhood, i.e. elements that share an edge with the element being limiting, should not be used for limiting. This is because the limiter is only first order accurate, even on meshes of reasonably good quality, e.g. Delaunay discretization of a square domain. While the edge neighborhood comes naturally as it is the stencil of the DG method, larger neighborhoods, e.g. the vertex neighborhood, should be employed for the solution to stay second order accurate. Unfortunately, this destroys the locality of the DG method, which is one of its advantageous aspects. Although a limiter that uses the DG stencil and preserves locality exists [18], it has its own drawbacks. It requires precomputing and storing geometrical coefficients, which is costly and it needs a user-defined parameter. A possible extension of the present work would be to analyze the admissible range of this parameter and its optimal value.

To summarize, the best limiter is the one based on the vertex neighborhood, though it is almost three times as expensive as one based on smaller neighborhoods. The edge neighborhood provides visibly worse solutions both at shocks and smooth regions. Although the maximum principle can only be enforced for scalar equations, the performance of the limiter on scalar equations is a predictor of performance on systems of equations. Numerical experiments with the Euler equations confirm this.

We show that the local maximum principle is satisfied under a suitable time step restriction. The

analysis is valid on one forward Euler time step and the bound is shown to be tight. This restriction involves a new measure of cell size, which is the width of the cell in the direction of flow. This is larger than the commonly used radius of the inscribed circle. A convex combination of forward Euler time steps can extend this stability restriction to high order time integration schemes, e.g. SSP-RK methods. Experimentally, we find that a larger time step can be taken for the SSP-RK2 method without violating the local maximum principle. Finding the analytical CFL number in this case is subject of future work.

10. 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. Proof of proposition 1

For simplicity of discussion, we translate the element Ω_i such that its centroid is located at the origin. The vectors pointing from the origin to the three vertices of the triangle are $\mathbf{v}_{i,1}$, $\mathbf{v}_{i,2}$, and $\mathbf{v}_{i,3}$. The vectors $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2$, and $\boldsymbol{\xi}_3$, pointing from the origin to the one- or two-point Gauss-Legendre quadrature points can be written as

$$\begin{aligned}\boldsymbol{\xi}_1 &= \epsilon\mathbf{v}_{i,2} + (1-\epsilon)\mathbf{v}_{i,3},\\ \boldsymbol{\xi}_2 &= \epsilon\mathbf{v}_{i,3} + (1-\epsilon)\mathbf{v}_{i,1},\\ \boldsymbol{\xi}_3 &= \epsilon\mathbf{v}_{i,1} + (1-\epsilon)\mathbf{v}_{i,2},\end{aligned}$$

where $\epsilon = \frac{1}{2}$ for the one-point rule, and $\epsilon = \frac{1}{2} \pm \frac{\sqrt{3}}{6}$ for the two-point rule. Let $\hat{\boldsymbol{\xi}}_{23,\perp}$ be the unit vector that is perpendicular to the vector $\boldsymbol{\xi}_2 - \boldsymbol{\xi}_3$, and that is pointing toward $\boldsymbol{\xi}_1$, i.e., $\boldsymbol{\xi}_1 \cdot \hat{\boldsymbol{\xi}}_{23,\perp} > 0$. Let $\hat{\boldsymbol{\xi}}_{1,\perp}$ be the unit vector that is perpendicular to $\boldsymbol{\xi}_1$, and that is pointing towards $\boldsymbol{\xi}_2$, i.e. $\hat{\boldsymbol{\xi}}_{1,\perp} \cdot \boldsymbol{\xi}_2 > 0$ (Figure A.24).

Because the centroid of Ω_i has been translated to the origin, we have

$$\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2 + \boldsymbol{\xi}_3 = \mathbf{0}, \tag{A.1}$$

which gives

$$\boldsymbol{\xi}_2 - \boldsymbol{\xi}_3 = -\boldsymbol{\xi}_1 - 2\boldsymbol{\xi}_3 = \boldsymbol{\xi}_1 + 2\boldsymbol{\xi}_2. \tag{A.2}$$

Multiplying (A.2) by $\boldsymbol{\xi}_{23,\perp}$, we have

$$0 = -\boldsymbol{\xi}_1 \cdot \hat{\boldsymbol{\xi}}_{23,\perp} - 2\boldsymbol{\xi}_3 \cdot \hat{\boldsymbol{\xi}}_{23,\perp} = \boldsymbol{\xi}_1 \cdot \hat{\boldsymbol{\xi}}_{23,\perp} + 2\boldsymbol{\xi}_2 \cdot \hat{\boldsymbol{\xi}}_{23,\perp}. \tag{A.3}$$

Rearranging terms gives the following relations

$$\boldsymbol{\xi}_1 \cdot \hat{\boldsymbol{\xi}}_{23,\perp} = -2\boldsymbol{\xi}_2 \cdot \hat{\boldsymbol{\xi}}_{23,\perp} = -2\boldsymbol{\xi}_3 \cdot \hat{\boldsymbol{\xi}}_{23,\perp}. \tag{A.4}$$

We now use this to prove proposition 1, which we restate here.

Proposition 1. *For a quadrature point* $\mathbf{x}$*, there exists a multiplier* $0 \leq r \leq 2$ *and another quadrature point* $\mathbf{x}'$ *on a different edge, such that*

$$U_i(\mathbf{x}) - \overline{U}_i = r(\overline{U}_i - U_i(\mathbf{x}')).$$

Proof. Without loss of generality, let us assume that $\mathbf{x}$ in the Proposition is $\boldsymbol{\xi}_1$ in Figure A.24. We will show that for a given gradient of numerical solution U_i, $\mathbf{g}$, there is a different quadrature point $\mathbf{x}'$ with $0 \leq r \leq 2$.

From the definition of $\hat{\boldsymbol{\xi}}_{23,\perp}$, $\boldsymbol{\xi}_1 \cdot \hat{\boldsymbol{\xi}}_{23,\perp} > 0$ (Figure A.24) and from (A.4) we have that $\boldsymbol{\xi}_2 \cdot \hat{\boldsymbol{\xi}}_{23,\perp} < 0$ and $\boldsymbol{\xi}_3 \cdot \hat{\boldsymbol{\xi}}_{23,\perp} < 0$. Additionally, taking the dot product of (A.1) and $\hat{\boldsymbol{\xi}}_{1,\perp}$, we have $\boldsymbol{\xi}_2 \cdot \hat{\boldsymbol{\xi}}_{1,\perp} = -\boldsymbol{\xi}_3 \cdot \hat{\boldsymbol{\xi}}_{1,\perp}$. Therefore $\boldsymbol{\xi}_2 \cdot \hat{\boldsymbol{\xi}}_{1,\perp}$ and $\boldsymbol{\xi}_3 \cdot \hat{\boldsymbol{\xi}}_{1,\perp}$ are of opposite sign. By definition of $\hat{\boldsymbol{\xi}}_{1,\perp}$, we have $\boldsymbol{\xi}_2 \cdot \hat{\boldsymbol{\xi}}_{1,\perp} > 0$, which implies that $\boldsymbol{\xi}_3 \cdot \hat{\boldsymbol{\xi}}_{1,\perp} < 0$.

Now, if the vector $\mathbf{g}$ lies between $\hat{\boldsymbol{\xi}}_{23,\perp}$ and $\hat{\boldsymbol{\xi}}_{1,\perp}$, i.e. in the region that we denote by (I) in Figure A.24, then we can write $\mathbf{g}$ as

$$\mathbf{g} = c_1\hat{\boldsymbol{\xi}}_{23,\perp} + c_2\hat{\boldsymbol{\xi}}_{1,\perp} \text{ with } c_1, c_2 > 0. \tag{A.5}$$

Recalling (A.4) and the definition of $\hat{\boldsymbol{\xi}}_{1,\perp}$, we have

$$\begin{cases} \boldsymbol{\xi}_1 \cdot \hat{\boldsymbol{\xi}}_{23,\perp} = -2\boldsymbol{\xi}_3 \cdot \hat{\boldsymbol{\xi}}_{23,\perp}, \\ \boldsymbol{\xi}_1 \cdot \hat{\boldsymbol{\xi}}_{1,\perp} = 0. \end{cases} \tag{A.6}$$

Multiplying the first line in (A.6) by c_1 and the second by c_2, then summing, we have by (A.5)

$$\boldsymbol{\xi}_1 \cdot (c_1\hat{\boldsymbol{\xi}}_{23,\perp} + c_2\hat{\boldsymbol{\xi}}_{1,\perp}) = \boldsymbol{\xi}_1 \cdot \mathbf{g} = -2\boldsymbol{\xi}_3 \cdot (c_1\hat{\boldsymbol{\xi}}_{23,\perp}). \tag{A.7}$$

Since $\boldsymbol{\xi}_3 \cdot \hat{\boldsymbol{\xi}}_{23,\perp} < 0$, we have $\boldsymbol{\xi}_1 \cdot \mathbf{g} > 0$ by (A.7). Further, because $\boldsymbol{\xi}_3 \cdot \hat{\boldsymbol{\xi}}_{1,\perp} < 0$, the last term of (A.7) can be bounded below and above by

$$0 < \boldsymbol{\xi}_1 \cdot \mathbf{g} = -2\boldsymbol{\xi}_3 \cdot (c_1\hat{\boldsymbol{\xi}}_{23,\perp}) < -2\boldsymbol{\xi}_3 \cdot (c_1\hat{\boldsymbol{\xi}}_{23,\perp} + c_2\hat{\boldsymbol{\xi}}_{1,\perp})$$

i.e.

$$0 < \boldsymbol{\xi}_1 \cdot \mathbf{g} < -2\boldsymbol{\xi}_3 \cdot \mathbf{g}.$$

Therefore, for $\mathbf{g}$ in region (I)

$$0 < -\frac{\boldsymbol{\xi}_1 \cdot \mathbf{g}}{\boldsymbol{\xi}_3 \cdot \mathbf{g}} < 2.$$

From this we can conclude

$$0 < \frac{U_i(\mathbf{x}_1) - \overline{U}_i}{\overline{U}_i - U_i(\mathbf{x}_3)} < 2.$$

Recognizing that $r = -\frac{\boldsymbol{\xi}_1 \cdot \mathbf{g}}{\boldsymbol{\xi}_3 \cdot \mathbf{g}}$, we have the bounds $0 \leq r \leq 2$ and that $\mathbf{x}' = \boldsymbol{\xi}_3$.

This also holds for vectors in Region (III), in particular, $-\mathbf{g}$. This can be shown by multiplying the numerator and denominator by -1:

$$0 < -\frac{\boldsymbol{\xi}_1 \cdot (-\mathbf{g})}{\boldsymbol{\xi}_3 \cdot (-\mathbf{g})} < 2.$$

If $\mathbf{g}$ lies in region (II) or (IV), the same arguments can be made for the ratio $r = -\frac{\boldsymbol{\xi}_1 \cdot \mathbf{g}}{\boldsymbol{\xi}_2 \cdot \mathbf{g}}$, i.e. $\mathbf{x}' = \mathbf{x}_2$ and $0 \leq r \leq 2$.

□

[1] B. Van Leer, "Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method," *Journal of Computational Physics*, vol. 32, no. 1, pp. 101–136, 1979.

[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.

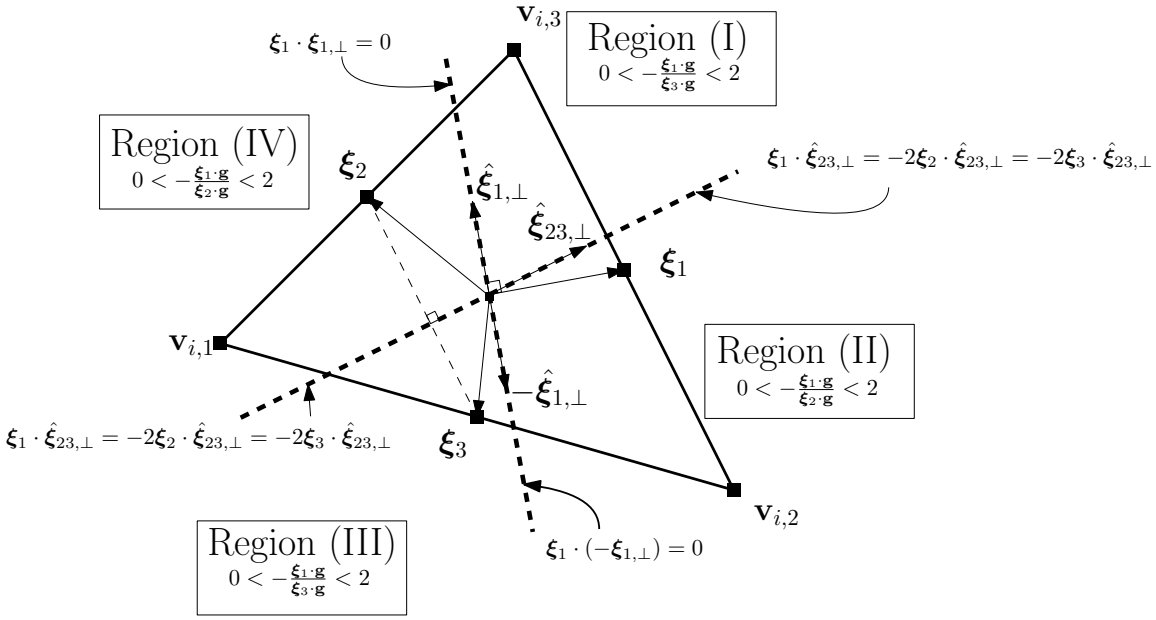


Figure A.24: Geometric set-up for proposition 1.

[4] 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.

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

[6] 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.

[7] S. Spekreijse, "Multigrid solution of monotone second-order discretizations of hyperbolic conservation laws," *Mathematics of Computation*, vol. 49, no. 179, pp. 135–155, 1987.

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

[9] 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.

[10] P. Batten, C. Lambert, and D. Causon, "Positively conservative high-resolution convection schemes for unstructured elements," *International Journal for Numerical Methods in Engineering*, vol. 39, no. 11, pp. 1821–1838, 1996.

[11] J. S. Park, S.-H. Yoon, and C. Kim, "Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids," *Journal of Computational Physics*, vol. 229, no. 3, pp. 788–812, 2010.

[12] J. S. Park and C. Kim, "Multi-dimensional limiting process for finite volume methods on unstructured grids," *Computers & Fluids*, vol. 65, pp. 8–24, 2012.

[13] D. Kuzmin, "Slope limiting for discontinuous Galerkin approximations with a possibly non-orthogonal Taylor basis," *International Journal for Numerical Methods in Fluids*, vol. 71, no. 9, pp. 1178–1190, 2013.

[14] 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.

[15] 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.

[16] B. Cockburn and C.-W. Shu, "TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework," *Mathematics of Computation*, vol. 52, no. 186, pp. 411–435, 1989.

[17] 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.

[18] B. Cockburn and C.-W. Shu, "The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems," *Journal of Computational Physics*, vol. 141, no. 2, pp. 199–224, 1998.

[19] 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.

[20] 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.

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

[22] M. Yang and Z.-J. Wang, "A parameter-free generalized moment limiter for high-order methods on unstructured grids," *Adv. Appl. Math. Mech*, vol. 1, no. 4, pp. 451–480, 2009.

[23] A. Giuliani and L. Krivodonova, "A moment limiter for the discontinuous Galerkin method on unstructured triangular meshes," *Draft available at* `http://www.math.uwaterloo.ca/~agiulian/GIULIANI_moment.pdf`, 2017.

[24] J. S. Park and C. Kim, "Higher-order multi-dimensional limiting strategy for discontinuous Galerkin methods in compressible inviscid and viscous flows," *Computers & Fluids*, vol. 96, pp. 377–396, 2014.

[25] M. Dumbser, O. Zanotti, R. Loubère, and S. Diot, "A posteriori subcell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws," *Journal of Computational Physics*, vol. 278, pp. 47–75, 2014.

[26] T. Barth and M. Ohlberger, "Finite volume methods: foundation and analysis," *Encyclopedia of Computational Mechanics*, 2004.

[27] B. Swartz, "Good neighborhoods for multidimensional van Leer limiting," *Journal of Computational Physics*, vol. 154, no. 1, pp. 237–241, 1999.

[28] V. Aizinger, A. Kosík, D. Kuzmin, and B. Reuter, "Anisotropic slope limiting for discontinuous Galerkin methods," *International Journal for Numerical Methods in Fluids*, 2017.

[29] S. Gottlieb, "On high order strong stability preserving Runge–Kutta and multi-step time discretizations," *Journal of Scientific Computing*, vol. 25, no. 1, pp. 105–128, 2005.

[30] 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.

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

[32] 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.