Third order Maximum-Principle-Satisfying Direct discontinuous Galerkin methods for time dependent convection diffusion equations on unstructured triangle mesh

Zheng Chen,[*] Hongying Huang,[†] Jue Yan[‡]

October 6, 2018

Abstract

We develop 3rd order maximum-principle-satisfying direct discontinuous Galerkin methods [8, 9, 19, 21] for convection diffusion equations on unstructured triangular mesh. We carefully calculate the normal derivative numerical flux across element edges and prove that, with proper choice of parameter pair (β_0, β_1) in the numerical flux, the quadratic polynomial solution satisfies strict maximum principle. The polynomial solution is bounded within the given range and third order accuracy is maintained. There is no geometric restriction on the meshes and obtuse triangles are allowed in the partition. A sequence of numerical examples are carried out to demonstrate the accuracy and capability of the maximum-principle-satisfying limiter.

Keywords: Discontinuous Galerkin methods; Convection diffusion equation; Maximum Principle; Positivity Preserving; Incompressible Navier-Stokes equations;

1 Introduction

In this article, we study direct discontinuous Galerkin finite element method [8] and its variations [9, 19, 21] to solve two-dimensional convection diffusion equations of the form,

$$u_t + \nabla \cdot F(u) - \nabla \cdot (A(u)\nabla u) = 0, \qquad (x, y, t) \in \Omega \times (0, T), \tag{1.1}$$

with zero or periodic boundary conditions. We have spacial domain $\Omega \subset \mathbb{R}^2$ and initial condition $u(x, y, 0) = u_0(x, y)$. The convection flux is denoted as $F(u) = (f(u), g(u))$ and diffusion matrix $A(u) = (a_{ij}(u))$ is assumed symmetric and positive definite.

On the continuous level, solution of (1.1) may satisfy the maximum principle, which states the evolution solution $u(x, y, t)$ being bounded below and above by the given constants, $m \leq u(x, y, t) \leq M$. Here m and M are the lower and upper bounds of the initial and boundary data. It is desirable that the numerical solution satisfies the *discrete maximum principle*. The discrete maximum principle can be considered as a strong L^∞ sense stability result. Failure of preserving the bounds or maintaining the positivity of the numerical solution may lead to ill-posed problems

[*]Department of Mathematics, Iowa State University, Ames, IA 50011. Email: zchen@iastate.edu

[†]School of Mathematics, Physics and Information Science, Zhejiang Ocean University; Zhejiang and Key Laboratory of Oceanographic Big Data Mining & Application of Zhejiang Province, zhoushan, Zhejiang, China, Email: huanghy@lsec.cc.ac.cn

[‡]Department of Mathematics, Iowa State University, Ames, IA 50011. Email: jyan@iastate.edu

and practically cause the computations to blow up. Thus it is attractive to have the numerical solution satisfy discrete maximum principle (or preserve positivity). Solution of equation (1.1) may represent a specific physical meaning and is supposed to be positive, thus negative value approximation loses physical meanings in such cases.

Generally it is very difficult to design high order numerical methods that satisfy discrete maximum principle for convection diffusion equations (1.1). No finite difference method is known to achieve better than second-order accuracy [5, 25, 20] that satisfies discrete maximum principle. Much less is known for higher order methods such as spectral FEM, hp-FEM or finite volume methods [2, 6, 13, 23]. Compared to the elliptic type, more restrictive conditions on mesh are required to obtain discrete maximum principle for the parabolic type equations, see [18, 16, 7, 17].

In this article, we study direct discontinuous Galerkin method [8] and its variations [9, 19, 21], and prove the polynomial solution satisfy discrete maximum principle with third order of accuracy. Discontinuous Galerkin (DG) method is a class of finite element method that use completely discontinuous piecewise functions as numerical approximations. Since the basis functions can be completely discontinuous, these methods have the flexibility that is not shared by standard finite element methods, such as the allowance of arbitrary triangulations with hanging nodes, complete freedom of choosing polynomial degrees in each element (p adaptivity), and extremely local data structure and the resulting high parallel efficiency.

Recently in [24, 25, 26], Zhang and Shu designed a maximum-principle-satisfying limiter for high order DG and finite volume methods for hyperbolic conservation laws. The key step in Zhang and Shu's discussion is to show the polynomial solution average falling in the given minimum and maximum bounds. For hyperbolic type equations, the solution average evolution only relies on the solution polynomial values on the element edges. For diffusion type equations, the evolution of solution average depends on the *solution derivative* values on the edges, thus the technique developed in [24] can not be applied.

In [8], we developed the direct DG method (DDG) as a new diffusion solver. The key contribution of direct DG method is the introduction of numerical flux to approximate the solution derivative at the discontinuous element boundaries. The scheme is directly based on the weak formulation of diffusion equation, thus gains its name the direct DG method. Now let's use the simple 2-D heat equation to go through the main idea of direct DG method,

$$u_t - \triangle u = 0. \tag{1.2}$$

Multiply the heat equation with test function v, integrate over the element K, have integration by parts and formally we obtain,

$$\int_K u_t v \ dxdy - \int_{\partial K} \widehat{u_{\mathbf{n}}} v \ ds + \int_K \nabla u \cdot \nabla v \ dxdy = 0.$$

The numerical flux $\widehat{u_{\mathbf{n}}}$ introduced in [8] is defined as follows,

$$\widehat{u_{\mathbf{n}}} = \widehat{\nabla u \cdot \mathbf{n}} = \beta_0 \frac{[u]}{h_K} + \overline{\frac{\partial u}{\partial \mathbf{n}}} + \beta_1 h_K [u_{\mathbf{nn}}].$$

It involves the jump, the derivative average and higher order derivative jumps to approximate the normal derivative $u_{\mathbf{n}}$ on the element boundary ∂K. Here $\mathbf{n} = (n_1, n_2)$ is the outward unit normal along ∂K and h_K is the diameter of element K. The coefficient pair (β_0, β_1) is chosen to guarantee the convergence of the scheme.

Due to accuracy loss of the original DDG method [8], we further developed DDG method with interface correction in [9] in which optimal $(k + 1)$th order convergence is obtained with any order P^k polynomial approximations. We also have the symmetric [19] and nonsymmetric version [21] of the DDG methods. In this paper, we mainly carry out the maximum principle study on DDG method with interface correction [9] since it is the most efficient solver for time dependent diffusion equations. The maximum principle arguments discussed in the following sections also apply to DDG method [8] and its symmetric and nonsymmetric variations [19, 21].

In [22], we prove the DDG solutions satisfy discrete maximum principle on rectangular and uniform triangular meshes with 3rd order of accuracy. We use an algebraic methodology and a monotonicity argument to show the polynomial solution average being bounded within the given range. The DG polynomial solution was written out in the Lagrange format with the unknowns carefully chosen on the element. With Euler forward in time, we show the solution average at next time level depends on the current time level solution values in a monotone fashion. For unstructured mesh with possible obtuse triangles, it is very hard to identify six degrees of freedom to represent the P^2 quadratic polynomial solution such that the monotone argument in [22] can be applied.

In this article we extend maximum principle studies of (1.1) on unstructured triangle mesh. Again let's use the heat equation to illustrate the new technique to carry out the proof. Notice that the key step of the discussion is to show the solution average falling in the given range. Take test function $v = 1$ in the DDG scheme and discretize in time with Euler forward, we have the solution average evolving in time as,

$$\overline{u}_K^{n+1} = \overline{u}_K^n + \frac{\Delta t}{area(K)} \int_{\partial K} \widehat{u_{\mathbf{n}}} \, ds,$$

with the average defined as $\overline{u}_K^n = \frac{1}{area(K)} \int_K u_K^n(x, y) \, dxdy$ and $u_K^n(x, y)$ as the polynomial solution at time step t_n in element K.

Instead of identifying suitable locations as degrees of freedom and writing out $u_K^n(x, y)$ in the Lagrange format as in [22], we directly calculate the normal derivative flux $\widehat{u_{\mathbf{n}}}$ from the given solution values in element K and its neighbors. Given suitable choice of coefficient pair (β_0, β_1) in the numerical flux, we can bound the solution average $\overline{u}_K^{n+1} \in [m, M]$ at time level t_{n+1} once we know $u_K^n(x, y) \in [m, M]$ at previous time level t_n. Finally we borrow the maximum principle discussion of [25] to show the DG polynomial solution of general convection diffusion equations (1.1) satisfy strict maximum principle with 3rd order of accuracy. A sequence of numerical examples are carried out to demonstrate the DG solutions are strictly bounded by the given values and at the same time maintain the 3rd order accuracy. Solutions to nonlinear porous medium equations with nonnegative initial data are maintained sharply nonnegative. Examples of incompressible Navier-Stokes equations with high Reynold numbers are tested. Overshoot and undershoots are removed with maximum principle limiter applied.

The key feature of direct DG methods is the introduction of numerical flux $\widehat{u_{\mathbf{n}}}$ that approximates the solution derivative $u_{\mathbf{n}}$ on the discontinuous element boundary. This gives direct DG methods the extra flexibility and advantage over IPDG method [1] and LDG method [4]. Following this maximum principle framework, both IPDG method and LDG method can be proved to satisfy maximum principle with up to second order of accuracy. In [27], DG solutions with piecewise linear polynomial approximations are shown satisfying maximum principle on unstructured triangle mesh.

The paper is organized as follows. We first review the scheme formulation of direct DG method with interface correction [9] in section 2. In Section 3, we prove the direct DG solutions satisfy discrete maximum principle with 3rd order accuracy. We conduct numerical tests to validate the theoretical results in Section 4. Section 5 serves as the Appendix in which we provide one way to

construct a specific quadrature rule for quadratic polynomials with selected points as quadrature points.

2   Direct DG method with interface correction

We first recall the scheme formulation of direct DG method with interface correction [9] for two-dimensional diffusion equations,

$$u_t - \nabla \cdot (A(u)\nabla u) = 0, \qquad (x, y, t) \in \Omega \times (0, T), \tag{2.1}$$

with initial condition $u(x, y, 0) = u_0(x, y)$ and zero or periodic boundary conditions. The complete scheme formulation of convection diffusion equation (1.1) will be laid out toward the end of this section. We should specify the DG method is for spatial discretization and we will incorporate high order TVD Runge-Kutta methods [15, 14] to march forward the solution in time. As an explicit scheme, our method is thus more efficient for convection dominated problems. However, the extremely local dependency allows a very efficient parallelization and dramatically improves the efficiency of the explicit method.

Let $\mathcal{T}_h$ be a shape-regular partition of the polygonal domain Ω into triangle elements $\{K\}_{K \in \mathcal{T}_h}$ with $\overline{\Omega} = \cup_{K \in \mathcal{T}_h} \overline{K}$. By $h_K = \text{diam}(K)$, we denote the diameter of the triangle element $K \in \mathcal{T}_h$. We denote $h = \max_{K \in \mathcal{T}_h} h_K$ as the mesh size of the partition. We have $P^k(K)$ representing the kth degree polynomial space on element K. The DG solution space is defined as,

$$\mathbb{V}_h^k = \{v \in L^2(\Omega) : v|_K \in P^k(K), \forall K \in \mathcal{T}_h\}.$$

Suppose K and K' are two adjacent triangles and share one common edge e. There are two traces of v along the edge e, where we add or subtract those values to obtain the average and the jump. The outward normal vector from K to its neighbor element K' is denoted by $\mathbf{n} = (n_1, n_2)$. Now the average and jump of v on the edge e are defined as follows,

$$\overline{v} = \frac{1}{2}\left(v|_K + v|_{K'}\right), \qquad [v] = v|_{K'} - v|_K.$$

The original DDG scheme of (2.1) defined in [8] is to find DG solution $u \in \mathbb{V}_h^k$, such that for any test function $v \in \mathbb{V}_h^k$ we have,

$$\int_K u_t v \, dxdy - \int_{\partial K} (A\widehat{(u)\nabla u} \cdot \mathbf{n})v \, ds + \int_K A(u)\nabla u \cdot \nabla v \, dxdy = 0, \quad \forall K \in \mathcal{T}_h. \tag{2.2}$$

The numerical flux $A\widehat{(u)\nabla u} \cdot \mathbf{n}$ (equation (3.7) of [21] in dimension-by-dimension format) along the element edge is defined as,

$$A\widehat{(u)\nabla u} \cdot \mathbf{n} = \left(\widehat{b_{11}(u)_x} + \widehat{b_{12}(u)_y}\right) n_1 + \left(\widehat{b_{21}(u)_x} + \widehat{b_{22}(u)_y}\right) n_2,$$

where $b_{ij}(u) = \int a_{ij}(u)du$ with $a_{ij}(u)$ as the diffusion matrix $A(u)$ entry. The outward normal is given with $\mathbf{n} = (n_1, n_2)$. Similar to (3.7) of [21], for example, the numerical flux $\widehat{b_{11}(u)_x}$ is calculated with formula,

$$\widehat{b_{11}(u)_x} = \beta_0 \frac{[b_{11}(u)]}{h_K} n_1 + \overline{b_{11}(u)_x} + \beta_1 h_K \left\{[b_{11}(u)_{xx}]n_1 + [b_{11}(u)_{yx}]n_2\right\}.$$

4

The coefficient pair (β_0, β_1) should be chosen carefully to ensure the stability and convergence of the scheme. Again we have h_K as element K's diameter or the length of edge ∂K. Notice in [22], we further require the entries $a_{ij}(u) \geq 0$ to carry out the discrete maximum principle proof.

In this article, we simplify the calculation of numerical flux $\widehat{A(u)\nabla u} \cdot \mathbf{n}$ to the following,

$$\widehat{A(u)\nabla u} \cdot \mathbf{n} = \widehat{\nabla u \cdot \boldsymbol{\gamma}} = \beta_0 \frac{[u]}{h_K} + \overline{u_{\boldsymbol{\gamma}}} + \beta_1 h_K [u_{\boldsymbol{\gamma\gamma}}], \tag{2.3}$$

where $\boldsymbol{\gamma} = A^T(u)\mathbf{n}$ is a vector pointing from element K into its neighbor along the edge ∂K. The simplification holds true since $(A(u)\nabla u) \cdot \mathbf{n} = \nabla u \cdot (A^T(u)\mathbf{n})$ and $A(u)$ is positive definite. We should point out the simplified version (2.3) is very important to carry out the maximum principle discussion in the following sections. Now the DDG interface correction [9] of (2.1) is defined to find solution $u \in \mathbb{V}_h^k$, such that for any test function $v \in \mathbb{V}_h^k$ we have,

$$\int_K u_t v \, dxdy - \int_{\partial K} (\widehat{A(u)\nabla u} \cdot \mathbf{n})v \, ds + \int_K A(u)\nabla u \cdot \nabla v \, dxdy + \int_{\partial K} \overline{A(v)\nabla v} \cdot \mathbf{n}[u] \, ds = 0, \forall K \in \mathcal{T}_h, \tag{2.4}$$

with the numerical flux (2.3). The last term in (2.4) is the the extra added interface correction. Notice that the test function v is taken to be zero outside the element K, thus the derivative average degenerates to $\overline{A(v)\nabla v} = \frac{1}{2}A(v)\nabla v|_K$ on the edge ∂K.

The complete scheme formulation of (1.1) with DDG interface correction follows,

$$\int_K u_t v \, dxdy + \int_{\partial K} \widehat{F \cdot \mathbf{n}} v \, ds - \int_K F \cdot \nabla v \, dxdy$$
$$= \int_{\partial K} (\widehat{A(u)\nabla u} \cdot \mathbf{n})v \, ds - \int_K A(u)\nabla u \cdot \nabla v \, dxdy - \int_{\partial K} \overline{A(v)\nabla v} \cdot \mathbf{n}[u] \, ds, \tag{2.5}$$

with the convection term Lax-Friedrichs flux defined as,

$$\widehat{F \cdot \mathbf{n}} = \frac{1}{2}\left(F(u_K) \cdot \mathbf{n} + F(u_{K'}) \cdot \mathbf{n} - \alpha(u_{K'} - u_K)\right), \quad \text{with} \quad \alpha = \max_u |\mathbf{F}'(u) \cdot \mathbf{n}|.$$

3 Maximum-Principle-Satisfying DDG methods

In this section we prove DDG polynomial solutions of nonlinear diffusion equations (2.1) satisfy discrete maximum principle with the M-P-S limiter applied. We first discuss the linear case on unstructured triangle mesh in section 3.1. Then we extend the study to nonlinear diffusion equations in section 3.2.

Notice that the second derivative jump term has no contribution to the calculation of the numerical flux with low order P^0 and P^1 approximations. The scheme of DDG method with interface correction (2.4) degenerates to IPDG methods with low order approximations. In this paper, we focus on P^2 quadratic polynomial approximations with 3rd order of accuracy. We skip the trivial piecewise constant case and refer to [27] for 2rd order linear approximations.

On the continuous level the maximum principle states that $m \leq u(x, y, t) \leq M$, given m and M as the minimum and maximum of the initial data $u(x, y, 0) = u_0(x, y)$ and boundary data. We have $u(x, y, t_n)$ to denote the exact solution at time level t_n and $u_K^n(x, y)$ as the polynomial solution on element K and at time level t_n. Our goal is to prove the polynomial solution satisfy $m \leq u_K^n(x, y) \leq M$ without losing the 3rd order accuracy at all time levels. We can simplify the discussion to Euler forward time discretization, since the full scheme (high order strong stability

preserving (SSP) Runge-Kutta method) is a convex combination of Euler forward scheme. For example, the third order SSP Runge-Kutta method in [14] is

$$
\begin{cases}
u^{(1)} = u^n + \Delta t H(u^n) \\
u^{(2)} = \frac{3}{4}u^n + \frac{1}{4}(u^{(1)} + \Delta t H(u^{(1)})) \\
u^{n+1} = \frac{1}{3}u^n + \frac{2}{3}(u^{(2)} + \Delta t H(u^{(2)}))
\end{cases}
$$

Now assume at time level t_n, we have 1) the DDG solution is 3rd order accurate; and 2) we have $u_K^n(x, y) \in [m, M]$ on all elements. The goal is to prove the solution polynomial $u_K^{n+1}(x, y)$ at next time level t_{n+1} still stay inside the bounds $[m, M]$ without losing accuracy. To carry out this study we need to consider following two steps:

1. *to prove the polynomial solution average $\overline{u}_K^{n+1}$ stay inside the bounds $[m, M]$;*

2. *to prove the whole polynomial $u_K^{n+1}(x, y)$ stay inside $[m, M]$ without losing accuracy.*

The most challenging and the major step is to show the polynomial average $\overline{u}_K^{n+1}$ falling in $[m, M]$. For the second step, we simply apply a linear scaling limiter [11] to $u_K^{n+1}(x, y)$ and obtain a modified polynomial $\widetilde{u}_K^{n+1}(x, y)$ such that the whole polynomial $\widetilde{u}_K^{n+1}(x, y) \in [m, M]$ without losing accuracy. we refer to [25] for the proof of this accuracy preserving limiter. We have the DDG solution approximates the exact solution with 3rd order accuracy, thus the polynomial solution can only jump out of the bounds $[m, M]$ in the scale of h^3 with h as the mesh size. The limiter [11] is applied to compress and squeeze the polynomial in the scale of h^3 and put it back into the bounds $[m, M]$. The extra cost to preserve maximum principle is to apply the limiter [11] to maintain the bounds.

3.1 Linear diffusion equation

In this section, we prove the DDG quadratic polynomial solutions of heat equation (1.2) satisfy discrete maximum principle. Again we focus on the first step and investigate under what conditions the solution average $\overline{u}_K^{n+1} \in [m, M]$ given $u_K^n(x, y) \in [m, M]$ on all elements.

The scheme formulation of DDG with interface correction for (1.2) is to find DG solution $u \in \mathbb{V}_h^k$, such that for any test function $v \in \mathbb{V}_h^k$ we have,

$$
\int_K u_t v \, dxdy - \int_{\partial K} \widehat{u_{\mathbf{n}}} v \, ds + \int_K \nabla u \cdot \nabla v \, dxdy + \int_{\partial K} \overline{v_{\mathbf{n}}}[u] \, ds = 0, \quad \forall K \in \mathcal{T}_h. \tag{3.1}
$$

The numerical flux $\widehat{u_{\mathbf{n}}}$ on the element boundary ∂K is given with,

$$
\widehat{u_{\mathbf{n}}} = \beta_0 \frac{[u]}{h_K} + \overline{u_{\mathbf{n}}} + \beta_1 h_K [u_{\mathbf{nn}}]. \tag{3.2}
$$

To obtain the solution average evolution, we take test function $v = 1$ in (3.1), discretize in time with forward Euler and formally we have,

$$
\overline{u}_K^{n+1} = \overline{u}_K^n + \frac{\Delta t}{area(K)} \int_{\partial K} \widehat{u_{\mathbf{n}}} \, ds. \tag{3.3}
$$

The solution average is given with $\overline{u}_K^n = \frac{1}{area(K)} \int_K u_K^n(x, y) \, dxdy$ and Δt is denoted as the time step size. The average $\overline{u}_K^n$ can be calculated out exactly, since $u_K^n(x, y)$ is a quadratic polynomial.

Thus we see the quantity $\overline{u}_K^{n+1}$ of (3.3) is essentially determined by the integral of $\widehat{u_{\mathbf{n}}}$ on the three edges. Recall that the numerical flux $\widehat{u_{\mathbf{n}}}$ of (3.2) involves the solution jump, normal derivative average and second order normal derivative jumps on ∂K. The quantity $\overline{u}_K^{n+1}$ eventually is a function of the four solution polynomials that spread out in K and its three neighbors.

For uniform triangular mesh in [22], we pick six solution values on each element and write out the P^2 solution polynomial in Lagrange format, then use a monotone argument to bound $\overline{u}_K^{n+1} \in [m, M]$. For arbitrary triangular mesh, it's hard to identify such six points inside each triangle. We will use a new idea to bound $\overline{u}_K^{n+1}$.

We observe that the three quantities, namely u, $u_{\mathbf{n}}$ and $u_{\mathbf{nn}}$ restricted on the edges from each element, contribute to the calculation of $\widehat{u_{\mathbf{n}}}$. If we manage to calculate u, $u_{\mathbf{n}}$ and $u_{\mathbf{nn}}$ from the given solution polynomials, we can easily bound $\overline{u}_K^{n+1} \in [m, M]$ once we have $u_K^n(x, y) \in [m, M]$ for all K.

Figure 1: Left: K and its neighbor elements. Right: selected points to calculate $\widehat{u_{\mathbf{n}}}$ on edge AB.

Let's use Figure 1 to illustrate the solution points selected to calculate u, $u_{\mathbf{n}}$ and $u_{\mathbf{nn}}$. For example, we consider the edge AB shared by K and its neighbor element K_3 (the right one in Figure 1). Notice that $u_{K_3}^n(x, y)$ is a P^2 polynomial. The three points, namely $\mathbf{x}_{K_3,1}$, $\mathbf{x}_{K_3,2}$, $\mathbf{x}_{K_3,3}$, are enough to represent the restriction of $u_{K_3}^n(x, y)$ on edge AB. The second normal derivative $u_{\mathbf{nn}}$ degenerates to a constant, and we can use three points $\mathbf{x}_{K_3,3}^n$, $\mathbf{x}_{K_3,4}^n$, $\mathbf{x}_{K_3,5}^n$ on the normal line through the edge center to calculate $u_{\mathbf{nn}}$. The first normal derivative $u_{\mathbf{n}}$ on AB is a linear polynomial, thus the same three points on the normal line are good enough to calculate the line integral of $u_{\mathbf{n}}$. With this new idea to calculate the numerical flux $\widehat{u_{\mathbf{n}}}$ of (3.3), we are ready to bound the average $\overline{u}_K^{n+1}$.

Theorem 3.1. *Consider DDG scheme with interface correction (3.1) - (3.2) with P^2 quadratic approximations on unstructured triangular mesh. Given $u_K^n(x, y)$ in the range of $[m, M]$ for all K, we have $\overline{u}_K^{n+1} \in [m, M]$ provided,*

$$\beta_0 \geq \frac{9}{4} - 6\beta_1, \quad \frac{1}{8} \leq \beta_1 \leq \frac{1}{4}, \quad \lambda = \frac{\Delta t}{area d(K)} \leq A(\beta_0, \beta_1, \check{\theta}, \hat{\theta}). \tag{3.4}$$

Here (β_0, β_1) is the coefficient pair in the numerical flux (3.2). We have $\check{\theta}$ and $\hat{\theta}$ denoted as the

maximum and minimum angle of the partition $\mathcal{T}_h$, and A is a function of β_0, β_1, $\check{\theta}$ and $\hat{\theta}$, i.e.,

$$A = \tan(\hat{\theta}) \cdot \min\left\{\frac{w_1}{72(1-4\beta_1)}, \frac{\tan(\hat{\theta})}{\tan(\check{\theta})} \min\left(\frac{1}{6(8\beta_1-1)}, \frac{w_1}{8(\beta_0 - \frac{9}{4} + 6\beta_1)}, \frac{w_1}{4\beta_0}\right)\right\}, \qquad (3.5)$$

where $w_1 = \frac{2}{81}$ as shown in (5.4).

Proof. To bound $\bar{u}_K^{n+1}$ of (3.3), we see it's important to carefully calculate the numerical flux $\widehat{u_{\mathbf{n}}}$ on ∂K. From the numerical flux formula (3.2) of the DDG schemes, we have h_K taken as the element diameter or the length of the edge ∂K. To simplify the proof, here we modify h_K to incorporate with the mesh geometrical information. We should comment that numerically we observe no difference with either choice of h_K.

Again we use edge AB in Figure 1 to illustrate the definition of h_K chosen in the numerical flux formula (3.2). Let's have the parametric equation $\mathbf{r}(t) = t\mathbf{n} + \mathbf{x}_{K,3}$, $t \in \mathbb{R}$ to represent the normal line through the edge center. And we have points $\mathbf{x}_K^0$ and $\mathbf{x}_{K_3}^0$ as the intersection of the normal line with the other two edges of K and K_3. Restricted on edge $\partial K = AB$, we take $h_K = h_{AB} = \min\left\{\|\mathbf{x}_{K,3} - \mathbf{x}_K^0\|, \|\mathbf{x}_{K_3,3} - \mathbf{x}_{K_3}^0\|\right\}$ and we have,

$$\int_{AB} \widehat{u_{\mathbf{n}}} \, ds = \int_{AB} \beta_0 \frac{[u]}{h_{AB}} \, ds + \int_{AB} \overline{u_{\mathbf{n}}} \, ds + \int_{AB} \beta_1 h_{AB} [u_{\mathbf{nn}}] \, ds. \qquad (3.6)$$

To calculate $u_{\mathbf{nn}}$, we pick two more points along the normal line from each side. We have points $\mathbf{x}_{K,5}(t = -h_{AB})$ and $\mathbf{x}_{K,4}(t = -\frac{1}{2}h_{AB})$ taken in element K, and points $\mathbf{x}_{K_3,5}(t = h_{AB})$ and $\mathbf{x}_{K_3,4}(t = \frac{1}{2}h_{AB})$ taken in element K_3, as shown in Figure 1. Furthermore we denote $u_{K,1}, \ldots, u_{K,5}$ as the $u_K^n(x, y)$ quadratic polynomial solution values on points $\mathbf{x}_{K,1}, \ldots, \mathbf{x}_{K,5}$. As discussed previously, the five points $u_{K,1}, \ldots, u_{K,5}$ are enough to calculate $\int_{AB} \widehat{u_{\mathbf{n}}} ds$ from the side of element K. Similarly, the five points $u_{K_3,1}, \ldots, u_{K_3,5}$ are enough to calculate $\int_{AB} \widehat{u_{\mathbf{n}}} ds$ from the side of element K_3. Essentially the quantity $\int_{AB} \widehat{u_{\mathbf{n}}} ds$ can be explicitly written out in terms of the ten solution values spread out in elements K and K_3 as follows,

$$\int_{AB} \beta_0 \frac{[u]}{h_{AB}} ds = \frac{\beta_0 l_{AB}}{6h_{AB}} \left\{(u_{K_3,1} + u_{K_3,2} + 4u_{K_3,3}) - (u_{K,1} + u_{K,2} + 4u_{K,3})\right\}$$

$$\int_{AB} \overline{u_{\mathbf{n}}} ds = \frac{l_{AB}}{2h_{AB}} \left\{(-3u_{K_3,3} - u_{K_3,5} + 4u_{K_3,4}) + (3u_{K,3} + u_{K,5} - 4u_{K,4})\right\}$$

$$\int_{AB} \beta_1 h_{AB} [u_{\mathbf{nn}}] ds = \frac{4\beta_1 l_{AB}}{h_{AB}} \left\{(u_{K_3,3} + u_{K_3,5} - 2u_{K_3,4}) - (u_{K,3} + u_{K,5} - 2u_{K,4})\right\}. $$

$$(3.7)$$

Here we have l_{AB} denoting the length of edge AB. Finally the average $\bar{u}_K^{n+1}$ of (3.3) can be written out as a function of the solution values that spread out in element K and its neighbors K_1, K_2, K_3. Thus formally we have,

$$\bar{u}_K^{n+1} = \bar{u}_K^n + \frac{\Delta t}{area(K)} \left\{\int_{AB} \widehat{u_{\mathbf{n}}} ds + \int_{BC} \widehat{u_{\mathbf{n}}} ds + \int_{CA} \widehat{u_{\mathbf{n}}} ds\right\}$$

$$= H\left\{u_K^n(\cdot, \cdot), u_{K_1}^n(\cdot, \cdot), u_{K_2}^n(\cdot, \cdot), u_{K_3}^n(\cdot, \cdot)\right\}. \qquad (3.8)$$

8

The functional $H(\cdots)$ involves 28 arguments with 15 points from K_1, K_2, K_3 and 13 points from K. The first 12 points from K are selected from the calculation of $\int_{\partial K} \widehat{u_{\mathbf{n}}} ds$, see Figure 1. The 13th one is to be selected by the quadrature rule for cell average $\overline{u}_K^n$, see Appendix 5.1.

Our goal here is to prove solution average $\overline{u}_K^{n+1} \in [m, M]$ given $u_K^n(x, y) \in [m, M]$ on all elements. Again, we use a monotone argument showing $\overline{u}_K^{n+1}$ is a convex combination of the selected solution points. To study the conditions to guarantee that $H(\Uparrow, \Uparrow, \Uparrow, \Uparrow)$ is monotonically increasing on the total 28 arguments, it is enough to check out the ten points selected inside K and K_3 of (3.8). We first check the five points selected on element K_3. From (3.7) - (3.8) we have,

$$\frac{\partial H}{\partial u_{K_3,1}} = \frac{\partial H}{\partial u_{K_3,2}} = \lambda \frac{l_{AB}}{h_{AB}} \frac{\beta_0}{6}, \qquad \frac{\partial H}{\partial u_{K_3,3}} = \lambda \frac{l_{AB}}{h_{AB}} (\frac{2}{3} \beta_0 - \frac{3}{2} + 4\beta_1),$$

$$\frac{\partial H}{\partial u_{K_3,4}} = \lambda \frac{2l_{AB}}{h_{AB}} (1 - 4\beta_1), \qquad \frac{\partial H}{\partial u_{K_3,5}} = \lambda \frac{l_{AB}}{2h_{AB}} (8\beta_1 - 1).$$

With $\lambda = \frac{\Delta t}{area(K)} > 0$, we only need $\beta_0 \geq \frac{9}{4} - 6\beta_1$ and $\frac{1}{8} \leq \beta_1 \leq \frac{1}{4}$ to guarantee the coefficients of the five solution values in K_3 being non-negative. Before we carry out the discussion on the five points in element K, we need an inequality (refer to Appendix 5.1) which reflects geometrical property of the mesh partition.

$$h_{AB} = \frac{l_{AB}}{2} \tan(\min(\theta_1, \theta_2, \theta_4, \theta_5)) \geq \frac{l_{AB}}{2} \tan(\hat{\theta}).$$

From (3.7) and (3.8), we have,

$$\frac{\partial H}{\partial u_{K,1}} = \frac{\partial \overline{u}_K^n}{\partial u_{K,1}} - \lambda \frac{\beta_0}{6} \left(\frac{l_{AB}}{h_{AB}} + \frac{l_{CA}}{h_{CA}} \right) \geq \frac{\partial \overline{u}_K^n}{\partial u_{K,1}} - \lambda \frac{2\beta_0}{3 \tan(\hat{\theta})}$$

$$\frac{\partial H}{\partial u_{K,2}} = \frac{\partial \overline{u}_K^n}{\partial u_{K,2}} - \lambda \frac{\beta_0}{6} \left(\frac{l_{AB}}{h_{AB}} + \frac{l_{BC}}{h_{BC}} \right) \geq \frac{\partial \overline{u}_K^n}{\partial u_{K,2}} - \lambda \frac{2\beta_0}{3 \tan(\hat{\theta})}$$

$$\frac{\partial H}{\partial u_{K,3}} = \frac{\partial \overline{u}_K^n}{\partial u_{K,3}} - \lambda \frac{l_{AB}}{h_{AB}} \left(\frac{2}{3} \beta_0 - \frac{3}{2} + 4\beta_1 \right) \geq \frac{\partial \overline{u}_K^n}{\partial u_{K,3}} - \lambda \frac{4 \left(\beta_0 - \frac{9}{4} + 6\beta_1 \right)}{3 \tan(\hat{\theta})}$$

$$\frac{\partial H}{\partial u_{K,4}} = \frac{\partial \overline{u}_K^n}{\partial u_{K,4}} - \lambda \frac{2l_{AB}}{h_{AB}} (1 - 4\beta_1) \geq \frac{\partial \overline{u}_K^n}{\partial u_{K,4}} - \lambda \frac{4 (1 - 4\beta_1)}{\tan(\hat{\theta})}$$

$$\frac{\partial H}{\partial u_{K,5}} = \frac{\partial \overline{u}_K^n}{\partial u_{K,5}} - \lambda \frac{l_{AB}}{2h_{AB}} (8\beta_1 - 1) \geq \frac{\partial \overline{u}_K^n}{\partial u_{K,5}} - \lambda \frac{8\beta_1 - 1}{\tan(\hat{\theta})}.$$

To guarantee the coefficients of the five solution points in K being non-negative, we need a special quadrature rule with all positive weights on all the 12 selected points in K. We refer to Appendix 5.1 for details of this quadrature rule. The 13th point $u_{K,13}$ is selected by the quadrature rule also with positive weight. With CFL restriction (3.4)- (3.5) and the quadrature weights (5.1), we see $H(\cdots)$ is monotonically increasing on $u_{K,1}$, $u_{K,2}$, $u_{K,3}$, $u_{K,4}$ and $u_{K,5}$. Similar argument applies to edge BC and edge CA, involving solution values in elements K_1 and K_2. Easily we see functional $H(\cdots)$ is monotonically increasing w.r.t. all 28 point values. With the consistency and the monotonicity of $H(\cdots)$, we obtain,

$$m = H(m, \cdots, m) \leq \overline{u}_K^{n+1} = H(\cdots) \leq H(M, \cdots, M) = M.$$

provided that $m \leq u_K^n(x, y), u_{K_1}^n(x, y), u_{K_2}^n(x, y), u_{K_3}^n(x, y) \leq M$.

$\square$

3.2 Nonlinear diffusion equation

In this section we extend the study of 3rd order M-P-S DDG scheme with interface correction (2.3) - (2.4) to general nonlinear diffusion equations (2.1) on unstructured triangle mesh. Again, our goal is to bound the solution average $\overline{u}_K^{n+1} \in [m, M]$ provided $u_K^n(x, y)$ in the range of $[m, M]$. Take test function $v = 1$ in (2.4) and discretize in time with Euler forward, we have the solution average evolving in time as,

$$\overline{u}_K^{n+1} = \overline{u}_K^n + \frac{\Delta t}{area(K)} \int_{\partial K} (A\widehat{(u)\nabla u} \cdot \mathbf{n}) \, ds = \overline{u}_K^n + \frac{\Delta t}{area(K)} \int_{\partial K} \widehat{\nabla u \cdot \boldsymbol{\gamma}} \, ds. \qquad (3.9)$$

Figure 2: Selected points along direction $\boldsymbol{\gamma} = \mathbf{A}^T(u^n(\mathbf{x}_{i,j}))\mathbf{n}_i$ for representing the numerical flux

As discussed previously in (2.3), we apply a new way to calculate numerical flux $A\widehat{(u)\nabla u} \cdot \mathbf{n} = \widehat{\nabla u \cdot \boldsymbol{\gamma}}$ with $\boldsymbol{\gamma} = A^T(u)\mathbf{n}$ as a vector pointing from K into its neighbor along the edge. This is true because the diffusion matrix $A(u)$ is positive definite and we have $\boldsymbol{\gamma} \cdot \mathbf{n} = A^T(u)\mathbf{n} \cdot \mathbf{n} > 0$. In a word, γ is a vector always pointing into its neighbor element. To bound $\overline{u}_K^{n+1}$, we need to manage to calculate $\int_{\partial K} \widehat{\nabla u \cdot \boldsymbol{\gamma}} \, ds$ on the three edges of element K. Notice that vector $\boldsymbol{\gamma}$ is a nonlinear function of the solution, thus we apply a quadrature rule to calculate $\int_{\partial K} \widehat{\nabla u \cdot \boldsymbol{\gamma}} \, ds$. For example, we consider 2-point Gaussian quadrature rule along each edge to approximate the line integral. Let's use point $\mathbf{x}_{i,j}$ to denote the jth Gaussian point on the ith edge. For each Gaussian point $\mathbf{x}_{i,j}$, shown in Figure 2, we see six solution values are enough to calculate the numerical flux,

$$\widehat{\nabla u \cdot \boldsymbol{\gamma}}|_{\mathbf{x}_{i,j}} = \widehat{u_{\boldsymbol{\gamma}}}|_{\mathbf{x}_{i,j}} = \beta_0 \frac{[u]}{h_{i,j}} + \overline{u_{\boldsymbol{\gamma}}} + \beta_1 h_{i,j} [u_{\boldsymbol{\gamma}\boldsymbol{\gamma}}]\Big|_{\mathbf{x}_{i,j}}, \qquad (3.10)$$

where $\boldsymbol{\gamma} = A^T(u^n(\mathbf{x}_{i,j}))\mathbf{n}_i$ and $h_{i,j}$ as the shortest distance from point $\mathbf{x}_{i,j}$ to the other edges of K and K_i along $\boldsymbol{\gamma}$. Again, we bound $\overline{u}_K^{n+1}$ by showing it is a convex combination of polynomial solution values that spread in element K and its three neighbors K_1, K_2 and K_3.

Theorem 3.2. *(Nonlinear diffusion equation) Consider DDG scheme with interface correction (2.3) - (2.4) with P^2 quadratic approximations on a triangular mesh. Given $u_K^n(x, y)$ in the range of $[m, M]$, we have $\overline{u}_K^{n+1} \in [m, M]$ provided,*

$$\beta_0 \geq \frac{3}{2} - 4\beta_1, \quad \frac{1}{8} \leq \beta_1 \leq \frac{1}{4}, \quad \lambda = \frac{\Delta t}{area(K)} \leq A(\beta_0, \beta_1, \check{\theta}, \hat{\theta}). \qquad (3.11)$$

Again (β_0, β_1) is the coefficient pair in the numerical flux (2.3), and $\check{\theta}$ and $\hat{\theta}$ as the maximum and minimum angles of the partition $\mathcal{T}_h$. Function A depends on β_0, β_1, $\check{\theta}$ and $\hat{\theta}$ as,

$$A = \sin(\hat{\theta}) \cdot \frac{3 - \sqrt{3}}{3} w_0 \cdot \min\left\{\frac{1}{2\beta_0 + 8\beta_1 + 3}, \frac{1}{8\beta_1 + 1}\right\}, \tag{3.12}$$

where w_0 is the minimum quadrature weight in the quadrature rule.

Proof. Similar to Theorem 3.1, it suffices to study the monotonicity of $\overline{u}_K^{n+1}$ with respect to the points selected to evaluate the right hand side of (3.9). Specifically it is enough to study the Gaussian points that are used to approximate the line integral on the edges. For one Gaussian point $\mathbf{x}_{i,j}$, as shown in Figure 2, six solution points selected along $\gamma = A^T(u^n(\mathbf{x}_{i,j}))\mathbf{n}_i$ direction are enough to calculate the numerical flux (3.10). We denote u^1, u^2, u^3 as the three solution values of $u_K^n(x, y)$ selected in K and u_o^1, u_o^2, u_o^3 as the ones selected in K_i. Again we have $h_{i,j}$ denoting the shortest distance from point $\mathbf{x}_{i,j}$ to the other edges of K and K_i along $\boldsymbol{\gamma}$.

Now at the Gaussian point $\mathbf{x}_{i,j}$, we can explicitly write out the value of $\widehat{u_\gamma}$ of (3.10) with,

$$[u] = u_o^1 - u^1$$

$$\overline{u_\gamma} = \frac{1}{2}\left(\frac{4u^2 - 3u^1 - u^3}{h_{i,j}} + \frac{4u_o^2 - 3u_o^1 - u_o^3}{h_{i,j}}\right) = \frac{1}{2h_{i,j}}\left(4u^2 - 3u^1 - u^3 + 4u_o^2 - 3u_o^1 - u_o^3\right)$$

$$[u_{\gamma\gamma}] = \frac{u_o^1 + u_o^3 - 2u_o^2}{\left(\frac{h_{i,j}}{2}\right)^2} - \frac{u^1 + u^3 - 2u^2}{\left(\frac{h_{i,j}}{2}\right)^2} = \frac{4}{h_{i,j}^2}\left[u_o^1 + u_o^3 - 2u_o^2 - (u^1 + u^3 - 2u^2)\right].$$

Introduce notation $\alpha^{i,j} = \frac{\Delta t}{area(K)}l_K^i\omega_j$ with l_K^i as the ith edge length and ω_j as the jth Gaussian point weight. Since we use 2-point Gaussian quadrature rule, the weight $\omega_j \equiv 1$. From (3.9), we have,

$$\frac{\partial \overline{u}_K^{n+1}}{\partial u_o^3} = \frac{\alpha^{i,j}}{2h_{i,j}}(8\beta_1 - 1), \quad \frac{\partial \overline{u}_K^{n+1}}{\partial u_o^2} = \frac{2\alpha^{i,j}}{h_{i,j}}(1 - 4\beta_1),$$

$$\frac{\partial \overline{u}_K^{n+1}}{\partial u_o^1} = \frac{\alpha^{i,j}}{h_{i,j}}(\beta_0 + 4\beta_1 - \frac{3}{2}), \quad \frac{\partial \overline{u}_K^n}{\partial u^3} = \frac{\partial \overline{u}_K^n}{\partial u^3} - \frac{\alpha^{i,j}}{2h_{i,j}}(8\beta_1 + 1),$$

$$\frac{\partial \overline{u}_K^{n+1}}{\partial u^2} = \frac{\partial \overline{u}_K^n}{\partial u^2} + \frac{2\alpha^{i,j}}{h_{i,j}}(4\beta_1 + 1), \quad \frac{\partial \overline{u}_K^{n+1}}{\partial u^1} = \frac{\partial \overline{u}_K^n}{\partial u^1} - \frac{\alpha^{i,j}}{2h_{i,j}}(2\beta_0 + 8\beta_1 + 3).$$

With $\beta_0 \geq \frac{3}{2} - 4\beta_1$, $\frac{1}{8} \leq \beta_1 \leq \frac{1}{4}$ satisfied in the numerical flux (2.3), we have $\overline{u}_K^{n+1}$ as a monotone increasing function on the solution values u_o^1, u_o^2, u_o^3 chosen from element K_3. Similar discussion applies to the solution values used in element K_1 and K_2. With 2-point Gaussian quadrature rule approximating the line integral, the quantity $\overline{u}_K^{n+1}$ is monotone increasing with respect to the total 18 solution values from K_1, K_2 and K_3.

Similar to Theorem 3.1, we need to have a special quadrature rule for the cell average $\overline{u}_K^n$ that use all the selected points inside K (18 points in total) with positive weights. We use a similar method as the linear case to find such a quadrature rule, see Appendix 5.1. Let w_0 be the minimum weight for the selected points.

With geometry information of the mesh partition,

$$\frac{l_K^i}{h_{i,j}} \leq \frac{1}{C\sin\hat{\theta}}, \qquad C \geq \frac{3 - \sqrt{3}}{6}, \tag{3.13}$$

11

we see condition (3.11) is sufficient to guarantee the coefficients of solution values used in element K being none-negative. Therefore, $\overline{u}_K^{n+1}$ is monotonically increasing w.r.t. all the selected points inside K and its neighbors. Finally we conclude with,

$$m \leq \overline{u}_K^{n+1} \leq M,$$

provided that $m \leq u_K^n(x,y), u_{K_1}^n(x,y), u_{K_2}^n(x,y), u_{K_3}^n(x,y) \leq M$.

$\square$

Implementation of the M-P-S limiter

Given the quadratic DG polynomial solution $u_K^n(x,y)$ with cell average $\overline{u}_K^n \in [m,M]$, the following limiter ensures $\widetilde{u}_K^n(x,y) \in [m,M]$ for any $(x,y) \in K$.

$$\widetilde{u}_K^n(x,y) = \theta(u_K^n(x,y) - \overline{u}_K^n) + \overline{u}_K^n, \qquad \theta = \min\left\{1, \left|\frac{M - \overline{u}_K^n}{M_K - \overline{u}_K^n}\right|, \left|\frac{m - \overline{u}_K^n}{m_K - \overline{u}_K^n}\right|\right\}, \qquad (3.14)$$

with M_K and m_K as the maximum and minimum of $u_K^n(x,y)$ on element K,

$$M_K = \max_{(x,y)\in K} u_K^n(x,y), \qquad m_K = \min_{(x,y)\in K} u_K^n(x,y). \qquad (3.15)$$

Since $u_K^n(x,y)$ is a quadratic polynomial, it is easy to calculate the maximum and minimum value over K.

Notice that the limiter (3.14) does not change the cell average. Moreover, $\|\widetilde{u}_K^n(x,y) - u_K^n(x,y)\|_\infty = \mathcal{O}(h^3)$, if the exact solution is smooth. The proof can be found in [24]. Thus, $\widetilde{u}_K^n(x,y) \in [m,M]$ has uniform third order accuracy for smooth exact solution.

Algorithm 3.1. maximum-principle-satisfying DDG scheme with interface correction

1. *At time level t_n, we apply M-P-S limiter (3.14) - (3.15) to $u_K^n(x,y)$ and obtain $\widetilde{u}_K^n(x,y)$.*

2. *Apply DDG scheme with interface correction (2.3) - (2.4) to $\widetilde{u}_K^n(x,y)$ and evolve in time with SSP Runge-Kutta method [14] to march forward the solution to the next time level t_{n+1}.*

For the convection part of (1.1), as discussed in [24], the solution average at next time level $\overline{u}_K^{n+1}$ is a monotone function with respect to certain solution values (Gauss-Lobatoo points) at time level t_n. Thus the M-P-S limiter (3.14) - (3.15) with M_K and m_K as the maximum and minimum of the polynomial solution $u_K^n(x,y)$ over the whole element K is enough to guarantee the solution average staying in the given bound.

Remark 3.1. *For general convection diffusion equation (1.1), we apply same procedure listed in Algorithm 3.1 to guarantee the quadratic polynomial solution stay in the given bound and at the same time maintain the 3rd order accuracy. For (1.1), we apply DDG with interface correction scheme (2.5) for spatial discretization.*

4   Numerical Examples

In this section, we present a sequence of examples to demonstrate the performance of M-P-S limiter. For all examples in this section, we have the coefficient pair taken as $(\beta_0, \beta_1) = (5, \frac{1}{8})$ in the numerical flux formula (3.2).

Example 4.1. *Accuracy test on linear diffusion equation*

We start with accuracy check of the DDG with interface correction (3.1) with and without M-P-S limiter (3.14) applied on the following linear diffusion equation,

$$u_t - \epsilon \Delta u = 0, \qquad (x, y) \in [0, 1] \times [0, 1], \qquad t \in (0, T),$$

with initial data $u(x, y, 0) = u_0(x, y) = \sin(2\pi(x + y))$ and periodic boundary condition. The exact solution is given with,

$$u(x, y, t) = \exp(-8\pi^2 \epsilon t) \sin(2\pi(x + y)).$$

Here, we take $\epsilon = 1$ and final time $t = 0.0001$. We implement the scheme with P^2 quadratic polynomials on unstructured mesh in Figure 3(a) and on mesh with obtuse triangles with largest angle about $\frac{3}{5}\pi$ in Figure 3(b). Third order of accuracy is maintained with and without M-P-S limiter (3.14) applied, see Table 1 and Table 2. At each time step t_n, we set the bounds to be $ue_{\min} = -\exp(-8\pi^2 \epsilon t_n)$ and $ue_{\max} = \exp(-8\pi^2 \epsilon t_n)$, which are the minimum and maximum of the exact solution. We use $u_{\min}$ and $u_{\max}$ to denote the DG solution minimum and maximum values. The overshoots and undershoots are eliminated after the M-P-S limiter applied, see Table 1 and Table 2.

(a) Triangular mesh with $h = 0.117$.   (b) Mesh with obtuse triangles, $h = 0.148$.

Figure 3: Illustration of meshes

Example 4.2. *Porous medium equation*

In this example we consider the nonlinear porous medium equation

$$u_t = \left(u^2\right)_{xx} + \left(u^2\right)_{yy}, \quad (x, y) \in [-10, 10] \times [-10, 10],$$

with initial condition

$$u(x, y, 0) = \begin{cases} 1, & (x - 2)^2 + (y + 2)^2 < 6, \\ 1, & (x + 2)^2 + (y - 2)^2 < 6, \\ 0, & otherwise, \end{cases}$$

13

h	without M-P-S limiter						with M-P-S limiter					
	L^2 error	Order	L^∞ error	Order	$u_{\min} - ue_{\min}$	$u_{\max} - ue_{\max}$	L^2 error	Order	L^∞ error	Order	$u_{\min} - ue_{\min}$	$u_{\max} - ue_{\max}$
0.117	1.21e-03		1.10e-02		-3.91e-03	3.16e-03	1.28e-03		1.10e-02		0	0
0.0587	1.94e-04	2.64	1.28e-03	3.10	-2.40e-04	2.49e-04	1.95e-04	2.71	1.28e-03	3.10	0	0
0.0293	2.60e-05	2.90	1.59e-04	3.01	-1.49e-05	1.50e-05	2.60e-05	2.91	1.59e-04	3.01	0	0
0.0147	3.27e-06	2.99	2.01e-05	2.98	-9.61e-07	9.41e-07	3.27e-06	2.99	2.01e-05	2.98	0	0
0.00733	4.11e-07	2.99	2.52e-06	2.99	-5.82e-08	5.96e-08	4.11e-07	2.99	2.52e-06	2.99	0	0

Table 1: Accuracy table on triangular mesh of Figure 3(a).

h	without M-P-S limiter						with M-P-S limiter					
	L^2 error	Order	L^∞ error	Order	$u_{\min} - ue_{\min}$	$u_{\max} - ue_{\max}$	L^2 error	Order	L^∞ error	Order	$u_{\min} - ue_{\min}$	$u_{\max} - ue_{\max}$
0.148	8.23e-04		1.46e-02		-1.17e-02	9.70e-03	1.08e-03		1.46e-02		0	0
0.0741	1.23e-04	2.74	1.82e-03	3.00	-7.86e-04	5.49e-05	1.26e-04	3.11	2.02e-03	2.85	0	0
0.0371	1.68e-05	2.87	2.17e-04	3.06	-2.66e-05	4.24e-05	1.68e-05	2.90	2.17e-04	3.22	0	0
0.0185	2.14e-06	2.97	2.69e-05	3.02	-1.12e-06	9.22e-07	2.14e-06	2.97	2.69e-05	3.02	0	0
0.00927	2.70e-07	2.99	3.33e-06	3.01	-7.17e-08	1.33e-08	2.70e-07	2.99	3.33e-06	3.01	0	0

Table 2: Accuracy table on unstructured mesh with obtuse triangles of Figure 3(b).

and zero boundary condition. Piecewise quadratic polynomial solutions implemented on unstructured mesh (Figure 3(a)) with size $h = 0.00733$ are shown in Figure 4. Notice that the minimum of the solution is zero. Numerical approximation without M-P-S limiter may become negative which may lead the problem ill-posed and cause the computations blow up. Implementations on a coarser mesh with $h = 0.0587$ are carried out, see Figure 5. With M-P-S limiter applied, DDG interface correction solutions are maintained strictly inside the bound $[0, 1]$.

Example 4.3. *Strongly degenerate parabolic problem*

We consider the following strongly degenerate parabolic problem with DDG interface correction method (2.5),

$$u_t + (u^2)_x + (u^2)_y = \epsilon(\nu(u)\nabla u_x)_x + \epsilon(\nu(u)\nabla u_y)_y, \quad (x, y) \in [-1.5, 1.5] \times [-1.5, 1.5].$$

Initial condition is given with,

$$u(x, y, 0) = \begin{cases} 1, & (x + 0.5)^2 + (y + 0.5)^2 < 0.16, \\ -1, & (x - 0.5)^2 + (y - 0.5)^2 < 0.16, \\ 0, & otherwise, \end{cases}$$

and zero boundary condition is applied. We have $\epsilon = 0.1$ and,

$$\nu(u) = \begin{cases} 0, & |u| \leq 0.25, \\ 1, & |u| > 0.25. \end{cases}$$

Quadratic polynomial implementation with M-P-S limiter is carried out. Here we apply the simple slope limiter [3] to compress the oscillations caused from the nonlinear convection term. For the slope limiter, we take $\gamma = 1.5$ and $M = 5$. Implementation on mesh (Figure 3(a)) with $h = 0.0147$ is shown in Figure 6. The result agrees well those in literature, see [12, 21].

Example 4.4. *Incompressible Navier-Stokes equation in vorticity stream-function formulation*

In this example, we consider to solve two-dimensional incompressible Navier-Stokes equation,

$$\begin{cases} w_t + (uw)_x + (vw)_y = \frac{1}{Re}\nabla w, \\ \Delta\phi = w, \quad \langle u, v \rangle = \langle -\phi_y, \phi_x \rangle, \\ \langle u, v \rangle \cdot \mathbf{n} = given, \quad (x, y) \in \partial\Omega, \end{cases} \tag{4.1}$$

14

(a) t = 0

(b) t = 0.1

(c) t = 0.5

(d) t = 2

Figure 4: Nonlinear porous medium problem with $h = 0.00733$.

Figure 5: The cut of the DDG interface correction solution along line $x + y = 0$ at $t = 0.005$. Red circle symbol: no M-P-S limiter. Blue diamond symbol: M-P-S limiter applied.

Figure 6: Strongly degenerate parabolic problem solution at $t = 0.5$.

written out in the vorticity stream-function format. We focus on the incompressible flow with high Reynolds numbers ($Re \gg 1$), thus explicit treatment on both convection term and diffusion term is efficient.

The initial vorticity is given with $w(x, y, 0) = w_0(x, y)$ and periodic boundary condition is applied. We have $\phi(x, y)$ denoting the stream function and the velocity field is denoted as $\langle u, v \rangle$. We adopt the method of [10] by Liu and Shu to solve (4.1). Thus at each time step, we first apply P^2 continuous finite element method as the Poisson solver to obtain stream function ϕ, then have the velocity field and plug them into the vorticity equation and discretize in space with DDG interface correction method (2.5), finally update the vorticity DG solution to the next time level. High order SSP Runge-Kutta explicit scheme [14] is applied to march forward solution in time. As remarked in [10] that there is a natural match between the vorticity DG solution and the stream function. The normal component of velocity field $\langle u, v \rangle \cdot \mathbf{n}$ is continuous across all triangle edges, thus DG implementation on the convection part of vorticity equation is straight forward.

We carry out two tests in this example. First one is for accuracy check with exact solution maximum and minimum available and being applied with M-P-S limiter at each time step. The second one is a vortex patch problem.

Accuracy Test

We solve (4.1) with $Re = 100$. Initial condition is $w_0(x, y) = -2 \sin(x) \sin(y)$ with $\Omega = [0, 2\pi] \times [0, 2\pi]$ and periodic boundary condition is applied. Exact solution is available with

$$w(x, y, t) = -2 \sin(x) \sin(y) \exp\left(-2t/Re\right).$$

Quadratic P^2 implementations are carried out on mesh Figure 3(a) and on mesh Figure 3(b) with obtuse triangles. Errors and orders are listed in Table 3 and Table 4. Again we have we_{max} and we_{min} represent exact solution maximum and minimum values. We observe that the M-P-S limiter removes all overshoots and undershoots and still maintains the third order accuracy.

$h/2\pi$	without limiter						with limiter					
	L^2 error	Order	L^∞ error	Order	$w_{\min} - we_{\min}$	$w_{\max} - we_{\max}$	L^2 error	Order	L^∞ error	Order	$w_{\min} - we_{\min}$	$w_{\max} - we_{\max}$
0.117	1.90e-03		1.85e-02		-8.17e-03	7.72e-03	1.98e-03		1.88e-02		0	0
0.0587	2.68e-04	2.82	2.07e-03	3.16	-3.64e-04	6.41e-04	2.68e-04	2.88	2.07e-03	3.18	0	0
0.0293	3.62e-05	2.89	2.92e-04	2.82	-6.40e-05	-4.50e-06	3.62e-05	2.89	2.92e-04	2.82	0	-4.52e-06
0.0147	4.54e-06	3.00	2.56e-05	3.51	-1.29e-06	3.62e-06	4.54e-06	3.00	2.56e-05	3.51	0	0
0.00733	5.84e-07	2.96	4.05e-06	2.66	-1.53e-07	1.95e-07	5.84e-07	2.96	4.05e-06	2.66	0	0

Table 3: Accuracy check on mesh Figure 3(a), final time $t = 0.1$.

$h/2\pi$	without limiter						with limiter					
	L^2 error	Order	L^∞ error	Order	$w_{\min} - we_{\min}$	$w_{\max} - we_{\max}$	L^2 error	Order	L^∞ error	Order	$w_{\min} - we_{\min}$	$w_{\max} - we_{\max}$
0.148	1.31e-03		2.74e-02		-8.40e-04	-7.18e-05	1.31e-03		2.74e-02		0	-7.18e-05
0.0741	1.72e-04	2.93	3.35e-03	3.03	3.56e-05	-6.40e-05	1.72e-04	2.93	3.35e-03	3.03	4.15e-05	-6.40e-05
0.0371	2.37e-05	2.86	4.47e-04	2.90	3.36e-06	1.71e-06	2.37e-05	2.86	4.47e-04	2.90	3.36e-06	0
0.0185	2.99e-06	2.99	5.33e-05	3.07	1.79e-07	1.43e-07	2.99e-06	2.99	5.33e-05	3.07	1.79e-07	0

Table 4: Accuracy check on mesh Figure 3(b) with obtuse triangles, final time $t = 0.1$.

Vortex patch problem

$Re = 100$	without limiter		with limiter	
$h/2\pi$	$w_{\min} - we_{\min}$	$w_{\max} - we_{\max}$	$w_{\min} - we_{\min}$	$w_{\max} - we_{\max}$
0.117	-6.96e-01	6.87e-01	0	0
0.0587	-2.72e-01	2.66e-01	0	0
0.0293	-8.02e-02	4.22e-02	0	0
0.0147	-3.17e-03	3.00e-03	0	0
0.00733	-6.41e-04	7.41e-04	0	0

Table 5: Maximum and minimum of the solutions, $Re = 100$ at $t = 0.1$.

$Re = 10000$	without limiter		with limiter	
$h/2\pi$	$w_{\min} - we_{\min}$	$w_{\max} - we_{\max}$	$w_{\min} - we_{\min}$	$w_{\max} - we_{\max}$
0.117	-8.62e-01	8.62e-01	0	0
0.0587	-6.14e-01	7.82e-01	0	0
0.0293	-5.14e-01	5.77e-01	0	0
0.0147	-4.35e-01	4.79e-01	0	0
0.00733	-3.52e-01	2.81e-01	0	0

Table 6: Maximum and minimum of the solutions, $Re = 10000$ at $t = 0.1$.

Now we consider problem (4.1) with initial condition,

$$w_0(x,y) = \begin{cases} -1, & (x,y) \in [\frac{\pi}{2}, \frac{3\pi}{2}] \times [\frac{\pi}{4}, \frac{3\pi}{4}], \\ 1, & (x,y) \in [\frac{\pi}{2}, \frac{3\pi}{2}] \times [\frac{5\pi}{4}, \frac{7\pi}{4}], \\ 0, & otherwise, \end{cases} \tag{4.2}$$

and periodic boundary condition. The Reynolds number is chosen to be $Re = 100$ or $Re = 10000$. We compare the maximum and minimum of the numerical solutions with and without M-P-S limiter applied, see Table 5 and Table 6. Mesh Figure 3(a) is used and quadratic polynomials is applied. We also plot the solutions for the case $Re = 100$ at $t = 1$, shown in Figure 7, and the case $Re = 10000$ at $t = 5$, shown in Figure 8. It is clear that the overshoots and undershoots are removed with the M-P-S limiter applied.

5 Appendix

As discussed in Theorem 3.1, we need to find a 13-point quadrature rule that is exact for P^2 polynomial and include all the selected points (total 12) as quadrature points. For example we consider the edge AB shared by K and K_3. As shown in Figure 1, five points $\mathbf{x}_{K,i}$ $(i = 1, \cdots, 5)$ from element K's side are used in the calculation of (3.6) - (3.7). In this section, our goal is to find such a quadrature rule with the selected points as quadrature points and having non-negative weights. Specifically we need to find the minimum weight in the quadrature rule, since it is used in (3.4) - (3.5) to bound CFL condition and further identify suitable time step size for time evolution. For convenience, we introduce notation $|K|$ to represent the area of triangle element K.

5.1 Quadrature rules for cell average

Again we use edge AB to illustrate the way to find such quadrature rule with $\mathbf{x}_{K,i}$ $(i = 1, \cdots, 5)$ as quadrature points. The method we investigate is that we look for weights ω_i in the following

Figure 7: Contours of the solutions, $Re = 100$, at $t = 1$ with mesh size $h = 0.0293 \times 2\pi$. Top: no M-P-S limiter ($w_{\min} = -1.0011$, $w_{\max} = 1.0008$); Bottom: add M-P-S limiter ($w_{\min} = -1$, $w_{\max} = 1$). 30 equally spaced contour lines are plotted.

Figure 8: Contours of the solutions, $Re = 10000$ at $t = 5$ and mesh size $h = 0.0293 \times 2\pi$. Top: no M-P-S limiter ($w_{\min} = -1.1211$, $w_{\max} = 1.1508$); Bottom: add M-P-S limiter ($w_{\min} = -1$, $w_{\max} = 1$). 30 equally spaced contour lines are plotted.

Point	Case 1	Case 2
$\mathbf{x}_{K,4}$	$\frac{1}{6}\cdot\left(\frac{2A_1}{|K|}\cdot\frac{1}{3}\right)\geq\frac{1}{18}\tan(\hat\theta)\cot(\check\theta)$	$\frac{1}{6}\cdot\left(\frac{2A_1}{|K|}\cdot\frac{1}{3}\right)\geq\frac{1}{18}\tan(\hat\theta)\cot(\check\theta)$
$\mathbf{x}_{K,1}$	$\frac{1}{6}\cdot\left(\frac{A_1+A_2}{|K|}w_1\right)\geq\frac{w_1}{12}\tan(\hat\theta)\cot(\check\theta)$	$\frac{1}{6}\cdot\left(\frac{A_1+A_2}{|K|}w_1\right)\geq\frac{w_1}{12}\tan(\hat\theta)\cot(\check\theta)$
$\mathbf{x}_{K,2}$	$\frac{1}{6}\cdot\left(\frac{A_1+A_3}{|K|}w_1\right)\geq\frac{w_1}{12}\tan(\hat\theta)\cot(\check\theta)$	$\frac{1}{6}\cdot\left(\frac{A_1}{|K|}w_1\right)\geq\frac{w_1}{12}\tan(\hat\theta)\cot(\check\theta)$
$\mathbf{x}_{K,3}$	$\frac{1}{6}\cdot\left(\frac{2A_1}{|K|}w_1\right)\geq\frac{w_1}{6}\tan(\hat\theta)\cot(\check\theta)$	$\frac{1}{6}\cdot\left(\frac{2A_1}{|K|}w_1\right)\geq\frac{w_1}{6}\tan(\hat\theta)\cot(\check\theta)$
$\mathbf{x}_{K,5}$	$\frac{1}{6}\cdot\left(\frac{2A_1+A_2+A_3}{|K|}w_1\right)=\frac{w_1}{6}$	$\frac{1}{6}\cdot\left(\frac{2A_1+A_2}{|K|}w_1\right)=\frac{w_1}{6}$

Table 7: Estimate of the quadrature weights

format,

$$\frac{1}{3}\overline{u}_K^n = \frac{1}{3|K|}\int_K u_K^n(x,y)dxdy = \sum_{i=1}^{5} w_i u_K^n(\mathbf{x}_{K,i}) + \sum_{j=1}^{l} w_j^* u_K^n(\mathbf{x}_{K,j}^*),$$

with other quadrature points $\mathbf{x}_{K,j}^*$ $(j=1,\cdots,l)$ selected in the following to complete the quadrature rule. Notice we will combine these points together with other points from edges BC and CA to find the location and weight for the 13-th point.

(a) Angles in K and K_3 (b) Case 1   (c) Case 2

Figure 9: Left: Element K with its neighbor K_3; Right: Two cases of points selected along normal vector($A1$, $A2$, $A3$ denote small triangles area).

The subsection 5.2 below offers one way to find a quadrature rule on triangle element with vortices included as quadrature points (with weight ω_1 in (5.4)). We also use a quadrature rule for triangle element only with edge centers with equal weights 1. We divide triangle K into small triangles with $\mathbf{x}_{K,i}$ $(i=1,\cdots,5)$ being vertices or edge centers and use each rule for a half $\frac{1}{6}\overline{u}_K^n$. In this way, all other vertices and edge centers are selected as quadrature points as well.

Let's focus the discussion on point $\mathbf{x}_{K,4}$ and $\mathbf{x}_{K,5}$, shown in Figure 1 or the blue dots in Figure 9. First of all, the location of $\mathbf{x}_{K,4}$ and $\mathbf{x}_{K,5}$ are determined by elements K and K_3. There are two cases: (1) $\min(\theta_1,\theta_2) > \min(\theta_4,\theta_5)$, then $\mathbf{x}_{K,5}$ is inside K, see Figure 9(b); (2) $\min(\theta_1,\theta_2) \leq \min(\theta_4,\theta_5)$, then $\mathbf{x}_{K,5}$ is on the edge of K, see Figure 9(c). The triangle geometrical information is shown and marked in Figure 9. We can estimate the quadrature weights of $\mathbf{x}_{K,i}$ $(i=1,\cdots,5)$ in each case, see Table 7.

We apply the same procedure to edge BC and CA to include selected points as quadrature points. Collect all data from the three edges, we have the estimate on the weights as follows,

$$\frac{\partial \overline{u}_K^n}{\partial u_{K,1}}, \frac{\partial \overline{u}_K^n}{\partial u_{K,2}}, \frac{\partial \overline{u}_K^n}{\partial u_{K,3}} \geq \frac{w_1}{6} \tan(\hat{\theta}) \cot(\breve{\theta}), \qquad \frac{\partial \overline{u}_K^n}{\partial u_{K,4}} \geq \frac{1}{18} \tan(\hat{\theta}) \cot(\breve{\theta}), \qquad \frac{\partial \overline{u}_K^n}{\partial u_{K,5}} = \frac{w_1}{6}. \quad (5.1)$$

Now we are ready to find a non-negative quadrature rule with only 13 points inside triangle K which include the 12 selected points from numerical flux integral on the edges. Let's reorder the quadrature points, and denote $u_{K,i}$ $(i = 1, \cdots, 12)$ as the 12 points selected from evaluating $\int_{\partial K} \widehat{u_{\mathbf{n}}} ds$ with weights w_i $(i = 1, \cdots, 12)$. We also reorder the rest points and denote them as $u_{K,j}^*$ $(j = 1, \cdots, m,$ where m is an integer), with weights w_j^* $(j = 1, \cdots, m)$. Let $w_{13} = \sum_{j=1}^m w_j^*$, we have $\sum_{j=1}^{13} w_j = \sum_{j=1}^{12} w_j + \sum_{j=1}^m w_j^* = 1$. Moreover, $\frac{1}{w_{13}} \sum_{j=1}^m w_j^* u_{K,j}^* = \sum_{j=1}^m \frac{w_j^*}{w_{13}} u_{K,j}^*$ is a convex combination of the points $u_{K,j}^*$ $(j = 1, \cdots, m)$. By the mean value theory, one can find a point $u_{K,13}$ inside the convex hall of the points $u_{K,j}^*, j = 1, \cdots, m$ such that $u_{K,13} = \sum_{j=1}^m \frac{w_j^*}{w_{13}} u_{K,j}^*$. Therefore, we have a quadrature rule with positive weights of total 13 points inside triangle K which include the 12 selected points: $\{(u_{K,j}, w_j) | j = 1, \cdots, 12\}$.

5.2 Quadrature rule for triangle element with vertices

In this section, we design one quadrature rule that is exact for quadratic polynomial $P^2(K)$ on any triangle element K. Especially we include the three vortices and three edge centers in the quadrature points set. This work is inspired by the quadrature rule designed in [26]. Specifically we are interested in finding the weights before the vertices.

For convenience, we use the position vectors to denote the three vertices of K: $\mathbf{v}^1$, $\mathbf{v}^2$ and $\mathbf{v}^3$. Thus, the position vector $\mathbf{P}$ of any point P inside triangle K can be described by the barycentric coordinates (ξ_1, ξ_2, ξ_3), i.e., $\mathbf{P} = \xi_1 \mathbf{v}^1 + \xi_2 \mathbf{v}^2 + \xi_3 \mathbf{v}^3$.

We first consider the quadrature rule on the unit square with vertices coordinates as $\left(-\frac{1}{2}, -\frac{1}{2}\right)$, $\left(-\frac{1}{2}, \frac{1}{2}\right)$, $\left(\frac{1}{2}, -\frac{1}{2}\right)$ and $\left(\frac{1}{2}, \frac{1}{2}\right)$ in the u-v plane, and then we use projections/transformations to map it to a quadrature rule on triangle element K.

Let $\{\widehat{u}^\alpha : \alpha = 1, 2, 3\}$ denote the Gauss-Lobatto quadrature points on $\left[-\frac{1}{2}, \frac{1}{2}\right]$ with weights $\widehat{w}_\alpha$ (in Table 8), which is exact for one variable polynomial of degree 3. We have $\{v^\beta : \beta = 1, 2, 3\}$ (including the left boundary $v^1 = -\frac{1}{2}$) denote the 3-point Gauss-Radau quadrature points on $\left[-\frac{1}{2}, \frac{1}{2}\right]$ with weights w_β (in Table 9), which is exact for one variable polynomial of degree 4. For a two-variable polynomial $p(u, v)$, we use tensor product of 3-point Gauss-Lobatto for u and 3-point Gauss Radau for v as the quadrature rule on the square. The quadrature points can be written as $S^2 = \left\{\left(\widehat{u}^\alpha, v^\beta\right) : \alpha = 1, 2, 3; \beta = 1, 2, 3\right\}$ with weights $\widehat{w}_\alpha w_\beta$, listed in Figure 10(a).

α	1	2	3
$\widehat{u}^\alpha$	$-\frac{1}{2}$	0	$\frac{1}{2}$
$\widehat{w}_\alpha$	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

Table 8: 3-point Gauss-Lobatto quadrature rule

β	1	2	3
v^β	$-\frac{1}{2}$	$\frac{1}{10}(1 - \sqrt{6})$	$\frac{1}{10}(1 + \sqrt{6})$
w_β	$\frac{1}{9}$	$\frac{1}{36}(16 + \sqrt{6})$	$\frac{1}{36}(16 - \sqrt{6})$

Table 9: 3-point Gauss-Radau quadrature rule

Without loss of generality, we assume the orientation of the three vertices $\mathbf{v}^1$, $\mathbf{v}^2$ and $\mathbf{v}^3$ is marked clockwise. We define the following three functions as projections from the square to triangle K, mapping the top edge of the square into one vertex and the other three edges to the edges of

K.

$$\mathbf{g}_1(u,v) = \left(\frac{1}{2}+v\right)\mathbf{v}^1 + \left(\frac{1}{2}+u\right)\left(\frac{1}{2}-v\right)\mathbf{v}^2 + \left(\frac{1}{2}-u\right)\left(\frac{1}{2}-v\right)\mathbf{v}^3,$$

$$\mathbf{g}_2(u,v) = \left(\frac{1}{2}+v\right)\mathbf{v}^2 + \left(\frac{1}{2}+u\right)\left(\frac{1}{2}-v\right)\mathbf{v}^3 + \left(\frac{1}{2}-u\right)\left(\frac{1}{2}-v\right)\mathbf{v}^1,$$

$$\mathbf{g}_3(u,v) = \left(\frac{1}{2}+v\right)\mathbf{v}^3 + \left(\frac{1}{2}+u\right)\left(\frac{1}{2}-v\right)\mathbf{v}^1 + \left(\frac{1}{2}-u\right)\left(\frac{1}{2}-v\right)\mathbf{v}^2.$$

Under each projection $\mathbf{g}_i$ ($i = 1, 2$ or 3), the quadrature points S^2 are mapped onto the triangle K, i.e. $\mathbf{g}_i(S^2)$, as in Figure 10(b) - Figure 10(d) . Let $S_K^2 = \mathbf{g}_1(S^2) \cup \mathbf{g}_2(S^2) \cup \mathbf{g}_3(S^2)$.

(a) S^2 (b) $\mathbf{g}_1(S^2)$ (c) $\mathbf{g}_2(S^2)$ (d) $\mathbf{g}_3(S^2)$

Figure 10: Illustration of quadrature points mapping from rectangle element to triangle element.

We use $\mathbf{g}_i(i = 1, 2, 3)$ and S^2 to construct our triangle element quadrature rule. Let $p_K(x, y)$ be a two-variable polynomial of degree 2 with cell average $\overline{p_K} = \frac{1}{|K|}\int_K p_K(x,y)dxdy$, then we have,

$$\overline{p_K} = \frac{1}{|K|}\int_K p_K(x,y)\,dA = \frac{1}{|K|}\int_{-\frac{1}{2}}^{\frac{1}{2}}\int_{-\frac{1}{2}}^{\frac{1}{2}} p_K(\mathbf{g}_i(u,v))\left|\frac{\partial \mathbf{g}_i(u,v)}{\partial(u,v)}\right|\,dudv, \quad i = 1,2,3$$

$$= \int_{-\frac{1}{2}}^{\frac{1}{2}}\int_{-\frac{1}{2}}^{\frac{1}{2}} p_K(\mathbf{g}_i(u,v))2\left(\frac{1}{2}-v\right)\,dudv = \sum_{\alpha=1}^{3}\sum_{\beta=1}^{3} p_K(\mathbf{g}_i(\widehat{u}^\alpha, v^\beta))2\left(\frac{1}{2}-v^\beta\right)\widehat{w}_\alpha w_\beta$$

$$= \sum_{i=1}^{3}\sum_{\alpha=1}^{3}\sum_{\beta=1}^{3} p_K(\mathbf{g}_i(\widehat{u}^\alpha, v^\beta))\frac{2}{3}\left(\frac{1}{2}-v^\beta\right)\widehat{w}_\alpha w_\beta = \sum_{\mathbf{x}\in S_K^2} p_K(\mathbf{x})w_{\mathbf{x}}. \tag{5.2}$$

With the three vertices $\mathbf{v}^1$, $\mathbf{v}^2$ and $\mathbf{v}^3$ orientated clockwise, we have the Jacobian $\left|\frac{\partial \mathbf{g}_i(u,v)}{\partial(u,v)}\right| = 2|K|(\frac{1}{2}-v)$. Notice that $p_K(\mathbf{g}_i(u,v))2(\frac{1}{2}-v)$ is a polynomial of u and v with degree 2 and degree 3, therefore the quadrature rule on S^2 is exact.

It is easy to show the weights $w_{\mathbf{x}}$ for quadrature points $\mathbf{x} \in S_K^2$ are non-negative, then we can rewrite (5.2) as a combination of quadrature points, see below,

$$\overline{p_K} = \sum_{\mathbf{x}\in S_K^2\setminus\{\mathbf{v}^1,\mathbf{v}^2,\mathbf{v}^3\}} p_K(\mathbf{x})w_{\mathbf{x}} + \sum_{i=1}^{3} p_K(\mathbf{v}^i)\overline{w}_i. \tag{5.3}$$

We are interested in the weights $\{\overline{w}_i\}_{i=1}^3$ for all three vertices $\mathbf{v}^1$, $\mathbf{v}^2$ and $\mathbf{v}^3$. Let us take $\mathbf{v}^1$ for example. Notice that $\mathbf{g}_2(-\frac{1}{2}, -\frac{1}{2})$ and $\mathbf{g}_3(\frac{1}{2}, -\frac{1}{2})$ are the same point $(1, 0, 0)$, i.e., $\mathbf{v}^1$. Therefore,

23

the weight of $(1, 0, 0)$ is

$$\overline{w}_1 \quad = \quad \frac{2}{3} \left[\frac{1}{2} - \left(-\frac{1}{2} \right) \right] \widehat{w}_1 w_1 + \frac{2}{3} \left[\frac{1}{2} - \left(-\frac{1}{2} \right) \right] \widehat{w}_3 w_1 = \frac{2}{3} \left(\widehat{w}_1 + \widehat{w}_3 \right) w_1 = \frac{2}{81}. \qquad (5.4)$$

Remark 5.1. *This section only provide one way to construct a quadrature rule on any triangle, with three vertices and edge centers included as quadrature points. The goal is to show that one can find such a quadrature rule with positive weights for quadrature points.*

References

[1] D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.*, 19(4):742–760, 1982.

[2] E. Bertolazzi and G. Manzini. A second-order maximum principle preserving finite volume method for steady convection-diffusion problems. *SIAM J. Numer. Anal.*, 43(5):2172–2199 (electronic), 2005.

[3] B. Cockburn, C. Johnson, C.-W. Shu, and E. Tadmor. *Advanced numerical approximation of nonlinear hyperbolic equations*, volume 1697 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1998. Papers from the C.I.M.E. Summer School held in Cetraro, June 23–28, 1997, Edited by Alfio Quarteroni, Fondazione C.I.M.E.. [C.I.M.E. Foundation].

[4] B. Cockburn and C.-W. Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal.*, 35(6):2440–2463 (electronic), 1998.

[5] S. Evje and K. H. Karlsen. Monotone difference approximations of BV solutions to degenerate convection-diffusion equations. *SIAM J. Numer. Anal.*, 37(6):1838–1860 (electronic), 2000.

[6] F. Gao, Y. Yuan, and D. Yang. An upwind finite-volume element scheme and its maximum-principle-preserving property for nonlinear convection-diffusion problem. *Internat. J. Numer. Methods Fluids*, 56(12):2301–2320, 2008.

[7] S. Holst. An a priori error estimate for a monotone mixed finite-element discretization of a convectiondiffusion problem. *Numerische Mathematik*, 109:101–119, 2008. 10.1007/s00211-007-0097-7.

[8] H. Liu and J. Yan. The direct discontinuous Galerkin (DDG) methods for diffusion problems. *SIAM J. Numer. Anal.*, 47(1):475–698, 2009.

[9] H. Liu and J. Yan. The direct discontinuous Galerkin (DDG) method for diffusion with interface corrections. *Commun. Comput. Phys.*, 8(3):541–564, 2010.

[10] J.-G. Liu and C.-W. Shu. A high-order discontinuous Galerkin method for 2D incompressible flows. *J. Comput. Phys.*, 160(2):577–596, 2000.

[11] X.-D. Liu and S. Osher. Nonoscillatory high order accurate self-similar maximum principle satisfying shock capturing schemes. I. *SIAM J. Numer. Anal.*, 33(2):760–779, 1996.

[12] Y. Liu, C.-W. Shu, and M. Zhang. High order finite difference WENO schemes for nonlinear degenerate parabolic equations. *SIAM J. Sci. Comput.*, 33(2):939–965, 2011.

[13] Z. Sheng and G. Yuan. The finite volume scheme preserving extremum principle for diffusion equations on polygonal meshes. *J. Comput. Phys.*, 230(7):2588–2604, 2011.

[14] C.-W. Shu and S. Osher. Efficient implementation of essentially nonoscillatory shock-capturing schemes. *J. Comput. Phys.*, 77(2):439–471, 1988.

[15] C.-W. Shu and S. Osher. Efficient implementation of essentially nonoscillatory shock-capturing schemes. II. *J. Comput. Phys.*, 83(1):32–78, 1989.

[16] V. Thomée. *Galerkin finite element methods for parabolic problems*, volume 25 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006.

[17] V. Thomée and L. B. Wahlbin. On the existence of maximum principles in parabolic finite element equations. *Math. Comp.*, 77(261):11–19 (electronic), 2008.

[18] T. Vejchodský. On the nonnegativity conservation in semidiscrete parabolic problems. In *Conjugate gradient algorithms and finite element methods*, Sci. Comput., pages 197–210. Springer, Berlin, 2004.

[19] C. Vidden and J. Yan. Direct discontinuous Galerkin method for diffusion problems with symmetric structure. *Journal of Computational Mathematics*, 31(6):638–662, 2013.

[20] Z. Xu. Parametrized maximum principle preserving flux limiters for high order schemes solving hyperbolic conservation laws: one-dimensional scalar problem. *Math. Comp.*, 83(289):2213–2238, 2014.

[21] J. Yan. A new nonsymmetric discontinuous Galerkin method for time dependent convection diffusion equations. *J. Sci. Comput.*, 54(2-3):663–683, 2013.

[22] J. Yan. Maximum principle satisfying 3rd order Direct discontinuous Galerkin methods for convection diffusion equations. *Commun. Comput. Phys.*, 2015, submitted.

[23] P. Yang, T. Xiong, J. Qiu, and Z. Xu. High order maximum principle preserving flux finite volume method for convection dominated problems. *J. Sci. Comput*, under review.

[24] X. Zhang and C.-W. Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. *J. Comput. Phys.*, 229(9):3091–3120, 2010.

[25] X. Zhang and C.-W. Shu. Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments. *Proc. R. Soc. A*, (467):2752–2776., 2011.

[26] X. Zhang, Y. Xia, and C.-W. Shu. Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes. *J. Sci. Comput.*, 50(1):29–62, 2012.

[27] Y. Zhang, X. Zhang, and C.-W. Shu. Maximum-principle-satisfying second order discontinuous galerkin schemes for convection-diffusion equations on triangular meshes. *J. Comput. Phys.*, 234:295–316, 2013.