Journal of Computational Physics 308 (2016) 198–217

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Third order maximum-principle-satisfying direct discontinuous Galerkin methods for time dependent convection diffusion equations on unstructured triangular meshes

Zheng Chen[a], Hongying Huang[b,c], Jue Yan[a,*]

[a] Department of Mathematics, Iowa State University, Ames, IA 50011, United States
[b] School of Mathematics, Physics and Information Science, Zhejiang Ocean University, Zhoushan, Zhejiang, China
[c] Key Laboratory of Oceanographic Big Data Mining & Application of Zhejiang Province, Zhoushan, Zhejiang, China

ARTICLE INFO

Article history:
Received 16 August 2015
Received in revised form 3 December 2015
Accepted 17 December 2015
Available online 21 December 2015

Keywords:
Discontinuous Galerkin methods
Convection diffusion equation
Maximum principle
Positivity preserving
Incompressible Navier–Stokes equations

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 formula, 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.

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 being uniformly 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 and practically cause the computations to blow up. Thus it is attractive to have the numerical solution satisfy discrete maximum principle (or pre-

[*] Corresponding author.
E-mail addresses: zchen@iastate.edu (Z. Chen), huanghy@lsec.cc.ac.cn (H. Huang), jyan@iastate.edu (J. Yan).

http://dx.doi.org/10.1016/j.jcp.2015.12.039
0021-9991/© 2015 Elsevier Inc. All rights reserved.

serve positivity). Solution of equation (1.1) may represent a specific physical meaning and is supposed to be positive, thus negative value approximation loses physical meaning 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] 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]. 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 satisfies discrete maximum principle with third order of accuracy. Discontinuous Galerkin (DG) method is a class of finite element methods 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 [24–26], 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 bounds. For hyperbolic type equations, the solution as a wave propagates with finite speed. Thus the evolution of solution average only relies on the solution integrals or essentially solution values on the element edges. For diffusion type equations, the evolution of solution average depends on *solution derivative* values on the edges, thus the technique developed in [24] cannot be directly applied. Recently in [20,23], Xu dynamically combined first order and third order methods and obtained maximum-principle-satisfying finite difference and finite volume schemes. Also in [27], maximum-principle-satisfying finite volume method on overlapped cells has been obtained.

In [8], we developed the direct DG method (DDG) as a new diffusion DG 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 element K, have integration by parts and formally we obtain,

$$\int\limits_K u_t v \, dxdy - \int\limits_{\partial K} \widehat{u_{\mathbf{n}}} v \, ds + \int\limits_K \nabla u \cdot \nabla v \, dxdy = 0.$$

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

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

It involves the jump $[u]$, the normal derivative average $\overline{u_{\mathbf{n}}}$ and second order normal derivative jump $[u_{\mathbf{nn}}]$ across element edge ∂K to approximate the normal derivative $u_{\mathbf{n}} = \nabla u \cdot \mathbf{n}$ on ∂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 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 triangular meshes. 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,

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

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 DG 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 integral $\int_{\partial K} \widehat{u_{\mathbf{n}}}\ ds$ from selected solution points values in element K and its neighbors. Given suitable choice of coefficient pair (β_0, β_1) in the numerical flux formula, 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. We also design a quadrature rule to calculate solution average on any triangle element such that the selected points are taken as quadrature points and have positive weights on all quadrature points.

Finally we combine 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. Overshoots 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 normal derivative $u_{\mathbf{n}}$ on discontinuous element edges. This gives direct DG methods 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 [28], DG solutions with piecewise linear polynomial approximations are shown satisfying maximum principle on unstructured triangular meshes.

The rest of the article 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. In Appendix A we provide two specific quadrature rules for the solution average.

2. Direct DG method with interface correction

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

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

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 applied 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 $h_K = \mathrm{diam}(K)$, 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,

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

With no ambiguity, for the rest of this article we use same letter u instead of the standard notation u_h to represent DG numerical solution. 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 pointing from K into 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,

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

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

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

where $b_{ij}(u)=\int^{u} a_{ij}(s)ds$ with $a_{ij}(u)$ as the diffusion matrix $A(u)$ entry. The outward normal on ∂K 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,

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

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.

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

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

where $\gamma=A^T(u)\mathbf{n}$ is a vector pointing from K into its neighbor on edge ∂K. This simplification holds true since $(A(u)\nabla u)\cdot\mathbf{n}=\nabla u\cdot(A^T(u)\mathbf{n})$ and we have $\gamma\cdot\mathbf{n}=A^T(u)\mathbf{n}\cdot\mathbf{n}>0$ with the diffusion matrix $A(u)$ being positive definite. We denote $\overline{\boldsymbol{\gamma}}=\gamma/\|\gamma\|$ as the normalized unit vector of γ on ∂K. We have $u_{\overline{\boldsymbol{\gamma}}}=\nabla u\cdot\overline{\boldsymbol{\gamma}}$ denoting the first order direction derivative of u and $u_{\overline{\boldsymbol{\gamma}}\overline{\boldsymbol{\gamma}}}=\nabla u_{\overline{\boldsymbol{\gamma}}}\cdot\overline{\boldsymbol{\gamma}}$ as the second order direction derivative of u along vector $\overline{\boldsymbol{\gamma}}$. Thus the numerical flux (2.3) involves the solution jump, solution first order direction derivative average and solution second order direction derivative jump on ∂K. We should point out the new version of calculating the numerical flux (2.3) is very important to carry out the maximum principle discussion in the following sections. Now the DDGIC [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,

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

associated with (2.3). The last term in (2.4) is the extra added interface correction term. Notice that test function v is taken being 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 edge ∂K.

The complete DDGIC scheme formulation of (1.1) follows,

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

with the convection term Lax–Friedrichs flux defined as,

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

We should point out quadrature rule is needed to approximate the nonlinear volume integrals and line integrals on element edges. For degenerate parabolic case, i.e. $A(u)=0$, we simply skip the contribution from diffusion.

3. Maximum-principle-satisfying DDG methods

In this section we prove DDGIC polynomial solutions of nonlinear diffusion equations (2.1) satisfy discrete maximum principle with third order of accuracy. We first discuss the linear case on unstructured triangular meshes 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 [28] for 2rd order linear approximations.

On the continuous level the maximum principle states that $m\le u(x,y,t)\le M$, 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 $u(x,y,t_n)$ denoting 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 $m\le u_K^n(x,y)\le M$ 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

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

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 DDGIC quadratic polynomial solution of heat equation (1.2) satisfy discrete maximum principle on unstructured triangular meshes. Again we focus on the first step and investigate under what conditions the solution average $m \le \overline{u}_K^{n+1} \le M$ once we have $m \le u_K^n(x,y) \le M$ on all elements at previous time level t_n.

The DDGIC scheme formulation of (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 edge ∂K is given with,

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

Again $\mathbf{n} = (n_1, n_2)$ is the unit outward normal on ∂K. And $\overline{u_{\mathbf{n}}}$ is the solution normal derivative average and $[u_{\mathbf{nn}}]$ is the second order normal derivative jump on edge ∂K. The coefficient pair (β_0, β_1) is chosen to guarantee the convergence of the scheme. For example, following [19] we can choose the parameter pair that satisfies following inequality,

$$\beta_0 > 4\left((\beta_1)^2 \frac{k^2(k^2-1)^2}{3} - \beta_1 \frac{k^2(k^2-1)}{2} + \frac{k^2}{4}\right),$$

to guarantee the scheme convergence. Here k is the polynomial degree. In this article we further investigate the range, in which we should choose the coefficient pair (β_0, β_1), to guarantee the maximum principle satisfied by the numerical solution. Again, we focus on the step of bounding the solution average. 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 defined and denoted with $\overline{u}_K^n = \frac{1}{area(K)} \int_K u_K^n(x,y)\, dxdy$. Here Δt denotes the time step size. On uniform triangular mesh [22], we pick six point values of $u_K^n(x,y)$ inside the triangle and write out the quadratic polynomial solution $u_K^n(x,y)$ in Lagrange format. Then we show the quantity $\overline{u}_K^{n+1}$ monotonically depends on the solution values at previous time level t_n. For arbitrary triangular mesh, it's hard to identify such six points inside each triangle element and obtain the monotonicity property. Now we apply a new idea to calculate the average $\overline{u}_K^{n+1}$.

Checking out the right hand side of (3.3), we see the integral of $\widehat{u_{\mathbf{n}}}$ on the three edges of ∂K contribute the most to the calculation of $\overline{u}_K^{n+1}$. Since $u_K^n(x,y)$ is a quadratic polynomial, we can find a quadrature rule with positive weights and integrate the average $\overline{u}_K^n$ out exactly. Recall that the numerical flux $\widehat{u_{\mathbf{n}}}$ of (3.2) involves the solution jump, normal derivative average and second order normal derivative jump values on ∂K, thus $\widehat{u_{\mathbf{n}}}$ is uniquely determined by the two solution polynomials defined on K and on its neighbor. As shown in Fig. 1 on the left, one triangle element K shares three edges with three neighbor elements. Thus the quantity $\overline{u}_K^{n+1}$ essentially is a function of *four* solution polynomials that spread out in K and in its three neighbors. After carefully calculating the right hand side of (3.3), we show quantity $\overline{u}_K^{n+1}$ can be written out as a convex combination of selected solution point values at previous time level t_n.

Now we consider one edge, for example edge AB shared by K and K_3 (the right one in Fig. 1), to illustrate the points selected to calculate $\int_{AB} \widehat{u_{\mathbf{n}}}\, ds$. Once we show the quantity $\overline{u}_K^{n+1}$ still monotonically depends on the solution values at t_n,

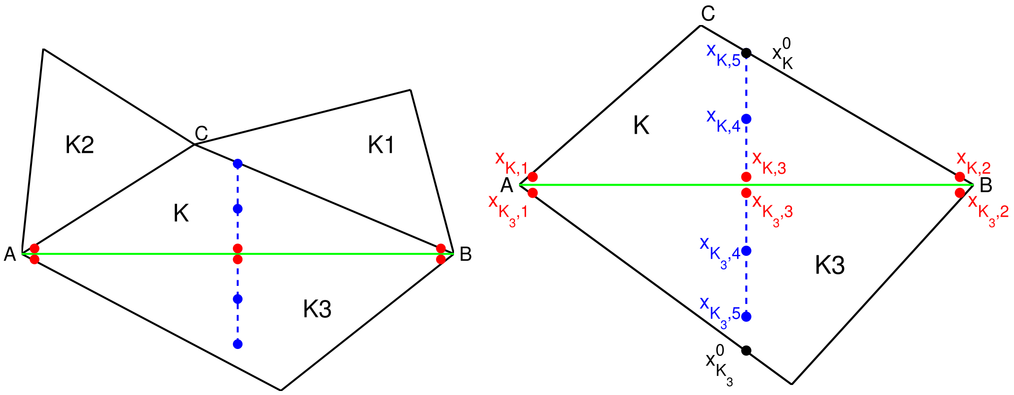


Fig. 1. Left: element K and its three neighbors. Right: selected points to calculate $\int_{AB}\widehat{u_{\mathbf{n}}}\,ds$.

we easily obtain the bound $m \le \overline{u}_K^{n+1} \le M$ given $m \le u_K^n(x,y) \le M$ on all K. From the numerical flux formula (3.2) we have,

$$\int\limits_{AB} \widehat{u_{\mathbf{n}}}\,ds = \frac{\beta_0}{h_{AB}} \int\limits_{AB} (u|_{K_3} - u|_K)\,ds + \int\limits_{AB} \frac{u_{\mathbf{n}}|_{K_3} + u_{\mathbf{n}}|_K}{2}\,ds + \beta_1 h_{AB} \int\limits_{AB} (u_{\mathbf{nn}}|_{K_3} - u_{\mathbf{nn}}|_K)\,ds.$$

Notice that $u_K^n(x,y)$ is a P^2 polynomial. Along the edge AB it degenerates to a one-dimensional quadratic polynomial. Thus three points on AB are enough to calculate $\int_{AB} u|_K\,ds$. Similarly the same three points on AB from the K_3 side are enough to calculate $\int_{AB} u|_{K_3}\,ds$. In Fig. 1, we use different markers to differentiate the points are taken from the side of K or the side of K_3. Essentially $\mathbf{x}_{K,1}$ and $\mathbf{x}_{K_3,1}$ are nothing but vertex A, $\mathbf{x}_{K,2}$ and $\mathbf{x}_{K_3,2}$ are vertex B, and $\mathbf{x}_{K,3}$ and $\mathbf{x}_{K_3,3}$ are both the middle point of edge AB.

The second normal derivative $u_{\mathbf{nn}}$ degenerates to a constant on either side of AB, thus three point values of $u_K^n(x,y)$ along the normal line, namely $\mathbf{x}_{K,3}$, $\mathbf{x}_{K,4}$ and $\mathbf{x}_{K,5}$, are sufficient to calculate the constant $u_{\mathbf{nn}}|_K$. Similarly the three values of $u_{K_3}^n(x,y)$ on $\mathbf{x}_{K_3,3}$, $\mathbf{x}_{K_3,4}$ and $\mathbf{x}_{K_3,5}$ are enough to calculate constant $u_{\mathbf{nn}}|_{K_3}$. Furthermore we see the first normal derivative $u_{\mathbf{n}}$ restricted on AB is a one-dimensional linear polynomial, thus the point value of $u_{\mathbf{n}}|_K$ at edge center, namely $\mathbf{x}_{K,3}$, is sufficient to calculate $\int_{AB} u_{\mathbf{n}}|_K\,ds$. In a word, the same three point values of $u_K^n(x,y)$ on $\mathbf{x}_{K,3}$, $\mathbf{x}_{K,4}$ and $\mathbf{x}_{K,5}$, which are used to calculate $u_{\mathbf{n}}|_K$ at point $\mathbf{x}_{K,3}$, are sufficient to compute quantity $\int_{AB} u_{\mathbf{n}}|_K\,ds$. Similar situation is applied to the side of K_3. We refer to Fig. 1 for all selected points. With this new idea to calculate the numerical flux $\widehat{u_{\mathbf{n}}}$ integral on element edge, 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 polynomial approximations on unstructured triangular mesh $\mathcal{T}_h$. Given $u_K^n(x,y)$ in the range of $[m,M]$ on all K, we have $\overline{u}_K^{n+1} \in [m,M]$ provided,*

$$\beta_0 \ge \frac{9}{4} - 6\beta_1, \quad \frac{1}{8} \le \beta_1 \le \frac{1}{4}, \quad \lambda = \frac{\Delta t}{area(K)} \le C(\beta_0, \beta_1, \mathcal{T}_h). \tag{3.4}$$

Here (β_0, β_1) is the coefficient pair in the numerical flux (3.2). We have $\hat{\theta}$ denoting the minimum angle of the mesh $\mathcal{T}_h$, and C is a constant determined by β_0, β_1 and mesh geometric information, i.e.,

$$C = \bar{w}_1 \tan(\hat{\theta}) \cdot \min\left\{ \frac{\theta_0}{2\beta_0}, \frac{\theta_0}{4(\beta_0 + 6\beta_1 - \frac{9}{4})}, \frac{\theta_0}{3(1-4\beta_1)}, \frac{1}{3(8\beta_1 - 1)} \right\}, \tag{3.5}$$

where $\bar{w}_1 = \frac{1}{27}$ is the minimum weight (A.2) of the specific quadrature rule constructed, see Appendix A.1, and θ_0 is a constant determined by the mesh $\mathcal{T}_h$, as in (A.5) and (A.7).

Proof. To bound $\overline{u}_K^{n+1}$ of (3.3), we know 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 Fig. 1 to illustrate the definition of h_K chosen in the numerical flux formula (3.2). Still denote $\mathbf{x}_{K,3}$ as the middle point of edge AB. 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,

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

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 Fig. 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 values $u_{K,1}, \ldots, u_{K,5}$ are enough to calculate $\int_{AB} \widehat{u_{\mathbf{n}}} ds$ from the side of element K. Similarly, the five points values $u_{K_3,1}, \ldots, u_{K_3,5}$ 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,

$$\begin{aligned}
\int_{AB} \beta_0 \frac{[u]}{h_{AB}} ds &= \frac{\beta_0 l_{AB}}{6h_{AB}} \left\{ \left(u_{K_3,1} + u_{K_3,2} + 4u_{K_3,3}\right) - \left(u_{K,1} + u_{K,2} + 4u_{K,3}\right) \right\} \\
\int_{AB} \overline{u_{\mathbf{n}}} ds &= \frac{l_{AB}}{2h_{AB}} \left\{ \left(-3u_{K_3,3} - u_{K_3,5} + 4u_{K_3,4}\right) + \left(3u_{K,3} + u_{K,5} - 4u_{K,4}\right) \right\} \\
\int_{AB} \beta_1 h_{AB} [u_{\mathbf{nn}}] ds &= \frac{4\beta_1 l_{AB}}{h_{AB}} \left\{ \left(u_{K_3,3} + u_{K_3,5} - 2u_{K_3,4}\right) - \left(u_{K,3} + u_{K,5} - 2u_{K,4}\right) \right\}.
\end{aligned} \tag{3.7}$$

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

$$\begin{aligned}
\overline{u}_K^{n+1} &= \overline{u}_K^n + \frac{\Delta t}{area(K)} \left\{ \int_{AB} \widehat{u_{\mathbf{n}}} ds + \int_{BC} \widehat{u_{\mathbf{n}}} ds + \int_{CA} \widehat{u_{\mathbf{n}}} ds \right\} \\
&= H\left\{ u_K^n(\cdot, \cdot), u_{K_1}^n(\cdot, \cdot), u_{K_2}^n(\cdot, \cdot), u_{K_3}^n(\cdot, \cdot) \right\}.
\end{aligned} \tag{3.8}$$

The functional $H(\cdots)$ of (3.8) involves total 28 arguments with 15 solution points values from elements K_1, K_2, K_3 and 13 solution points values from element K. The first 12 points from K are selected from evaluating $\int_{\partial K} \widehat{u_{\mathbf{n}}} ds$, see Fig. 1. The 13th one is to be selected by the quadrature rule for cell average $\overline{u}_K^n$, see Appendix A.2.

Our goal 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 quantity $\overline{u}_K^{n+1}$ is a convex combination of selected solution points values at time level t_n. 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,

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

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

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 following inequality (refer to Appendix A.2) which reflects geometrical property of the mesh partition,

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

where $\hat{\theta}$ is the minimum angle in the partition $\mathcal{T}_h$. From (3.7) and (3.8), we have,

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

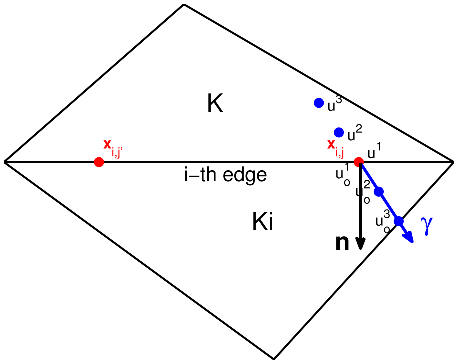


Fig. 2. Selected points along direction $\gamma = A^T(u^n(\mathbf{x}_{i,j}))\mathbf{n}_i$ for representing the numerical flux.

$$\begin{aligned}
\frac{\partial H}{\partial u_{K,3}} &= \frac{\partial \overline{u}_K^n}{\partial u_{K,3}} - \lambda \frac{l_{AB}}{h_{AB}}\left(\frac{2}{3}\beta_0 + 4\beta_1 - \frac{3}{2}\right) \geq \frac{\partial \overline{u}_K^n}{\partial u_{K,3}} - \lambda \frac{4\left(\beta_0 + 6\beta_1 - \frac{9}{4}\right)}{3\tan(\hat{\theta})} \\
\frac{\partial H}{\partial u_{K,4}} &= \frac{\partial \overline{u}_K^n}{\partial u_{K,4}} - \lambda \frac{2l_{AB}}{h_{AB}}(1-4\beta_1) \geq \frac{\partial \overline{u}_K^n}{\partial u_{K,4}} - \lambda \frac{4(1-4\beta_1)}{\tan(\hat{\theta})} \\
\frac{\partial H}{\partial u_{K,5}} &= \frac{\partial \overline{u}_K^n}{\partial u_{K,5}} - \lambda \frac{l_{AB}}{2h_{AB}}(8\beta_1 - 1) \geq \frac{\partial \overline{u}_K^n}{\partial u_{K,5}} - \lambda \frac{(8\beta_1 - 1)}{\tan(\hat{\theta})}.
\end{aligned} \tag{3.9}$$

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 A.2 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 estimate (A.6), we see $H(\cdots)$ is monotonically increasing on solution values $u_{K,1}$, $u_{K,2}$, $u_{K,3}$, $u_{K,4}$ 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 $H(u, \cdots, u) = u$ 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 $m \leq u_K^n(x,y), u_{K_1}^n(x,y), u_{K_2}^n(x,y), u_{K_3}^n(x,y) \leq M$. □

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 triangular meshes. 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,

$$\overline{u}_K^{n+1} = \overline{u}_K^n + \frac{\Delta t}{area(K)} \int_{\partial K} (A(\widehat{u)\nabla u} \cdot \mathbf{n})\, ds = \overline{u}_K^n + \frac{\Delta t}{area(K)} \int_{\partial K} \widehat{\nabla u \cdot \gamma}\, ds. \tag{3.10}$$

As discussed previously in (2.3), we apply a new method to calculate numerical flux $A(\widehat{u)\nabla u} \cdot \mathbf{n} = \widehat{\nabla u \cdot \gamma}$ with $\gamma = A^T(u)\mathbf{n}$ being a vector pointing from K into its neighbor along the edge. This is true because the diffusion matrix $A(u)$ is uniformly being positive definite and we have $\gamma \cdot \mathbf{n} = A^T(u)\mathbf{n} \cdot \mathbf{n} > 0$. 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 $\int_{\partial K} \widehat{\nabla u \cdot \gamma}\, ds$ on the three edges of element K. Notice that vector γ is a nonlinear function of the solution, thus we apply a quadrature rule to calculate $\int_{\partial K} \widehat{\nabla u \cdot \gamma}\, ds$. For example, we consider 2-point Gaussian quadrature rule on 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. On each Gaussian point $\mathbf{x}_{i,j}$, let's denote $\gamma = A^T(u^n(\mathbf{x}_{i,j}))\mathbf{n}_i$ with $u^n(\mathbf{x}_{i,j}) = (u_K^n(\mathbf{x}_{i,j}) + u_{K_i}^n(\mathbf{x}_{i,j}))/2$. Introduce the normalized vector $\overline{\boldsymbol{\gamma}} = \gamma/\|\gamma\|$, we see six solution values along γ direction, shown in Fig. 2, are sufficient to calculate the numerical flux,

$$\widehat{\nabla u \cdot \gamma}|_{\mathbf{x}_{i,j}} = \|\gamma\| \widehat{u_{\overline{\boldsymbol{\gamma}}}}|_{\mathbf{x}_{i,j}} = \|\gamma\| \left(\beta_0 \frac{[u]}{h_{i,j}} + \overline{u_{\overline{\boldsymbol{\gamma}}}} + \beta_1 h_{i,j}[u_{\overline{\boldsymbol{\gamma}}\overline{\boldsymbol{\gamma}}}]\right)\Bigg|_{\mathbf{x}_{i,j}}. \tag{3.11}$$

Again we denote $h_{i,j}$ as the shortest distance from point $\mathbf{x}_{i,j}$ to the other edges of K and K_i along γ. 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 $\mathcal{T}_h$. Given $u_K^n(x, y)$ in the range of $[m, M]$, we have $\overline{u}_K^{n+1} \in [m, M]$ provided,*

$$\beta_0 \geq \frac{3}{2} - 4\beta_1, \quad \frac{1}{8} \leq \beta_1 \leq \frac{1}{4}, \quad \lambda = \frac{\Delta t}{area(K)} \leq \frac{C(\beta_0, \beta_1, \mathcal{T}_h)}{\|A\|}. \tag{3.12}$$

Here $\|A\| = \max_u \|A(u)\|$ denotes the spectral norm of the diffusion matrix. Again (β_0, β_1) is the coefficient pair in the numerical flux (2.3), and $\hat{\theta}$ is the minimum angle of the partition $\mathcal{T}_h$. Constant C depends on β_0, β_1, and mesh information as,

$$C(\beta_0, \beta_1, \mathcal{T}_h) = \sin(\hat{\theta}) \cdot \frac{3-\sqrt{3}}{6} w_0 \cdot \min\left\{ \frac{1}{\beta_0 + 4\beta_1 - \frac{3}{2}}, \frac{2}{8\beta_1 - 1}, \frac{1}{2(1-4\beta_1)} \right\}, \tag{3.13}$$

where w_0 is the minimum quadrature weight (A.8) 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 (3.10). Specifically we 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 Fig. 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 (3.11). We denote u^1, u^2, u^3 as the three selected solution values of $u_K^n(x, y)$ 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 γ direction.

Now at the Gaussian point $\mathbf{x}_{i,j}$, we can explicitly write out the value of $\widehat{u_{\gamma}}$ of (3.11) with,

$$\begin{aligned}
[u] &= u_o^1 - u^1 \\
\overline{u_{\gamma}} &= \frac{1}{2}\left(\frac{-4u^2 + 3u^1 + u^3}{h_{i,j}} + \frac{4u_o^2 - 3u_o^1 - u_o^3}{h_{i,j}} \right) = \frac{1}{2h_{i,j}} \left(-4u^2 + 3u^1 + u^3 + 4u_o^2 - 3u_o^1 - u_o^3 \right) \\
[u_{\gamma\gamma}] &= \frac{u_o^1 + u_o^3 - 2u_o^2}{(\frac{h_{i,j}}{2})^2} - \frac{u^1 + u^3 - 2u^2}{(\frac{h_{i,j}}{2})^2} = \frac{4}{h_{i,j}^2}\left[u_o^1 + u_o^3 - 2u_o^2 - (u^1 + u^3 - 2u^2) \right].
\end{aligned} \tag{3.14}$$

Introduce notation $\alpha^{i,j} = \frac{\Delta t}{area(K)} l^i \omega_j \|A^T(u^n(\mathbf{x}_{i,j}))\mathbf{n}_i\|$ with l^i as the ith edge length and ω_j as the jth Gaussian point weight. For 2-point Gaussian quadrature rule, we have weight $\omega_j \equiv 1$. From (3.10)–(3.11) and (3.14), we have,

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

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. Now we consider the total 18 solution values contributed from element K.

Notice that with $\hat{\theta}$ denoting the minimum angle of the partition $\mathcal{T}_h$, we have,

$$\frac{l^i}{h_{i,j}} \leq \frac{1}{C_0 \sin\hat{\theta}}, \text{ with } C_0 \geq \frac{3-\sqrt{3}}{6}. \tag{3.15}$$

Again we need to have a special quadrature rule for the solution average $\overline{u}_K^n$ with the total 18 selected points included as quadrature points and having all the weights being positive. In Appendix A.3, we construct such a quadrature rule. Having w_0 denoting the minimum weight, we see condition (3.12) is sufficient to guarantee the coefficients of solution values used in K being non-negative. Therefore, $\overline{u}_K^{n+1}$ is monotonically increasing w.r.t. all the selected points inside K and its neighbors. With consistency and monotonicity, finally we have,

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

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

Implementation of the M-P-S limiter. Given the quadratic 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 $(x, y) \in K$.

$$\widetilde{u}_K^n(x, y) = \theta(u_K^n(x, y) - \overline{u}_K^n) + \overline{u}_K^n, \qquad \theta = \min\left\{1, \left|\frac{M - \overline{u}_K^n}{M_K - \overline{u}_K^n}\right|, \left|\frac{m - \overline{u}_K^n}{m_K - \overline{u}_K^n}\right|\right\}, \tag{3.16}$$

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

$$M_K = \max_{(x,y)\in K} u_K^n(x, y), \qquad m_K = \min_{(x,y)\in K} u_K^n(x, y). \tag{3.17}$$

Since $u_K^n(x, y)$ is a quadratic polynomial, it is easy to calculate its maximum and minimum value over K. Notice that the limiter (3.16) does not change the cell average. We have $\|\widetilde{u}_K^n(x, y) - u_K^n(x, y)\|_\infty = \mathcal{O}(h^3)$, 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 solution.

Algorithm 1 Maximum-principle-satisfying DDG scheme with interface correction.

```
1. At time level $t_n$, we apply M-P-S limiter (3.16)–(3.17) to $u_K^n(x, y)$ and obtain $\widetilde{u}_K^n(x, y)$.
2. Apply DDG with interface correction method (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–Lobatto points) at time level t_n. Thus the M-P-S limiter (3.16)–(3.17) 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 DDG with interface correction scheme (2.5) for spatial discretization. Same procedure as listed in Algorithm 1 is applied to guarantee the quadratic polynomial solution stay in the given bound and at the same time maintain the 3rd order accuracy.

Remark 3.2. Notice that Theorem 3.1 and Theorem 3.2 offer sufficient conditions to guarantee the numerical solutions satisfy strict maximum principle. Our analysis may not be optimal to maximize the CFL restriction. Practically a more relaxed CFL condition can be applied without losing 3rd order accuracy. In the following numerical Example 4.1 where we strictly apply exact solution upper and lower bounds at each time step with our M-P-S limiter, we do apply almost 10 times smaller than regular CFL to determine Δt. For the rest numerical examples in Section 4, we apply regular CFL (bigger CFL for porous medium equation) without observing accuracy loss.

4. Numerical examples

In this section, we present a sequence of numerical examples to demonstrate the accuracy and capability of M-P-S limiter. In all examples in this section, we take coefficient pair $(\beta_0, \beta_1) = (5, \frac{1}{8})$ 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 (3.16) applied on the following linear diffusion equation,

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

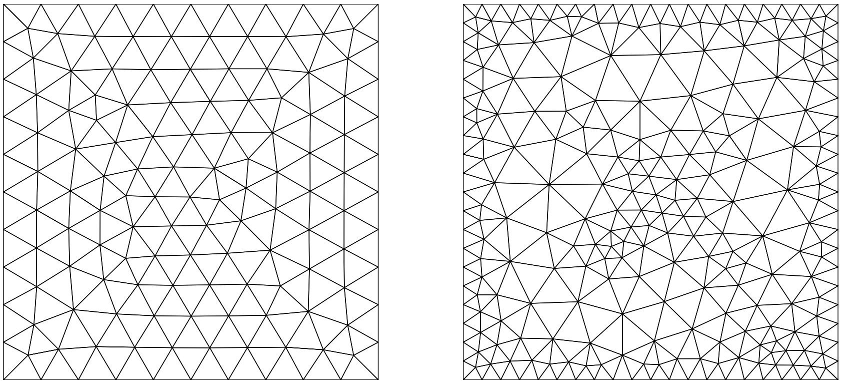
with initial data $u(x, y, 0) = u_0(x, y) = \sin(2\pi(x + y))$ and periodic boundary condition. The exact solution is given with,

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

Here, we take $\epsilon = 1$ and final time $t = 0.0001$. We implement the scheme with P^2 quadratic polynomials on unstructured mesh in Fig. 3(a) and on mesh with obtuse triangles with largest angle about $\frac{3}{5}\pi$ in Fig. 3(b). Third order of accuracy is maintained with and without M-P-S limiter (3.16) 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.

Example 4.2 *(Porous medium equation).* In this example we consider the nonlinear porous medium equation

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



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

Fig. 3. Illustration of meshes.

Table 1
Accuracy table on triangular mesh of Fig. 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	$u_{\min} - ue_{\min}$	$u_{\max} - ue_{\max}$
0.117	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 2
Accuracy table on unstructured mesh with obtuse triangles of Fig. 3(b).

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

with initial condition

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

and zero boundary condition. Piecewise quadratic polynomial solutions implemented on unstructured mesh (Fig. 3(a)) with size $h = 0.00733$ are shown in Fig. 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 Fig. 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),

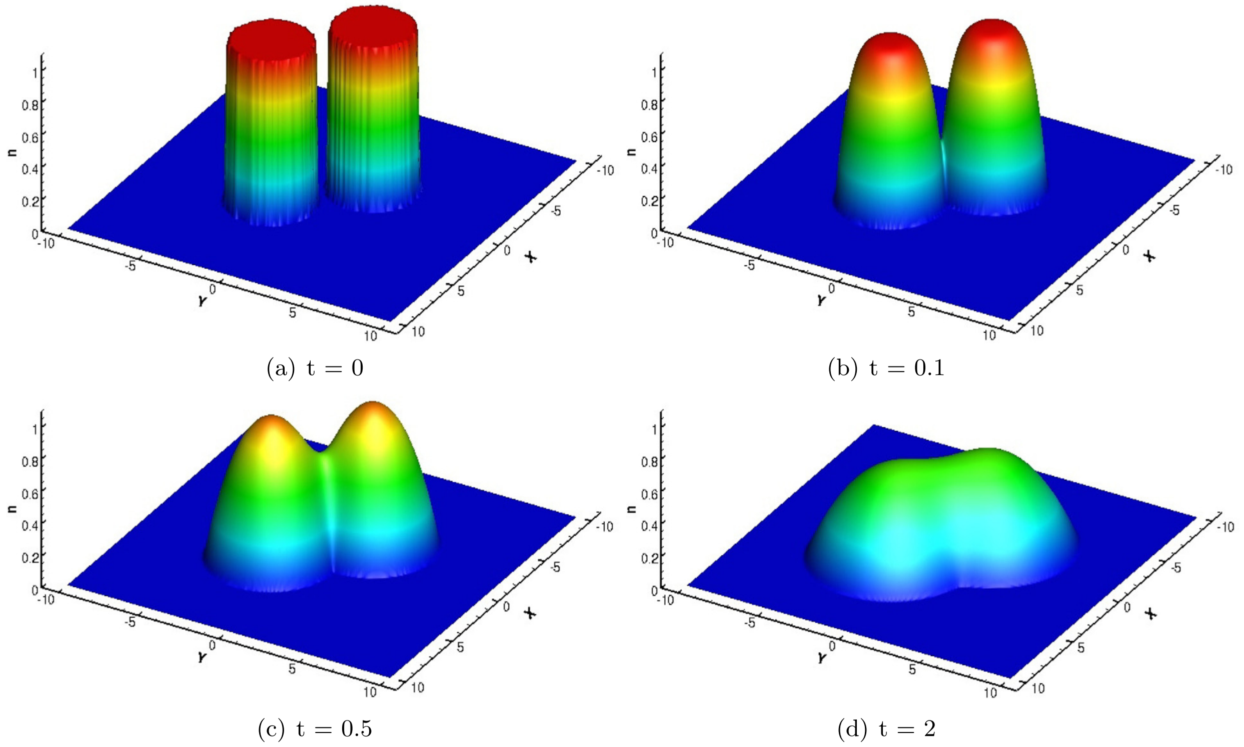
$$u_t + (u^2)_x + (u^2)_y = \epsilon(\nu(u)\nabla u_x)_x + \epsilon(\nu(u)\nabla u_y)_y, \quad (x, y) \in [-1.5, 1.5] \times [-1.5, 1.5].$$

Initial condition is given with,

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

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

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



(a) t = 0

(b) t = 0.1

(c) t = 0.5

(d) t = 2

Fig. 4. Nonlinear porous medium problem with $h = 0.00733$.

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 (Fig. 3(a)) with $h = 0.0147$ is shown in Fig. 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,

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

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

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

Quadratic P^2 implementations are carried out on mesh Fig. 3(a) and on mesh Fig. 3(b) with obtuse triangles. Errors and orders are listed in Table 3 and Table 4. Again we have $we_{\max}$ and $we_{\min}$ representing the exact solution maximum and

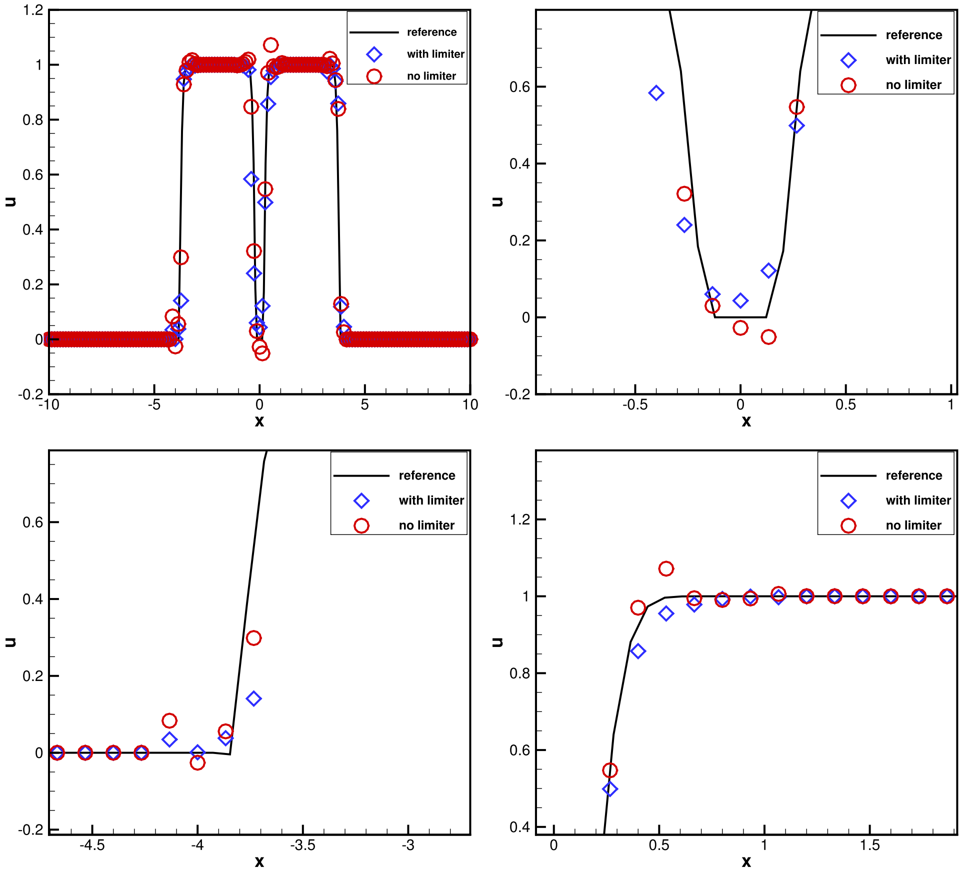


Fig. 5. The cut of the DDG interface correction solution along line $x + y = 0$ at $t = 0.005$. Red circle symbol: no M-P-S limiter. Blue diamond symbol: M-P-S limiter applied.

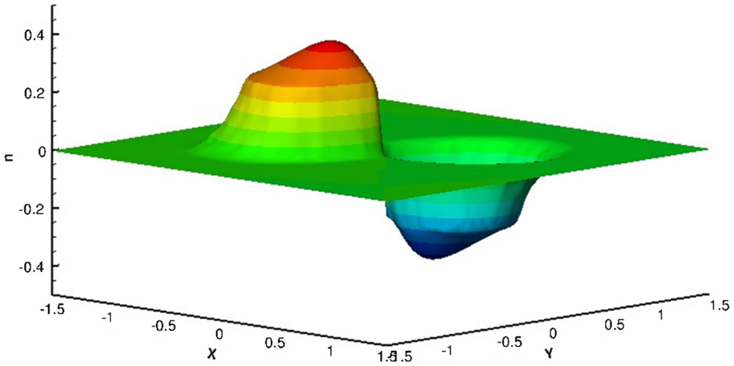


Fig. 6. Strongly degenerate parabolic problem solution at $t = 0.5$.

minimum values. We observe that the M-P-S limiter removes all overshoots and undershoots and still maintains the third order accuracy.

Vortex patch problem. Now we consider problem (4.1) with initial condition,

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

and periodic boundary condition. The Reynolds number is chosen to be $Re = 100$ or $Re = 10\,000$. 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 Fig. 3(a) is

Table 3
Accuracy check on mesh Fig. 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	$w_{\min} - we_{\min}$	$w_{\max} - we_{\max}$
0.117	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 4
Accuracy check on mesh Fig. 3(b) with obtuse triangles, 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 5
Maximum and minimum of the solutions, $Re = 100$ at $t = 0.1$.

$Re = 100$	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 6
Maximum and minimum of the solutions, $Re = 10\,000$ at $t = 0.1$.

$Re = 10\,000$	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

used and quadratic polynomials is applied. We also plot the solutions for the case $Re = 100$ at $t = 1$, shown in Fig. 7, and the case $Re = 10\,000$ at $t = 5$, shown in Fig. 8. It is clear that the overshoots and undershoots are removed with the M-P-S limiter applied.

Acknowledgements

Huang's work is supported by Natural Science Foundation of Zhejiang Province Grants Nos. LY14A010002 and LY12A01009.

Appendix A

We look for specific quadrature rule for the solution average $\overline{u}_K^n = \frac{1}{area(K)} \int_K u_K^n(x, y)dxdy$ in this section to complete the monotonicity discussion of Theorem 3.1 and Theorem 3.2. We construct such a quadrature rule that is exact for 2nd degree solution polynomial $u_K^n(x, y)$ on any triangle K and include all selected points as quadrature points with positive weights. Due to the fact that we have different selected points set for linear and nonlinear diffusion equations, we construct two quadrature rules for $\overline{u}_K^n$. For convenience, we introduce notation $|K|$ to represent the area of triangle K. In Section A.1, we first construct a quadrature rule for P^2 polynomial on any triangle such that vertices and edge centers are included in the quadrature points set and have all weights being positive. In Section A.2, we construct the desired quadrature rule for $\overline{u}_K^n$ that is used in Theorem 3.1. The result of Section A.1 are applied intensively to calculate the weights and estimate the lower bounds of the weights. In Section A.3, we construct another quadrature rule for $\overline{u}_K^n$ that is used in Theorem 3.2.

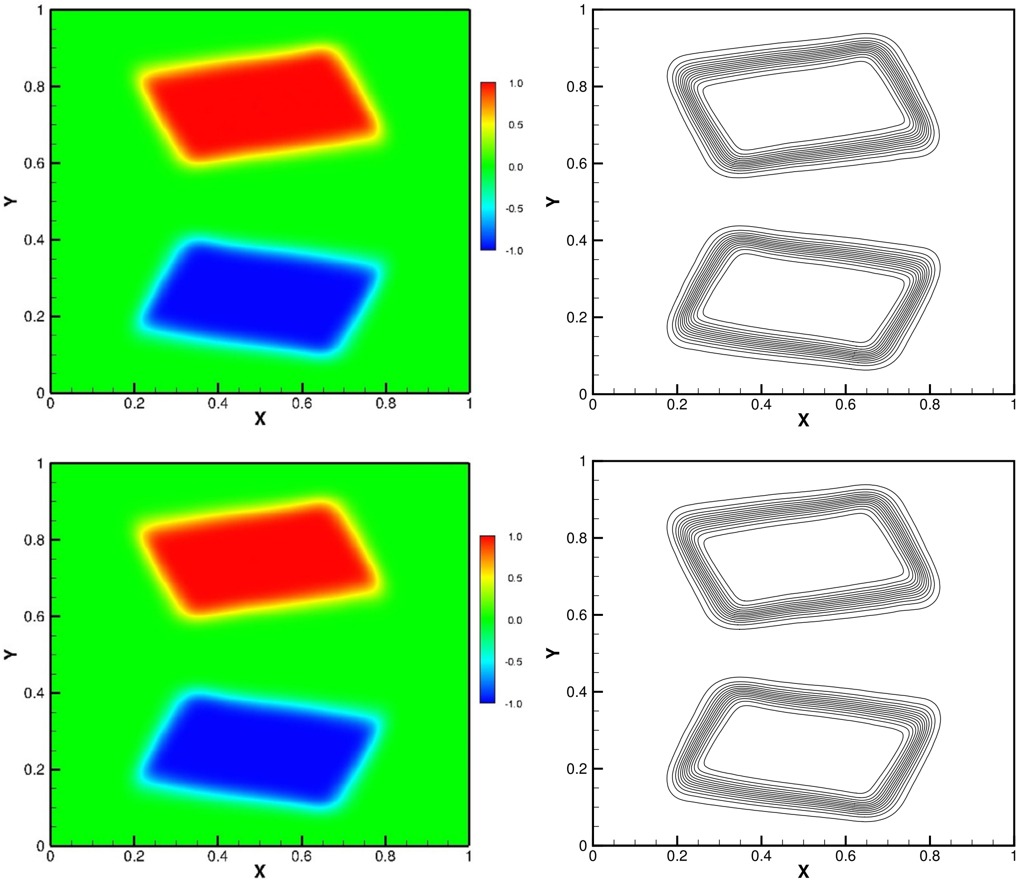


Fig. 7. Contours of the solutions, $Re = 100$, at $t = 1$ with mesh size $h = 0.0293 \times 2\pi$. Top: no M-P-S limiter ($w_{\min} = -1.0011$, $w_{\max} = 1.0008$); Bottom: add M-P-S limiter ($w_{\min} = -1$, $w_{\max} = 1$). 30 equally spaced contour lines are plotted.

Table 7
3-points Gauss–Lobatto.

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

A.1. Quadrature rule with vertices and edge centers as quadrature points

Now we follow [26] to construct one quadrature rule that is exact for quadratic polynomial $p(x, y)$ on any triangle T. All vertices and edge centers are included in the quadrature points set and we have positive weights for all quadrature points.

We start with a quadrature rule on unit square with vertices $\left(-\frac{1}{2}, -\frac{1}{2}\right)$, $\left(-\frac{1}{2}, \frac{1}{2}\right)$, $\left(\frac{1}{2}, -\frac{1}{2}\right)$ and $\left(\frac{1}{2}, \frac{1}{2}\right)$ on the u–v plane. After simple transformation/mapping, we convert the quadrature rule on square to the quadrature rule on any triangle T. The quadrature rule on square is constructed by tensor product. For variable u over $\left[-\frac{1}{2}, \frac{1}{2}\right]$, we take 3-points Gauss–Lobatto rule which is exact for polynomial up to degree 3. Let's have $\left\{(\widehat{u}^{\alpha}, \widehat{w}_{\alpha}) : \alpha = 1, 2, 3\right\}$ denoting the quadrature points and weights, see Table 7. For variable v over $\left[-\frac{1}{2}, \frac{1}{2}\right]$, we take the same quadrature rule and denote the quadrature points and weights as $\left\{(v^{\beta}, w_{\beta}) : \beta = 1, 2, 3\right\}$, see Table 7. For a two-variable polynomial $p(u, v)$ on square, we apply tensor product of 3-points Gauss–Lobatto in u and 3-points Gauss–Lobatto in v as the quadrature rule. The quadrature points on square are collected and denoted by $S^2 = \left\{\left(\widehat{u}^{\alpha}, v^{\beta}\right) : \alpha = 1, 2, 3; \beta = 1, 2, 3.\right\}$ with weights $\widehat{w}_{\alpha} w_{\beta}$, see Fig. 9(a).

For convenience, we consider position vector format to represent the vertices of T: $\mathbf{v}^1$, $\mathbf{v}^2$ and $\mathbf{v}^3$. For any point P in T, its position vector $\mathbf{P}$ can be described by barycentric coordinates (ξ_1, ξ_2, ξ_3), i.e., $\mathbf{P} = \xi_1\mathbf{v}^1 + \xi_2\mathbf{v}^2 + \xi_3\mathbf{v}^3$. The three vertices $\mathbf{v}^1$, $\mathbf{v}^2$ and $\mathbf{v}^3$ are oriented and marked clockwise. We define following three functions as projections from the square to the

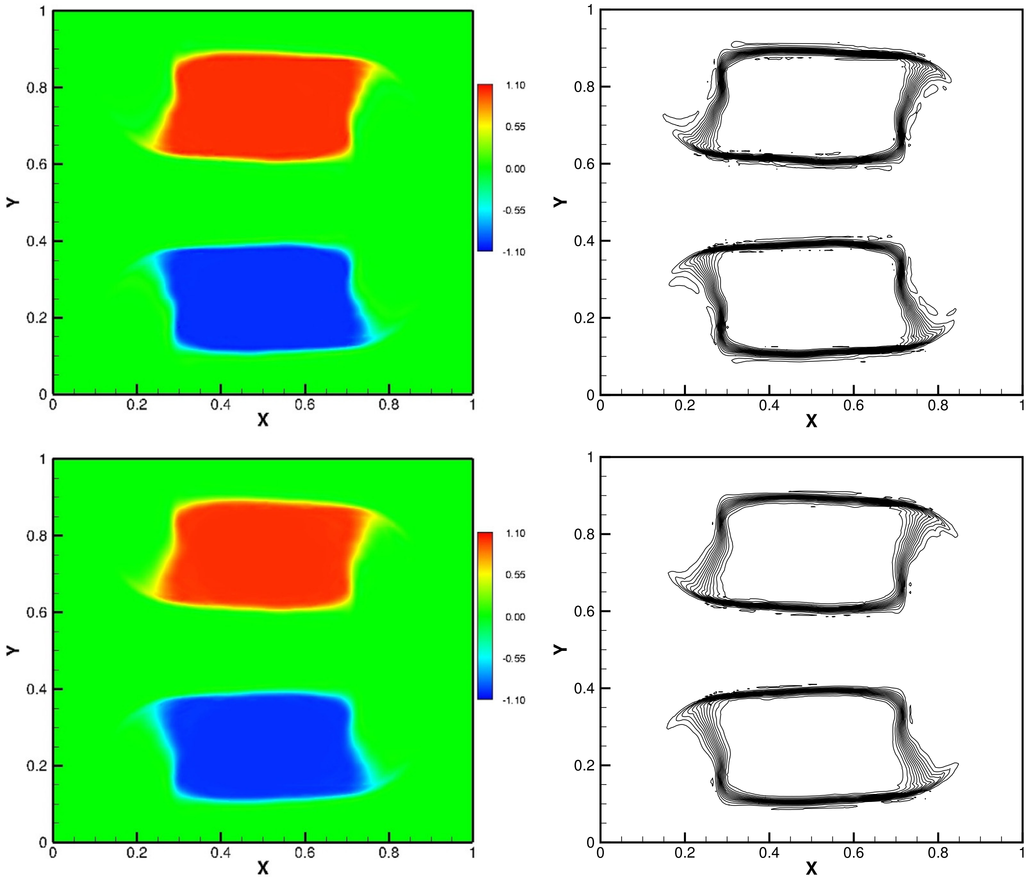
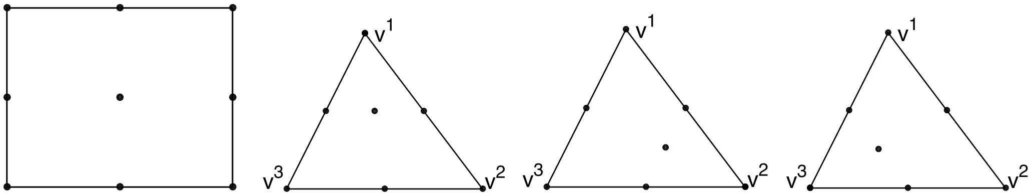


Fig. 8. Contours of the solutions, $Re = 10\,000$ at $t = 5$ and mesh size $h = 0.0293 \times 2\pi$. Top: no M-P-S limiter ($w_{\min} = -1.1211$, $w_{\max} = 1.1508$); Bottom: add M-P-S limiter ($w_{\min} = -1$, $w_{\max} = 1$). 30 equally spaced contour lines are plotted.



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

Fig. 9. Illustration of quadrature points mapping from the square to a triangle T.

triangle T. Each projection maps the top edge of the square to one vertex and maps the rest three edges to the edges of the triangle T.

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

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

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

Under each projection, points set S^2 is mapped onto T and denoted as $\mathbf{g}_i(S^2)$ $(i = 1, 2, 3)$, see Fig. 9(b)–Fig. 9(d) for the three projections. The collection set is $S_T^2 = \mathbf{g}_1(S^2) \cup \mathbf{g}_2(S^2) \cup \mathbf{g}_3(S^2)$. Now we are ready to use $\mathbf{g}_i$ $(i = 1, 2, 3)$ and S_T^2 to construct our quadrature rule on T. Notice that for any $i = 1, 2, 3$, we have $\int_T p(x,y)\,dxdy = \int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} p(\mathbf{g}_i(u,v)) \left| \frac{\partial \mathbf{g}_i(u,v)}{\partial(u,v)} \right| dudv$. With the three vertices orientated clockwise, the Jacobian is $\left|\frac{\partial \mathbf{g}_i(u,v)}{\partial(u,v)}\right| = 2|T|(\frac{1}{2}-v)$. Given a P^2 quadratic polynomial $p(x,y)$ on T with average $\overline{p} = \frac{1}{|T|}\int_T p(x,y)dxdy$, we have,

$$\begin{aligned}
\frac{1}{|T|}\int\limits_T p(x,y)\,dxdy &= \frac{1}{|T|}\int\limits_{-\frac{1}{2}}^{\frac{1}{2}}\int\limits_{-\frac{1}{2}}^{\frac{1}{2}} p(\mathbf{g}_i(u,v))\left|\frac{\partial \mathbf{g}_i(u,v)}{\partial(u,v)}\right| dudv, \\
&= \int\limits_{-\frac{1}{2}}^{\frac{1}{2}}\int\limits_{-\frac{1}{2}}^{\frac{1}{2}} p(\mathbf{g}_i(u,v))2(\frac{1}{2}-v)\,dudv = \sum_{\alpha=1}^{3}\sum_{\beta=1}^{3} p(\mathbf{g}_i(\widehat{u}^\alpha, v^\beta))2(\frac{1}{2}-v^\beta)\widehat{w}_\alpha w_\beta \\
&= \frac{1}{3}\sum_{i=1}^{3}\sum_{\alpha=1}^{3}\sum_{\beta=1}^{3} p(\mathbf{g}_i(\widehat{u}^\alpha, v^\beta))2(\frac{1}{2}-v^\beta)\widehat{w}_\alpha w_\beta = \sum_{\mathbf{x}\in S_T^2} p(\mathbf{x})\bar{w}_{\mathbf{x}}.
\end{aligned} \tag{A.1}$$

Here the previous quadrature rule on square is applied and is exact for polynomial $p(\mathbf{g}_i(u,v))2(\frac{1}{2}-v)$, since it has power of 2 for variable u and has power of 3 for variable v.

Finally we obtain the desired quadrature rule on triangle T that is exact for P^2 polynomial and have vertices and edge centers included as quadrature points (see definition of S_T^2). The weights $\bar{w}_{\mathbf{x}}$ of $\mathbf{x}\in S_T^2$ are obviously non-negative. Regrouping the quadrature points of (A.1) to subgroup of vertices, subgroup of edge centers and the rest, quadrature rule (A.1) is rewritten as,

$$\overline{p} = \sum_{i=1}^{3} p(\mathbf{v}^i)\bar{w}_i + \sum_{j=1}^{3} p(\frac{\mathbf{v}^j + \mathbf{v}^{j+1-3\lfloor\frac{j}{3}\rfloor}}{2})\tilde{w}_j + \sum_{\mathbf{x}\in S_T^2\backslash\{\mathbf{v}^1,\mathbf{v}^2,\mathbf{v}^3,\frac{\mathbf{v}^1+\mathbf{v}^2}{2},\frac{\mathbf{v}^2+\mathbf{v}^3}{2},\frac{\mathbf{v}^3+\mathbf{v}^1}{2}\}} p(\mathbf{x})\bar{w}_{\mathbf{x}}.$$

It turns out the weights $\{\bar{w}_i\}_{i=1}^3$ on three vertices $\mathbf{v}^1$, $\mathbf{v}^2$ and $\mathbf{v}^3$ are the same. We obtain one (i.e. vertex $\mathbf{v}^1$ with barycentric coordinates $(1,0,0)$) with,

$$\bar{w}_1 = \frac{2}{3}\left[\frac{1}{2}-\left(-\frac{1}{2}\right)\right]\widehat{w}_1 w_1 + \frac{2}{3}\left[\frac{1}{2}-\left(-\frac{1}{2}\right)\right]\widehat{w}_3 w_1 = \frac{2}{3}\left(\widehat{w}_1+\widehat{w}_3\right)w_1 = \frac{1}{27}. \tag{A.2}$$

The weights of $\left\{\tilde{w}_j\right\}_{j=1}^3$ on three edge centers are the same too. For example we take edge center $\frac{\mathbf{v}^1+\mathbf{v}^2}{2}$ with barycentric coordinates $(\frac{1}{2},\frac{1}{2},0)$ as the one with weight,

$$\tilde{w}_1 = \frac{2}{3}\left[\frac{1}{2}-\left(-\frac{1}{2}\right)\right]\widehat{w}_2 w_1 + \frac{2}{3}\left[\frac{1}{2}-0\right]\widehat{w}_3 w_2 + \frac{2}{3}\left[\frac{1}{2}-0\right]\widehat{w}_1 w_2 = \frac{4}{27} = 4\bar{w}_1. \tag{A.3}$$

Remark A.1. This section only provides one method to construct a quadrature rule on any triangle T with positive weights, and include three vertices and three edge centers as quadrature points.

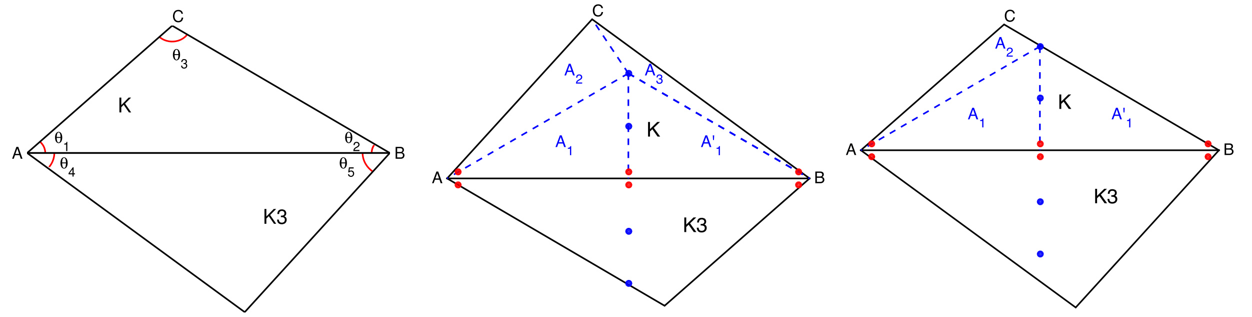
A.2. Quadrature rule I

In this section, we construct the quadrature rule for $\overline{u}_K^n$ with selected points as quadrature points and show positive weights are obtained for all quadrature points. Then we estimate the lower bounds of the weights which are used in inequalities (3.9) to bound the CFL condition. Notice that the selected points are chosen from evaluating the numerical flux line integral on three elements edges $\int_{\partial K}\widehat{u_{\mathbf{n}}}\,ds$. Thus we can focus on the selected points contributed from one edge, i.e $\int_{AB}\widehat{u_{\mathbf{n}}}\,ds$, to illustrate the method to find such a quadrature rule.

As shown in Fig. 1, the five points, namely $\mathbf{x}_{K,i}$ $(i=1,\cdots,5)$, are the selected points contributed from edge AB. The quadrature rule we investigate is written out in following format to include $\mathbf{x}_{K,i}$ $(i=1,\cdots,5)$ in the quadrature points set,

$$\frac{1}{3}\overline{u}_K^n = \frac{1}{3|K|}\int\limits_K u_K^n(x,y)dxdy = \sum_{i=1}^{5} w_i u_K^n(\mathbf{x}_{K,i}) + \sum_{j=1}^{l} w_j^* u_K^n(\mathbf{x}_{AB,j}^*). \tag{A.4}$$

Then we apply the same procedure to the edge of BC and CA, sum over three edges and finally obtain a quadrature rule for $\overline{u}_K^n$ that include total 12 selected points as quadrature points. Notice that one vertex is shared by two edges. The rest quadrature points from (A.4) and other two edges can be replaced by one point inside K with positive weight. Here we



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

Fig. 10. Left: element K with its neighbor K_3; Right: two cases of points selected along normal vector through edge AB center.

mark it as the 13-th point in the quadrature rule. Now let's demonstrate the details to obtain (A.4) with positive weights and calculate the weight size of w_i, $i = 1, \cdots, 5$.

From the discussion of Theorem 3.1, we have two possible locations of the five points $\mathbf{x}_{K,i}$ $(i = 1, \cdots, 5)$, see the two cases in Figs. 10(b)–10(c). In either case, we divide triangle K into smaller triangles and apply the quadrature rule of section A.1 on each small triangle to obtain $\overline{u}^n_K$, then collect the quadrature weights for each of the five points $\mathbf{x}_{K,i}$ to evaluate w_i in (A.4).

As shown in Fig. 1 or Fig. 10, geometrical information of K and K_3 (Fig. 10(a)) determines the locations of $\mathbf{x}_{K,4}$ and $\mathbf{x}_{K,5}$ (blue dots). We have: case (1) with $\min(\theta_1, \theta_2) > \min(\theta_4, \theta_5)$, then $\mathbf{x}_{K,5}$ lies strictly inside triangle K, see Fig. 10(b); or case (2) with $\min(\theta_1, \theta_2) \le \min(\theta_4, \theta_5)$, then $\mathbf{x}_{K,5}$ sits exactly on the edge of K, see Fig. 10(c). Now we can calculate and estimate the quadrature weight of w_i $(i = 1, \cdots, 5)$ in each case.

For case (1), we divide the triangle K into four small triangles and mark them as A_1, A_1', A_2 and A_3 (Fig. 10(b)) and we have $\int_K u^n_K(x, y)dxdy = \int_{A_1 + A_1' + A_2 + A_3} u^n_K(x, y)dxdy$. Apply the quadrature rule of Appendix A.1 over each small triangle and we can collect the weight of $\mathbf{x}_{K,i}$. For example, point $\mathbf{x}_{K,1}$ (vertex A) is shared by triangle A_1 and A_2. From quadrature weight formula (A.2) and quadrature definition (A.4), we have $w_1 = \bar{w}_1(|A_1| + |A_2|)/(3|K|)$. Here $|A_1|$ represents the area of triangle A_1. Similarly we have $w_2 = \bar{w}_1(|A_1| + |A_3|)/(3|K|)$ and $w_3 = 2\bar{w}_1|A_1|/(3|K|)$, here we use the fact that $\mathbf{x}_{K,3}$ is the center of edge AB and we have $|A_1| = |A_1'|$. For point $\mathbf{x}_{K,5}$, we have $w_5 = \bar{w}_1(2|A_1| + |A_2| + |A_3|)/(3|K|)$. Notice that point $\mathbf{x}_{K,4}$ is the edge center of triangle A_1 and A_1', from (A.3) we have $w_4 = 2\tilde{w}_1|A_1|/(3|K|) = 8\bar{w}_1|A_1|/(3|K|)$. Similarly for the case (2) of 10(c), we have $w_1 = \bar{w}_1(|A_1| + |A_2|)/(3|K|)$, $w_2 = \bar{w}_1|A_1|/(3|K|)$, $w_3 = 2\bar{w}_1|A_1|/(3|K|)$, $w_4 = 8\bar{w}_1|A_1|/(3|K|)$, $w_5 = \bar{w}_1(2|A_1| + |A_2|)/(3|K|)$.

Having the quadrature weights available on all triangle elements over the partition $\mathcal{T}_h$, we can estimate and obtain positive lower bounds based on mesh geometrical information. Again, we consider element K, the triangle ΔABC and edge AB as the example to estimate the lower bound. Denote H_{AB} as the height from vertex C to edge AB and h_{AB} as the height of smaller triangle A_1 to the same edge AB, we have

$$\theta_{AB} = \frac{h_{AB}}{H_{AB}} = \frac{1}{2}\tan(\min(\theta_4, \theta_5))\left(\cot(\theta_1) + \cot(\theta_2)\right). \tag{A.5}$$

Therefore, locally we can obtain the estimates of the weights w_i $(i = 1, \cdots, 5)$, as listed in Table 8. We apply same procedure to edge BC and CA and obtain the quadrature rule with total 12 selected points as quadrature points. Collect the data from the three edges (refer to (A.4)), we obtain lower bounds on the weights as follows,

$$\frac{\partial \overline{u}^n_K}{\partial u_{K,1}}, \frac{\partial \overline{u}^n_K}{\partial u_{K,2}}, \frac{\partial \overline{u}^n_K}{\partial u_{K,3}} \ge \frac{\bar{w}_1}{3} \cdot \theta_0, \qquad \frac{\partial \overline{u}^n_K}{\partial u_{K,4}} \ge \frac{4\bar{w}_1}{3}\theta_0, \qquad \frac{\partial \overline{u}^n_K}{\partial u_{K,5}} \ge \frac{\bar{w}_1}{3}, \tag{A.6}$$

where θ_0 is the minimum ratio of h_{AB}/H_{AB} over all edges in the triangular mesh partition $\mathcal{T}_h$,

$$\theta_0 = \min_{\mathcal{T}_h}\left(\frac{h_{AB}}{H_{AB}}\right). \tag{A.7}$$

To finish the discussion, we only need to show there exists the 13th point inside K with positive weight such that the quadrature rule include previously described 12 selected points and this extra one in the quadrature points set. Now let's assign new notations to the quadrature points, and denote $\mathbf{x}_{K,j}$ $(j = 1, \cdots, 12)$ as the 12 selected points with weights w_j $(j = 1, \cdots, 12)$. We also reorder the rest points and denote them as $\mathbf{x}^*_{K,j^*}$ with weights $w^*_{j^*}$ $(j^* = 1, \cdots, m$, where m is an integer). Let $w_{13} = \sum_{j^*=1}^m w^*_{j^*}$, then we have $\sum_{j=1}^{13} w_j = \sum_{j=1}^{12} w_j + \sum_{j^*=1}^m w^*_{j^*} = 1$. This is true because the quadrature rule we construct is exact for polynomials up to degree 2 and trivially exact for constant function. Furthermore we see $\frac{1}{w_{13}}\sum_{j^*=1}^m w^*_{j^*} u^n_K(\mathbf{x}^*_{K,j^*}) = \sum_{j^*=1}^m \frac{w^*_{j^*}}{w_{13}} u^n_K(\mathbf{x}^*_{K,j^*})$ is a convex combination of the point values $u^n_K(\mathbf{x}^*_{K,j^*})$ $(j^* = 1, \cdots, m)$.

Table 8
Local estimates of the quadrature weights. $\theta_{AB} = h_{AB}/H_{AB}$ of (A.5).

Point	Case 1	Case 2
$\mathbf{x}_{K,1}$	$\frac{\lvert A_1\rvert+\lvert A_2\rvert}{3\lvert K\rvert}\bar{w}_1 \geq \frac{\bar{w}_1}{6}\cdot\theta_{AB}$	$\frac{\lvert A_1\rvert+\lvert A_2\rvert}{3\lvert K\rvert}\bar{w}_1 \geq \frac{\bar{w}_1}{6}\cdot\theta_{AB}$
$\mathbf{x}_{K,2}$	$\frac{\lvert A_1\rvert+\lvert A_3\rvert}{3\lvert K\rvert}\bar{w}_1 \geq \frac{\bar{w}_1}{6}\cdot\theta_{AB}$	$\frac{\lvert A_1\rvert}{3\lvert K\rvert}\bar{w}_1 \geq \frac{\bar{w}_1}{6}\cdot\theta_{AB}$
$\mathbf{x}_{K,3}$	$\frac{2\lvert A_1\rvert}{3\lvert K\rvert}\bar{w}_1 \geq \frac{\bar{w}_1}{3}\cdot\theta_{AB}$	$\frac{2\lvert A_1\rvert}{3\lvert K\rvert}\bar{w}_1 \geq \frac{\bar{w}_1}{3}\cdot\theta_{AB}$
$\mathbf{x}_{K,4}$	$\frac{2\lvert A_1\rvert}{3\lvert K\rvert}4\bar{w}_1 \geq \frac{4\bar{w}_1}{3}\cdot\theta_{AB}$	$\frac{2\lvert A_1\rvert}{3\lvert K\rvert}4\bar{w}_1 \geq \frac{4\bar{w}_1}{3}\cdot\theta_{AB}$
$\mathbf{x}_{K,5}$	$\frac{2\lvert A_1\rvert+\lvert A_2\rvert+\lvert A_3\rvert}{3\lvert K\rvert}\bar{w}_1 = \frac{\bar{w}_1}{3}$	$\frac{2\lvert A_1\rvert+\lvert A_2\rvert}{3\lvert K\rvert}\bar{w}_1 = \frac{\bar{w}_1}{3}$

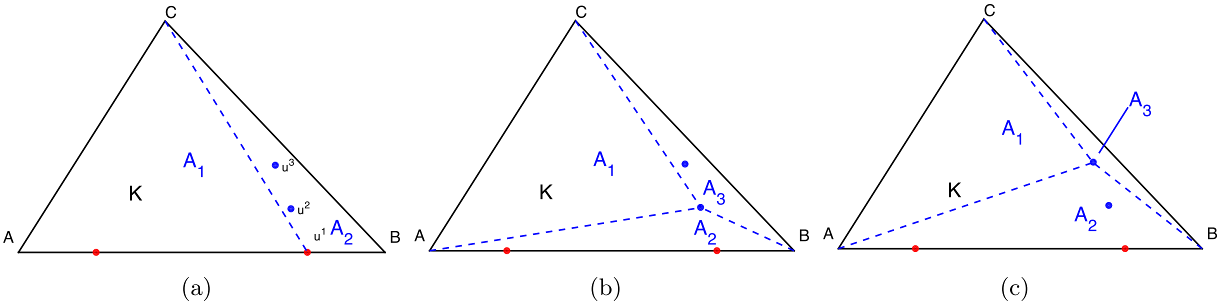


Fig. 11. Illustrations of dividing triangular element K into 2 or 3 small triangles.

By the mean value theorem, one can find a point $\mathbf{x}_{K,13}$ inside the convex hall of the points $\mathbf{x}^*_{K,j^*}$, $(j^* = 1, \cdots, m)$ such that $u^n_K(\mathbf{x}_{K,13}) = \sum_{j^*=1}^m \frac{w^*_{j^*}}{w_{13}} u^n_K(\mathbf{x}^*_{K,j^*})$. Therefore, we can construct a quadrature rule for the solution average with positive weights of total 13 points inside K which include the 12 selected points.

A.3. Quadrature rule II

In this section, we construct another quadrature rule for the solution average $\overline{u}^n_K$ that is used in Theorem 3.2 for nonlinear diffusion equations. Again the quadrature rule should include all selected points as quadrature points and have all positive weights.

According to Theorem 3.2 we consider two-points Gaussian quadrature rule to calculate the line integrals on the three edges of K. We have three selected points used for each Gaussian point, thus we have total 18 selected points. Now we need to include these 18 points as quadrature points for the average $\overline{u}^n_K$. Similar to the method used in Appendix A.2, see (A.4) and the discussion after, we consider one point at a time by including this point in the quadrature rule of $\frac{1}{18}\overline{u}^n_K$. Recall that for each Gaussian point we have three selected points used, in Fig. 11 we show how to cut the triangle K into smaller triangles and include that one selected point in $\frac{1}{18}\overline{u}^n_K$. Again we apply the quadrature rule of Appendix A.1 over each small triangle and collect the weights. Simple calculation shows the total 18 selected points share the same weight as,

$$w_0 = \frac{1}{18}\bar{w}_1 = \frac{1}{486}. \tag{A.8}$$

Remark A.2. This section only indicates we can construct such a quadrature rule for $\overline{u}^n_K$ with selected points as quadrature points. The constant weight we have is small but is independent of the nonlinear $\gamma = A^T(u^n(\mathbf{x}_{i,j}))\mathbf{n}_i$ direction along which three points are selected at each Gaussian point $\mathbf{x}_{i,j}$. The simple quadrature rule is constructed with no intention to maximize the quadrature weight.

References

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

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

[3] B. Cockburn, C. Johnson, C.-W. Shu, E. Tadmor, Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lect. Notes Math., vol. 1697, 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, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection–diffusion systems, SIAM J. Numer. Anal. 35 (6) (1998) 2440–2463 (electronic).

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

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

[7] S. Holst, An a priori error estimate for a monotone mixed finite-element discretization of a convection–diffusion problem, Numer. Math. 109 (2008) 101–119, http://dx.doi.org/10.1007/s00211-007-0097-7.

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

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

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

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

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

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

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

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

[16] V. Thomée, Galerkin Finite Element Methods for Parabolic Problems, 2nd ed., Springer Ser. Comput. Math., vol. 25, Springer-Verlag, Berlin, 2006.

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

[18] T. Vejchodský, On the nonnegativity conservation in semidiscrete parabolic problems, in: Conjugate Gradient Algorithms and Finite Element Methods, in: Sci. Comput., Springer, Berlin, 2004, pp. 197–210.

[19] C. Vidden, J. Yan, Direct discontinuous Galerkin method for diffusion problems with symmetric structure, J. Comput. Math. 31 (6) (2013) 638–662.

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

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

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

[23] P. Yang, T. Xiong, J. Qiu, Z. Xu, High order maximum principle preserving flux finite volume method for convection dominated problems, J. Sci. Comput. (2015), submitted for publication.

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

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

[26] X. Zhang, Y. Xia, 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) (2012) 29–62.

[27] X. Zhang, Y. Liu, C.-W. Shu, Maximum-principle-satisfying high order finite volume weighted essentially nonoscillatory schemes for convection–diffusion equations, SIAM J. Sci. Comput. 34 (2) (2012) A627–A658.

[28] Y. Zhang, X. Zhang, C.-W. Shu, Maximum-principle-satisfying second order discontinuous Galerkin schemes for convection–diffusion equations on triangular meshes, J. Comput. Phys. 234 (2013) 295–316.