Journal of Computational Physics 229 (2010) 3382–3401

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Linearity preserving nine-point schemes for diffusion equation on distorted quadrilateral meshes

Jiming Wu *, Zihuan Dai, Zhiming Gao, Guangwei Yuan

Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, P.O. Box 8009, Beijing 100088, PR China

ARTICLE INFO

Article history:
Received 30 March 2009
Received in revised form 9 December 2009
Accepted 6 January 2010
Available online 15 January 2010

Keywords:
Diffusion equation
Difference scheme
Linearity preserving method
Distorted mesh

ABSTRACT

In this paper, we employ the so-called linearity preserving method, which requires that a difference scheme should be exact on linear solutions, to derive a nine-point difference scheme for the numerical solution of diffusion equation on the structured quadrilateral meshes. This scheme uses firstly both cell-centered unknowns and vertex unknowns, and then the vertex unknowns are treated as a linear combination of the surrounding cell-centered unknowns, which reduces the scheme to a cell-centered one. The weights in the linear combination are derived through the linearity preserving approach and can be obtained by solving a local linear system whose solvability is rigorously discussed. Moreover, the relations between our linearity preserving scheme and some existing schemes are also discussed, by which a generalized multipoint flux approximation scheme based on the linearity preserving criterion is suggested. Numerical experiments show that the linearity preserving schemes in this paper have nearly second order accuracy on many highly skewed and highly distorted structured quadrilateral meshes.

1. Introduction

In many computational problems, such as the computational radiation hydrodynamics and the reservoir simulation, the discretization of the following diffusion term

$$\nabla \cdot (\kappa(x,y)\nabla u) \tag{1.1}$$

is of great interests. In the radiation hydrodynamics, u can be a certain material temperature and $\kappa(x,y)$ is the diffusion coefficient, while in reservoir simulation, u and $\kappa(x,y)$ denote the pressure and permeability, respectively.

In discretizing (1.1), one usually has to face two difficulties, i.e., (i) $\kappa(x,y)$ is discontinuous and strongly nonlinear, (ii) the mesh is highly distorted and highly skewed which usually occurs in the Lagrangian or ALE hydrodynamic computations [8,10]. There has been extensive study on developing efficient numerical schemes for (1.1), the issues about which range from the classical stability and accuracy to some other desirable numerical properties, including symmetry and positive definiteness of the resulting linear system, local stencil, local conservation, positivity preserving or monotonicity, simplicity, robustness, cost-efficiency, etc. To our knowledge, there exists no scheme satisfying all the above properties. Usually, a scheme possesses some properties at the cost of losing other ones. Among the aforementioned desirable properties, from our point of view, the accuracy and stability are the fundamental ones.

In this paper, we are more interested in the so-called linearity preserving property, which says that a difference scheme is exact on linear solutions. We observe that some authors mentioned this property in their works [6,7,22,23,27], for example,

* Corresponding author.
E-mail addresses: wu_jiming@iapcm.ac.cn (J. Wu), dai_zihuan@iapcm.ac.cn (Z. Dai), gao@iapcm.ac.cn (Z. Gao), yuan_guangwei@iapcm.ac.cn (G. Yuan).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.01.007

the authors in [22] suggested a so-called MDHW scheme, which preserves the homogeneous linear solutions $a + bx + cy$ and $a + br + cz$ in $x - y$ and $r - z$ geometries, respectively. To our knowledge, the authors usually claimed that certain scheme has or has not the linearity preserving property, but there has been no intensive investigation or study on this topic. Although the linearity preserving property has not been proved theoretically to be a sufficient or necessary condition for certain good numerical properties mentioned above, we observe from our numerical practice and other peoples' works that a difference scheme with linearity preserving property usually has good accuracy on highly distorted meshes. Motivated by this observation, we suggest here the so-called *linearity preserving criterion*, which requires that each step of the derivation of a difference scheme for diffusion equation is exact or linearly exact, i.e., exact in the sense whenever the solution is a linear function and the diffusion coefficient $\kappa(x, y)$ is a constant. Obviously, a difference scheme derived from the linearity preserving criterion is exact for the linear solution.

The linearity preserving criterion is applied here to improve the accuracy of a special difference scheme suggested originally in [20] through a control volume approach. In this scheme, the normal component of the flux (see (2.5)) on each cell edge is explicitly expressed by the two cell-centered unknowns with respect to the cells sharing that edge, and the two vertex unknowns defined at the two endpoints of the edge. Usually the vertex unknowns are treated as intermediate ones and are expressed by a linear combination of the surrounding cell-centered unknowns. On structured quadrilateral meshes, the above scheme involves nine cell-centered unknowns and as a result, is often called as the nine-point scheme (NPS) [14,28]. We note that Klausen and Winther [19] once gave a definition of the multipoint flux approximation (MPFA), which states that the MPFA discretization is a control volume method where more than two pressure values (here the values of u) are used to give an explicit discrete flux expression. According to this recent definition of MPFA, NPS can also be viewed as, to some sense, a kind of MPFA scheme.

The most important features of NPS are that it has a very simple explicit expression of the flux, involves less amount of computational cost and is easy for coding, so that NPS has been used for a long time in some hydrodynamics codes, such as LARED-I and MARED [11,26]. The main disadvantage of NPS is that it loses accuracy on highly distorted meshes, which is caused mainly by the rough or improper treatments on the vertex unknowns. How to improve the accuracy of NPS is a very interesting problem and has drawn some authors' attentions [5,7,28,30,33]. These improvements are either complicated and costly or not accurate enough. It is evident that a desirable improvement on NPS should keep its main advantages so that it is simple for coding and involves less amount of computational cost and moreover, does not result in a major change of the original codes.

We use the linearity preserving criterion to rederive the NPS and further, to obtain a simple treatment for the vertex unknowns. The computational cost of this new vertex unknown treatment is approximately one third of that in [28] and increases the accuracy to almost second order on many typical highly distorted and highly skewed meshes. As done in [28], our improvement treats the discontinuity rigorously and furthermore, is obtained not at the cost of massive change of the original codes. Since we design our algorithm by using the continuity of the flux, the scheme derived here also keeps the local conservation. For a treatment of vertex unknown, both the method in [28] and our present algorithm here depend upon the solution of a local 4×4 linear system, whose unknowns in the former are the weights in the linear combination mentioned above while in the present paper are some newly introduced ones. Compared with [28], the present local linear system has a simple structure and a simple explicit expression for its entries, which reduces largely the computational cost and makes it possible to analyze solvability.

In using the linearity preserving method to find a treatment for the vertex unknowns, we employ an MPFA-type technique to introduce some intermediate cell edge unknowns. The only difference is that our cell edge unknown is defined at a dynamic point on the whole edge, instead of a fixed point (known as continuity point) in certain half edge, such as the midpoint of a cell edge used in the usual MPFA type schemes. The idea to choose a dynamic point on the whole cell edge enables us not only to obtain a robust algorithm for the vertex unknowns but also to construct a generalized MPFA scheme.

In the construction of many discretization schemes for the diffusion equation, such as the local support operator scheme (LSOM) [23], the local flux mimetic finite difference scheme (LFMFD) [21], the physical space derived MPFA [1,19] together with its variations [9], the reference space derived MPFA [3,29], and the nine-point scheme in [28], one has to solve certain local linear systems. The local systems in some schemes, such as LSOM and the symmetric version of LFMFD, are symmetric and positive definite for meshes consisting of convex cells and as a consequence, the solvability of the local systems follows immediately. However, the solvability of the local linear systems in other schemes is seldom discussed and the corresponding algorithms run the risk of being breakdown in the computational course. This problem is neglected mainly because the breakdown rarely occurs in practical computation, however, theoretically speaking, it does exist. To our practice, the possibility for the breakdown increases when the cells in the mesh approach concave ones. We note that the authors in [19] obtained the solvability of the local linear system in a special MPFA scheme under the condition that the symmetric part of certain 2×2 matrix is positive definite. Usually, this condition is not satisfied by many highly distorted meshes. By introducing the dynamic continuity point, we are able to discuss rigorously the solvability of our local linear system. Then, the difficulty that arises from the possible singularity of our local linear system is overcome, which makes our algorithm a robust one. More interesting is that the discussion for our local linear system also contributes to the MPFA algorithm, since we notice that there exists certain relation between our linearity preserving nine-point scheme and the physical space derived MPFA.

The rest of this paper is organized as follows. In Section 2, we derive the NPS scheme by the linearity preserving method. In Section 3, we discuss in details the treatments for the vertex unknowns and in Section 4, we give the relations between our

linearity preserving nine-point scheme and some existing schemes, by which a generalized MPFA scheme is suggested. In Section 5, we study rigorously the solvability of the local linear system and locate some of the singular points. The numerical results are presented in Section 6 to show the efficiency of our linearity preserving schemes and some conclusions are given in the last section.

2. The nine-point scheme

For simplicity of exposition, we consider a diffusion equation of the form

$$-\nabla\cdot(\kappa(x,y)\nabla u)=f(x,y),\quad \text{in } \Omega \tag{2.1}$$

with Dirichlet boundary condition

$$u=u_0,\quad \text{on } \partial\Omega \tag{2.2}$$

or Neumann boundary condition

$$-(\kappa(x,y)\nabla u)\cdot\mathbf{n}=u_n,\quad \text{on } \partial\Omega, \tag{2.3}$$

where $\kappa(x,y), f, u_0, u_n$ are given functions, Ω is a bounded polygonal domain in R^2 with its boundary denoted by $\partial\Omega$, and $\mathbf{n}$ denotes the outward unit vector normal to the boundary $\partial\Omega$.

In this section, we shall derive a nine-point scheme for diffusion equation on the distorted structured quadrilateral mesh, which initially has both cell-centered unknowns and vertex unknowns. By certain proper treatment of the vertex unknowns, this scheme reduces to a cell-centered scheme and moreover, has the capability of solving problems either with or without discontinuous diffusion coefficients. The derivation of the nine-point scheme and the treatment of the vertex unknowns are subjected to the so-called *linearity preserving criterion*, which requires that each step of the derivation of a discretization scheme for (2.1) is exact or linearly exact, i.e., exact in the sense whenever the equation possesses a linear solution and the diffusion coefficient $\kappa(x,y)$ is a constant. Throughout, we shall endow the symbol $\simeq$ with a special meaning and assume that

- $=$ is used if the derivation is exact.
- $\simeq$ is used whenever the relevant approximation satisfies the linearity preserving criterion.
- $\approx$, by contrast, will be used when an approximation is not subjected to the linearity preserving criterion.

For example,

$$\int_a^b c(x)\,dx\simeq c\left(\frac{a+b}{2}\right)(b-a),\quad \int_a^b c(x)\,dx\simeq\frac{c(a)+c(b)}{2}(b-a),\quad \int_a^b c(x)\,dx\approx c(a)(b-a).$$

Suppose that Ω is partitioned into a number of nonoverlapping regular polygonal cells, i.e, the intersection of any two intersected cells is either a common edge or a common vertex. Let $A_1A_2\cdots A_k$ be an arbitrary cell with cell center denoted by O_A. The coordinates of O_A are given by

$$x_A=\frac{1}{k}\sum_{i=1}^{k}x_{A_i},\quad y_A=\frac{1}{k}\sum_{i=1}^{k}y_{A_i}, \tag{2.4}$$

where (x_{A_i},y_{A_i}) denote the coordinations of A_i. For each cell, we define an independent cell-centered unknown u_A at the cell center. We also define vertex unknowns, denoted by u_{A_i} and defined at vertex A_i. It will be clear later that these vertex unknowns are intermediate ones and can be represented locally by the cell-centered unknowns. Denote by $\mathbf{n}_A$ the outward unit vector normal to the cell boundary $\partial A_1A_2\cdots A_k$ and S_A the area of the cell $A_1A_2\cdots A_k$. For simplicity, we assume that edge A_3A_4 is on the domain boundary. Assume also that $B_1B_2\cdots B_m$ is another cell next to $A_1A_2\cdots A_k$ with cell center denoted by O_B. The common edge of these two cells is $A_1A_2(B_2B_1)$, see Fig. 1.

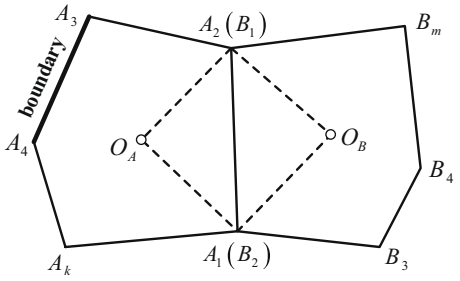


Fig. 1. Definition of mesh and unknowns.

Let the flux vector $\mathbf{F}$ be defined by

$$\mathbf{F} = -\kappa(x,y)\nabla u. \tag{2.5}$$

Obviously, $\mathbf{F}$ will be a constant vector if the solution is a linear function and the diffusion coefficient is a constant. The normal component of the flux on edge A_1A_2 can be expressed as

$$\mathbf{F}\cdot\mathbf{n}_{12} = -\kappa(x,y)\nabla u\cdot\mathbf{n}_{12}, \tag{2.6}$$

where $\mathbf{n}_{12}$ is the restriction of $\mathbf{n}_A$ on edge A_1A_2. On the triangle $\Delta O_A A_1 A_2$, we have [15,34]

$$\mathbf{n}_{12} = \frac{1}{h_A^{(1)}}\left(\alpha_{21}\,\vec{O_AA_1} + \alpha_{12}\,\vec{O_AA_2}\right), \quad \alpha_{ij} = \frac{1}{|A_1A_2|^2}\,\vec{O_AA_i}\cdot\vec{A_jA_i}, \tag{2.7}$$

and $h_A^{(1)}$ denotes the distance from O_A to edge A_1A_2. It follows from (2.6) and (2.7) that

$$\begin{aligned}\mathbf{F}\cdot\mathbf{n}_{12} &= -\frac{\kappa(x,y)}{h_A^{(1)}}\left(\alpha_{21}\nabla u\cdot\vec{O_AA_1} + \alpha_{12}\nabla u\cdot\vec{O_AA_2}\right) \simeq -\frac{\kappa(O_A)}{h_A^{(1)}}\left[\alpha_{21}(u_{A_1}-u_A)+\alpha_{12}(u_{A_2}-u_A)\right]\\ &= -\frac{\kappa(O_A)}{h_A^{(1)}}(\alpha_{21}u_{A_1}+\alpha_{12}u_{A_2}-u_A),\end{aligned} \tag{2.8}$$

where we have used $\alpha_{21}+\alpha_{12}=1$ and here and hereafter, $\kappa(O_A)$ denotes $\kappa(x_{O_A},y_{O_A})$ for simplicity. As a result,

$$-\frac{h_A^{(1)}}{\kappa(O_A)}\int_{A_1A_2}\mathbf{F}\cdot\mathbf{n}_A\,ds \simeq |A_1A_2|(\alpha_{21}u_{A_1}+\alpha_{12}u_{A_2}-u_A). \tag{2.9}$$

Analogously, on triangular domain $\Delta O_B A_2 A_1$, we have

$$-\frac{h_B^{(1)}}{\kappa(O_B)}\int_{A_1A_2}\mathbf{F}\cdot\mathbf{n}_B\,ds \simeq |A_1A_2|(\beta_{21}u_{A_1}+\beta_{12}u_{A_2}-u_B), \tag{2.10}$$

where

$$\beta_{ij} = \frac{1}{|A_1A_2|^2}\,\vec{O_BA_i}\cdot\vec{A_jA_i}\,.$$

Since the flux is continuous across the cell edge A_1A_2, it holds that

$$\int_{A_1A_2}\mathbf{F}\cdot\mathbf{n}_A\,ds = -\int_{A_1A_2}\mathbf{F}\cdot\mathbf{n}_B\,ds. \tag{2.11}$$

Substituting (2.11) into (2.10) leads to

$$-\frac{h_B^{(1)}}{\kappa(O_B)}\int_{A_1A_2}\mathbf{F}\cdot\mathbf{n}_A\,ds \simeq -|A_1A_2|(\beta_{21}u_{A_1}+\beta_{12}u_{A_2}-u_B). \tag{2.12}$$

Now, by summing (2.9) and (2.12), we obtain

$$\int_{A_1A_2}\mathbf{F}\cdot\mathbf{n}_A\,ds \simeq -K_A^{(1)}|A_1A_2|[u_B-u_A-D_A^{(1)}(u_{A_2}-u_{A_1})] := F_A^{(1)}, \tag{2.13}$$

where

$$K_A^{(1)} = \frac{\kappa(O_A)\kappa(O_B)}{\kappa(O_A)h_B^{(1)}+\kappa(O_B)h_A^{(1)}}, \quad D_A^{(1)} = \frac{1}{|A_1A_2|^2}\,\vec{A_1A_2}\cdot\vec{O_AO_B}\,. \tag{2.14}$$

The expressions of the normal component of the flux on other edges in the domain can be derived in a similar way. As for the boundary edge A_3A_4, if Dirichlet boundary condition is imposed, we get, by following the derivation of (2.9),

$$\int_{A_3A_4}\mathbf{F}\cdot\mathbf{n}_A\,ds \simeq -\frac{\kappa(O_A)|A_3A_4|}{h_A^{(3)}}\left\{\frac{\vec{A_3A_4}\cdot\vec{O_AA_4}\,u_{A_3}+\vec{A_4A_3}\cdot\vec{O_AA_3}\,u_{A_4}}{|A_3A_4|^2}-u_A\right\} := F_A^{(3)}. \tag{2.15}$$

Finally, integrating both sides of (2.1) over cell $A_1A_2\cdots A_k$ and using the divergence theorem yield

$$\sum_{i=1}^{k}\int_{A_iA_{i+1}}\mathbf{F}\cdot\mathbf{n}_A\,ds = \int_{A_1A_2\cdots A_k} f(x,y)\,dx\,dy, \tag{2.16}$$

where $A_{k+1}=A_1$. By using the standard point spatial discretization for the source term, we obtain the discrete counterpart of (2.1),

$$\frac{1}{S_A}\sum_{i=1}^{k} F_A^{(i)} = f(x_A, y_A), \tag{2.17}$$

where $F_A^{(i)}$, the integration of the normal components of the flux over certain domain edge (resp. boundary edge), is given in the exact way of (2.13) and (2.14) (resp. (2.15)).

Here we observe that the same (2.17) with $F_A^{(i)}$ given by (2.13) and (2.14) can be obtained in a number of ways, see, e.g., [20,28,30]. The characteristic feature about the present derivation is that it is conducted according to the linearity preserving criterion. Moreover, at the present stage, the above derivation is valid for either structured or unstructured meshes with arbitrary polygonal cells.

We point out that our discretization scheme for (2.1) is developed with coupled hydrodynamics applications in mind, where the structured quadrilateral mesh is often employed. On this type of mesh, a typical stencil of (2.17) with respect to a cell in the domain involves five cell-centered unknowns as well as four vertex unknowns. Since we have both cell-centered unknowns and vertex unknowns while possess only one discrete equation for each cell, the unknowns are more than the discrete equations. A possible way to solve this problem is to find a discrete equation for each vertex unknown. This can be done very efficiently by some special techniques, such as introducing another mesh with respect to the vertices, see [33]. However, methods of this kind often double the computational cost. Another way to solve the problem is to eliminate the vertex unknowns. More explicitly, in our practice, we treat the vertex unknowns as certain linear combination of the surrounding cell-centered unknowns. In this case, the stencil of the nine-point scheme involves nine cell-centered unknowns.

3. The treatments for the vertex unknowns

From now on, we shall confine ourselves to the case where the structured quadrilateral mesh is involved. In this section, we shall mainly discuss the problem of eliminating the vertex unknowns. Besides, we just consider the vertices in the domain where four edges meets at a common vertex. The case where the vertex is on the boundary can be discussed analogously. Usually, the vertex unknown u_0 is treated as a linear combination of the surrounding cell-centered unknowns, namely,

$$u_0 = \sum_{i=1}^{4} w_i u_i, \tag{3.1}$$

where $u_i (1 \leqslant i \leqslant 4)$ are the values of u at the centers of the four surrounding cells, $w_i (1 \leqslant i \leqslant 4)$ are the so-called weights. If the above formula is exact for the constant solution, one immediately gets

$$\sum_{i=1}^{4} w_i = 1. \tag{3.2}$$

3.1. The former treatments for the vertex unknowns

The simplest way to approximate the vertex unknown u_0 is to use the equal weights. In this approach, the weights in (3.1) are given by

$$w_i = \frac{1}{4}, \quad i = 1, 2, 3, 4. \tag{3.3}$$

Generally speaking, the above approximation does not meet the linearity preserving criterion and as a result, the symbol = in (3.1) must be replaced by $\approx$ when weights in (3.3) are used. Another type of weights is given by the formula below

$$w_i = \frac{\kappa(O_i)/d_i}{\sum_{j=1}^{4} \kappa(O_j)/d_j}, \quad i = 1, 2, 3, 4, \tag{3.4}$$

where $d_i (i = 1, 2, 3, 4)$ denote the distance between the cell centers O_j and the vertex. This choice of weights does not satisfy the linearity preserving criterion either. Although the above two approximations lead to poor accuracy on highly distorted meshes, they are often used in practical computation since they are very simple and lead to fairly good accuracy on smooth meshes, see, e.g., [14].

When the diffusion coefficient is continuous, we once suggested in [15,31] an approximation that satisfies the linearity preserving criterion. The key point is to use bilinear interpolation,

$$u_0 \simeq (1-\xi)(1-\eta)u_1 + \xi(1-\eta)u_2 + \xi\eta u_3 + \eta(1-\xi)u_4, \tag{3.5}$$

where ξ and η are determined by the following bilinear mapping

$$\begin{cases} (1-\xi)(1-\eta)x_1 + \xi(1-\eta)x_2 + \xi\eta x_3 + \eta(1-\xi)x_4 = x_0, \\ (1-\xi)(1-\eta)y_1 + \xi(1-\eta)y_2 + \xi\eta y_3 + \eta(1-\xi)y_4 = y_0, \end{cases} \tag{3.6}$$

here $(x_i, y_i)(i = 1, 2, 3, 4)$ and (x_0, y_0) stand for the coordinates of the four cell centers and the vertex being considered, respectively. In the case where the diffusion coefficient is discontinuous, the above approximation must be done with care such that the four cell unknowns is chosen to be located on the same side of the discontinuity line [15,31]. Thus, this method must be used under the condition that the discontinuity line is known beforehand, which decreases its efficiency and prevents it from being used in those cases where certain moving discontinuity such as shock wave occurs. In a recent paper [5], the authors applied this bilinear interpolation approach to the discontinuity case by incorporating the effect of the diffusion coefficient. However, the resulting nine-point scheme has only first order accuracy on distorted meshes.

Recently, a new approximation was suggested in [28], where the weights are obtained by making use of the continuity of the flux and the tangential derivatives across or along the four cell edges. Although this method is designed for both the continuous and discontinuous cases, it is a little complicate and involves a large amount of extra computation. We claim that this method satisfies the linearity preserving criterion. Here we omit the corresponding argument to keep our main point of this paper clear. The interested readers are referred to [28] for details.

There exist some other ways to treat the vertex unknowns, such as by Taylor expansion and minimization of the truncation error [16,30], or by certain adaptive choice of the stencil [7,32]. All these methods are either complicated or being less accurate on distorted meshes. In the rest part of this section, we shall suggest a new approach which results in relatively a small amount of extra computational cost without loss of accuracy.

3.2. A general framework of finding the new weights

To begin with, we introduce some new notations. Suppose that a vertex Q_0 is surrounded by the cells Ω_k with $k = 1, 2, 3, 4$. The cell Ω_k, with cell edges Q_0P_k and Q_0P_{k+1}, has its cell center at O_k. T_k is a dynamic point on edge Q_0P_k (see Fig. 2(a)), defined by

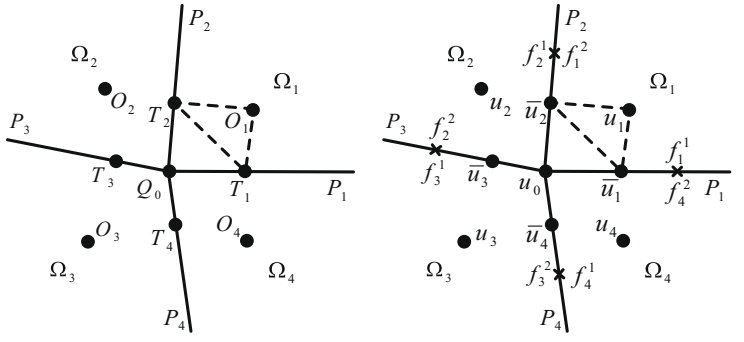
$$T_k = \tau(Q_0, k)Q_0 + [1 - \tau(Q_0, k)]P_k, \quad k = 1, 2, 3, 4, \tag{3.7}$$

where $\tau(Q_0, k)(0 < \tau(Q_0, k) < 1)$ is a dynamic parameter, dependent of Q_0 and the four edges sharing Q_0. In the following discussion, we shall only consider the case where $\tau(Q_0, k) = \tau(Q_0)$, i.e., the dynamic parameter assumes the same value on the four edges having Q_0 in common. Besides, we shall drop (Q_0) and simply use τ for simplicity. Here, we employ a periodic numbering such that $P_5 = P_1, P_4 = P_0, T_4 = T_0$, etc. Denote by u_k the cell-centered unknown at O_k and by u_0 the vertex unknown at Q_0. Let $\mathbf{n}_k$ be the outward unit vector normal to the boundary of cell Ω_k. Denote by f_k^1 and f_k^2 the restriction of $\mathbf{F}\cdot\mathbf{n}_k$ on cell edge Q_0P_k and Q_0P_{k+1}, respectively. Finally, we introduce some more unknowns $\bar{u}_k$, defined at T_k with $k = 1, 2, 3, 4$, see Fig. 2(b).

With these notations, we are ready to describe our method. We see that at the cell corner of Ω_k with respect to the vertex Q_0, there exist five unknowns, including the cell-centered unknown u_k, the newly introduced cell edge unknowns $\bar{u}_k, \bar{u}_{k+1}$ and the flux normal components f_k^1, f_k^2. Assume that these unknowns are related by the following linear system

$$\begin{pmatrix} |Q_0P_k|f_k^1 \\ |Q_0P_{k+1}|f_k^2 \end{pmatrix} = \begin{pmatrix} a_{11}^{(k)}(\tau) & a_{12}^{(k)}(\tau) \\ a_{21}^{(k)}(\tau) & a_{22}^{(k)}(\tau) \end{pmatrix} \begin{pmatrix} \bar{u}_k - u_k \\ \bar{u}_{k+1} - u_k \end{pmatrix} := \mathcal{A}_k(\tau) \begin{pmatrix} \bar{u}_k - u_k \\ \bar{u}_{k+1} - u_k \end{pmatrix}. \tag{3.8}$$

The notation (τ) indicates that the corresponding variable is a function of τ. In the following discussion, we shall drop (τ) for simplicity whenever there is no ambiguity. Here we point out that the fluxes in (3.8) are just intermediate variables, which may be different from the one given in (2.13) and will not be used in our nine-point scheme. The problem of constructing the cell corner matrix $\mathcal{A}_k$ will be discussed in the subsequent subsection. Since the normal component of the flux is continuous across the edge Q_0P_{k+1}, we have



(a) Notations for the points (b) Notations for the unknowns

Fig. 2. The stencil and notations around a mesh vertex.

$$a_{21}^{(k)}(\bar{u}_k - u_k) + a_{22}^{(k)}(\bar{u}_{k+1} - u_k) + a_{11}^{(k+1)}(\bar{u}_{k+1} - u_{k+1}) + a_{12}^{(k+1)}(\bar{u}_{k+2} - u_{k+1}) = 0,$$

which leads to the following linear system

$$\mathcal{M}\overline{U} = \mathcal{N}U, \tag{3.9}$$

where $\overline{U} = (\bar{u}_1, \bar{u}_2, \bar{u}_3, \bar{u}_4)^T, U = (u_1, u_2, u_3, u_4)^T$, and the nonzero entries of the 4 by 4 matrices $\mathcal{M} = (m_{ij})_{4\times4}$ and $\mathcal{N} = (n_{ij})_{4\times4}$ are detailed below:

$$m_{k,k} = a_{11}^{(k)} + a_{22}^{(k-1)}, \quad m_{k,k+1} = a_{12}^{(k)}, \quad m_{k,k-1} = a_{21}^{(k-1)}, \tag{3.10}$$

$$n_{k,k} = a_{11}^{(k)} + a_{12}^{(k)}, \quad n_{k+1,k} = a_{21}^{(k)} + a_{22}^{(k)}, \quad k = 1,2,3,4, \tag{3.11}$$

here the periodic numbering is employed such that $m_{4,5} = m_{4,1}, m_{1,0} = m_{1,4}, a_{21}^{(0)} = a_{21}^{(4)}$, etc. By solving (3.9), we get

$$\overline{U} = \mathcal{M}^{-1}\mathcal{N}U := \mathcal{B}U, \tag{3.12}$$

where $\mathcal{B} = (b_{ij})_{4\times4}$. By (3.12), we are able to evaluate the additional cell edge unknowns by the cell-centered unknowns.

We now come back to the problem of evaluating vertex unknown u_0. At the cell corner of Ω_k with respect to vertex Q_0, we can now construct a linear interpolation based on the unknowns $u_k, \bar{u}_k, \bar{u}_{k+1}$. Noting that vertex Q_0 is located outside of triangle $O_kT_{k+1}T_k$ (see Fig. 2), we get the following formula by extrapolation,

$$u_0 \simeq \lambda_k^1(Q_0)u_k + \lambda_k^2(Q_0)\bar{u}_k + \lambda_k^3(Q_0)\bar{u}_{k+1}, \tag{3.13}$$

where $\lambda_k^i (i = 1,2,3)$ are the area coordinates on triangle $O_kT_{k+1}T_k$ corresponding to O_k, T_k, T_{k+1}, respectively. Substituting (3.12) into (3.13), we get

$$u_0 \simeq \lambda_k^1(Q_0)u_k + \sum_{i=1}^{4}[\lambda_k^2(Q_0)b_{ki} + \lambda_k^3(Q_0)b_{k+1,i}]u_i := \sum_{i=1}^{4} w_i^{(k)}u_i, \tag{3.14}$$

where

$$w_i^{(k)} = \lambda_k^1(Q_0)\delta_{ki} + \lambda_k^2(Q_0)b_{ki} + \lambda_k^3(Q_0)b_{k+1,i}, \tag{3.15}$$

and δ_{ki} denotes the Kronecker delta. Finally, we choose our new weights as

$$w_i = w_i^{(k_0)}, \quad i = 1,2,3,4, \tag{3.16}$$

where k_0 is an integer such that

$$\sum_{i=1}^{4}|w_i^{(k_0)}| = \min_{1\leqslant k\leqslant 4}\sum_{i=1}^{4}|w_i^{(k)}|. \tag{3.17}$$

In practical computation, k_0 may not be unique. In this case, we adopt the following condition to ensure the uniqueness of k_0,

$$\sum_{i=1}^{4}\left(w_i^{(k_0)} - \frac{1}{4}\sum_{j=1}^{4}w_j^{(k_0)}\right)^2 = \min_{1\leqslant k\leqslant 4}\sum_{i=1}^{4}\left(w_i^{(k)} - \frac{1}{4}\sum_{j=1}^{4}w_j^{(k)}\right)^2. \tag{3.18}$$

3.3. The construction of $\mathcal{A}_k$ under linearity preserving criterion

From the derivation in the above subsection, we see that the weights in (3.16) meet the linearity preserving criterion provided that $\mathcal{A}_k$ is constructed in a way that (3.8) holds for linear solutions. Under the linearity preserving criterion, $\mathcal{A}_k$ is uniquely determined. In fact, when the diffusion coefficient $\kappa(x,y) = \kappa(O_k)$ is a constant and the solution is a linear function, we find from (3.8) that

$$-\kappa(O_k)\begin{pmatrix} \mathbf{v}\cdot\mathcal{R}\left(\vec{Q_0P_k}\right) \\ -\mathbf{v}\cdot\mathcal{R}\left(\vec{Q_0P_{k+1}}\right)\end{pmatrix} \simeq \begin{pmatrix} a_{11}^{(k)} & a_{12}^{(k)} \\ a_{21}^{(k)} & a_{22}^{(k)}\end{pmatrix}\begin{pmatrix} \mathbf{v}\cdot\vec{O_kT_k} \\ \mathbf{v}\cdot\vec{O_kT_{k+1}}\end{pmatrix}, \tag{3.19}$$

where $\mathbf{v}$ is an arbitrary constant vector and $\mathcal{R}$ denotes an operator on vectors which rotates a vector clockwise to its normal direction. Solving (3.19) and through a straightforward calculation, we get

$$a_{ij}^{(k)} = \frac{(-1)^{i+j}\kappa(O_k)}{2S_{\Delta O_kT_{k+1}T_k}}\vec{Q_0P_{k+i-1}}\cdot\vec{O_kT_{k-j+2}}, \quad i,j = 1,2, \tag{3.20}$$

where

$$S_{\Delta Q_0 T_{k+1} T_k} = \begin{cases} A_{\Delta Q_0 T_{k+1} T_k}, & Q_0, T_{k+1}, T_k \text{ are ordered anticlockwisely}, \\ -A_{\Delta Q_0 T_{k+1} T_k}, & Q_0, T_{k+1}, T_k \text{ are ordered clockwisely}, \end{cases} \tag{3.21}$$

and $A_{\Delta Q_0 T_{k+1} T_k}$ is the area of triangle $Q_0 T_{k+1} T_k$.

3.4. Implementation procedure

In order to facilitate coding, we give a sketch about the implementation of the linearity preserving nine-point scheme, which consists of two main steps:

- Step 1. Compute the weights for each interior vertex unknown.
  - (i) Compute the corner matrix $\mathcal{A}_k$ by (3.20).
  - (ii) Compute the 4 by 4 matrices $\mathcal{M}, \mathcal{N}$ and $\mathcal{B}$ defined in (3.9) and (3.12), respectively.
  - (iii) Compute $w_i^{(k)} (k = 1, 2, 3, 4)$ by (3.15).
  - (iv) Use (3.17) and (3.18) to determine k_0 and then get the weights by (3.16).
- Step 2. Compute the flux with respect to each cell edge by (2.13).

4. The relations with some existing schemes

We note that the corner matrix $\mathcal{A}_k$ plays an important role in the evaluation of the vertex unknowns in our linearity preserving nine-point scheme (LPNPS). We also observe that this same corner matrix or its variations is used in the derivation of some well-known cell-centered schemes, such as the cell-centered schemes in [6], the MPFA scheme in [1,19]. So it is interesting to describe the relations between LPNPS and some other existing schemes.

First, consider a special case where $\tau = 1$. In this case, T_k and T_{k+1} coincide with Q_0, and the two equations given in (3.8) reduce to one equation, which is the counterpart of formula (2.27) in [28] under linearity preserving criterion. The authors in [28] use their (2.27) and its counterparts with respect to the rest corners to obtain certain weights for (3.1).

Secondly, we can see from (3.20) that $\mathcal{A}_k$ is generally asymmetric. A symmetric $\mathcal{A}_k$ can be found in the case where the mesh consists of uniform parallelograms and $\tau = 0.5$, which is exact the one presented in formula (29) of [6] and obtained by the support operator method. However, the resulting scheme no longer satisfies the linearity preserving criterion and faces a loss of accuracy on highly distorted quadrilateral meshes.

Now, some interesting and natural questions or ideas arise. What happens if the corner matrix in formula (29) of [6] is replaced by (3.8) with $\mathcal{A}_k$ given by (3.20)? Why not use (3.8) directly, instead of (2.13), to compute the normal flux in our scheme? Both ideas lead us to a cell-centered scheme which can be viewed as certain generalized physical space derived MPFA scheme (GMPFA scheme for short). Here, the word 'generalized' means that, the continuity point T_k can be any interior point on the whole edge $Q_0 P_k$.

The implementation of the GMPFA scheme is almost the same as that of a traditional MPFA O-method in [1] except that here the continuity point can be any interior point on the whole cell edge. We observe that the accumulation of the cell edge fluxes is one of the important issues for an MPFA type method. Since each edge has two vertices, the correspondent cell edge flux is computed twice. In order to get a unique method, we must bear in mind that it is the half cell edge fluxes that are computed with respect to each vertex, and by accumulating the two half cell edge fluxes we get the whole flux across each cell edge.

Many authors [4,12,24,25] investigated their MPFA type schemes with a varying τ which is located mainly in $[0.5, 1)$. For example, a finite volume method was studied in [12] with a parameter p belonging to $(0, 0.5]$ while the authors in [24] derived the flux expressions for the general $O(\eta)$-method in the case of homogeneous medium, uniform parallelogram grid and $\eta \in [0, 1)$. The relations $p = 1 - \tau$ and $\eta = 2\tau - 1$ can be easily established by recalling the definitions of τ, p and η. In the GMPFA scheme, τ belongs to $(0, 1)$. Moreover, in the derivation of a MPFA scheme, a dual mesh consisting of dual cells or interaction regions is usually necessary [1], while in the GMPFA scheme, the dual mesh is not needed at all. When deriving the cell edge fluxes with respect to a mesh vertex Q_0, what we need is only the concept of cell corners, defined by $\angle Q_0 P_k P_{k+1}$. Another purpose of the introduction of a dynamic continuity point on the whole cell edge is to facilitate the analysis of the solvability of the local linear systems.

5. The solvability of the local linear systems

It is easy to see that both the construction of linearity preserving scheme in Section 2 and that of the GMPFA scheme depend upon the solvability of the local linear system (3.9), from which we can get an expression of $\overline{U}$. An alternative approach to obtain $\overline{U}$ is to invert (3.8) and use the continuity of edge normal flux $f_k := f_{k-1}^2 = -f_k^1$,

$$\begin{pmatrix} \bar{u}_k - u_k \\ \bar{u}_{k+1} - u_k \end{pmatrix} = \begin{pmatrix} \bar{a}_{11}^{(k)}(\tau) & \bar{a}_{12}^{(k)}(\tau) \\ \bar{a}_{21}^{(k)}(\tau) & \bar{a}_{22}^{(k)}(\tau) \end{pmatrix} \begin{pmatrix} |Q_0 P_k| f_k^1 \\ |Q_0 P_{k+1}| f_k^2 \end{pmatrix} := \overline{\mathcal{A}}_k(\tau) \begin{pmatrix} -|Q_0 P_k| f_k \\ |Q_0 P_{k+1}| f_{k+1} \end{pmatrix}. \tag{5.1}$$

Obviously, $\overline{\mathcal{A}}_k = \mathcal{A}_k^{-1}$ if $\mathcal{A}_k$ is invertible. Eliminating the edge unknowns $\bar{u}_{k+1}$, we have

$$\overline{\mathcal{M}}F = \overline{\mathcal{N}}U, \tag{5.2}$$

where $F = (|Q_0P_1|f_1, |Q_0P_2|f_2, |Q_0P_3|f_3, |Q_0P_4|f_4)^T, U = (u_1, u_2, u_3, u_4)^T$, and the nonzero entries of the 4 by 4 matrices $\overline{\mathcal{M}} = (\bar{m}_{ij})_{4\times4}$ and $\overline{\mathcal{N}} = (\bar{n}_{ij})_{4\times4}$ are as follows:

$$\begin{aligned} &\bar{m}_{k,k} = \bar{a}_{11}^{(k)} + \bar{a}_{22}^{(k-1)}, \quad \bar{m}_{k,k+1} = -\bar{a}_{12}^{(k)}, \quad \bar{m}_{k,k-1} = -\bar{a}_{21}^{(k-1)}, \\ &\bar{n}_{k,k} = 1, \quad \bar{n}_{k+1,k} = -1, \quad k = 1,2,3,4, \end{aligned} \tag{5.3}$$

here we have used once again the periodic numbering technique. Solving (5.2), we get

$$F = \overline{\mathcal{M}}^{-1}\overline{\mathcal{N}}U, \tag{5.4}$$

which expresses the flux vector F in term of the cell-centered unknown vector U. Substituting (5.4) into (5.1), one gets another expression for $\overline{U}$.

Obviously, the invertibility of matrices $\mathcal{M}$ and $\overline{\mathcal{M}}$ is necessary to assure the robustness of our schemes. Besides, since these two matrices, with certain special τ, are also used in some MPFA type schemes and LFMFD type schemes [21], the discussion in this section also contributes to these existing schemes.

If $\mathcal{M}$ is constructed in a way that the corner matrix $\mathcal{A}_k$ is given in [6], then the invertibility of $\mathcal{M}$ is straightforward since $\mathcal{A}_k$ is symmetric positive definite. As mentioned before, the use of this type of $\mathcal{A}_k$ will lead to a loss of accuracy in our schemes, so we do not care about this special case. The authors in [19] obtained the solvability of the local linear system under the condition that

$$-(\overline{\mathcal{A}}_k + \overline{\mathcal{A}}_k^T) \quad \text{with } \tau = \frac{1}{2} \quad \text{is positive definite}, \tag{5.5}$$

see (27) and Lemma A.1 in [19]. As mentioned by some authors, (5.5) is easy to be spoiled and not satisfied by many highly distorted meshes [18]. Here we adopt, throughout this section, a relatively weaker assumption on the mesh, i.e.,

(**A**) All cells in the structured quadrilateral mesh are strictly convex, i.e., each interior angle of the cells in the mesh is less than π.

For simplicity, we go back to Fig. 2 and use the notations therein for exposition. Still the periodic numbering technique and the concept of algebraic area defined in (3.21) are employed. First, by (3.7), we find that

$$2S_{\Delta O_kT_{k+1}T_k} = \left[\tau\vec{O_kQ_0} + (1-\tau)\vec{O_kP_{k+1}}\right] \cdot \left[\tau\mathcal{R}\left(\vec{O_kQ_0}\right) + (1-\tau)\mathcal{R}\left(\vec{O_kP_k}\right)\right] = 2(1-\tau)(\tau S_{\Delta Q_0P_kP_{k+1}} + S_{\Delta O_kP_{k+1}P_k}), \tag{5.6}$$

where we have used

$$S_{\Delta O_kQ_0P_k} + S_{\Delta O_kP_{k+1}Q_0} - S_{\Delta O_kP_{k+1}P_k} = S_{\Delta Q_0P_kP_{k+1}}.$$

Since the cell is a strictly convex one, we can define

$$\tau_k^\star = -\frac{S_{\Delta O_kP_{k+1}P_k}}{S_{\Delta Q_0P_kP_{k+1}}}. \tag{5.7}$$

It follows from (5.6) that

$$S_{\Delta O_kT_{k+1}T_k} = S_{\Delta Q_0P_kP_{k+1}}(1-\tau)(\tau - \tau_k^\star),$$

by which we can rewritten the entries of $\mathcal{A}_k$, given by (3.20), in the form below

$$a_{ij}^{(k)} = \frac{\alpha_{ij}^{(k)}(\tau)}{(\tau-1)(\tau-\tau_k^\star)}, \quad i,j = 1,2, \tag{5.8}$$

where

$$\alpha_{ij}^{(k)}(\tau) = \frac{(-1)^{i+j-1}\kappa(O_k)}{2S_{\Delta Q_0P_kP_{k+1}}}\vec{Q_0P_{k+i-1}} \cdot \vec{O_kT_{k-j+2}} \tag{5.9}$$

is a linear function of τ. Now we deduce from (3.10) and (5.8) that

$$\mathrm{Det}\,\mathcal{M} = \frac{\mathrm{Det}\,\mathcal{M}_1(\tau)}{(\tau-1)^4\prod_{k=1}^4(\tau-\tau_k^\star)^2}, \tag{5.10}$$

where

$$\mathcal{M}_1(\tau)=\begin{pmatrix} \beta_1(\tau) & \alpha_{12}^{(1)}(\tau-\tau_4^\star) & 0 & \alpha_{21}^{(4)}(\tau-\tau_1^\star)\\ \alpha_{21}^{(1)}(\tau-\tau_2^\star) & \beta_2(\tau) & \alpha_{12}^{(2)}(\tau-\tau_1^\star) & 0\\ 0 & \alpha_{21}^{(2)}(\tau-\tau_3^\star) & \beta_3(\tau) & \alpha_{12}^{(3)}(\tau-\tau_2^\star)\\ \alpha_{12}^{(4)}(\tau-\tau_3^\star) & 0 & \alpha_{21}^{(3)}(\tau-\tau_4^\star) & \beta_4(\tau) \end{pmatrix}, \tag{5.11}$$

and

$$\beta_k(\tau)=\alpha_{11}^{(k)}(\tau)(\tau-\tau_{k-1}^\star)+\alpha_{22}^{(k-1)}(\tau)(\tau-\tau_k^\star),\quad k=1,2,3,4.$$

The result below follows immediately.

Lemma 5.1. *Let $\mathcal{M}_1(\tau)$ be defined by (5.11) and (5.9), then*

(1) $\mathrm{Det}\,\mathcal{M}_1(\tau)$ *is a polynomial of τ with a degree not greater than 8;*
(2) $\mathcal{M}$ *is invertible if and only if τ is not a root of the following polynomial of degree not greater than 13,*

$$\mathcal{F}_{13}(\tau)=\mathrm{Det}\mathcal{M}_1(\tau)(\tau-1)\prod_{k=1}^{4}(\tau-\tau_k^\star).$$

By the above lemma, the distribution of the roots of $\mathcal{F}_{13}(\tau)$ is very important for the choice of τ in our schemes. To this end, we first determine the location of $\tau_k^\star$. If the algebraic area $S_{\Delta O_k P_{k+1} P_k}$ is positive and by (5.7), $\tau_k^\star$ is negative. On the contrary, if $S_{\Delta O_k P_{k+1} P_k}$ is non-positive and still by (5.7),

$$0\leqslant\tau_k^\star=-\frac{S_{\Delta O_k P_{k+1} P_k}}{S_{\Delta Q_0 P_k P_{k+1}}}=\frac{S_{\Delta O_k P_k P_{k+1}}}{S_{\Delta Q_0 P_k P_{k+1}}}<\frac{1}{2}.$$

Summarizing the above discussion, we arrive at

$$\tau_k^\star<\frac{1}{2},\quad k=1,2,3,4. \tag{5.12}$$

It is easy to see that the case where $\tau=1$ corresponds to $T_k=T_{k+1}=Q_0$. We can also understand the root $\tau_k^\star$ through a geometry approach. When $\tau=\tau_k^\star$, T_k and T_{k+1} are collinear with the cell center O_k, see Fig. 3. The distribution of the known roots of $\mathcal{F}_{13}(\tau)$ and the usual choices of τ are shown in Fig. 4.

Further investigation of the roots of $\mathcal{F}_{13}(\tau)$ can be done by proving

$$\mathrm{Det}\,\mathcal{M}_1(\tau)=\mathcal{F}_3(\tau)(\tau-1)\prod_{k=1}^{4}(\tau-\tau_k^\star),$$

where $\mathcal{F}_3(\tau)$ is a polynomial of τ with a degree not greater than 3. Since this approach involves tedious details, here we turn to another approach by investigating the invertibility of matrix $\overline{\mathcal{M}}$. Firstly, we have the result below.

Lemma 5.2. *Assume that the parameter $\tau\in(0,1)$ and $\tau\neq\tau_k^\star(k=1,2,3,4)$, defined in (5.7). Then* $\mathrm{Det}\overline{\mathcal{M}}(\tau)\neq0$ implies $\mathrm{Det}\mathcal{M}(\tau)\neq0$.

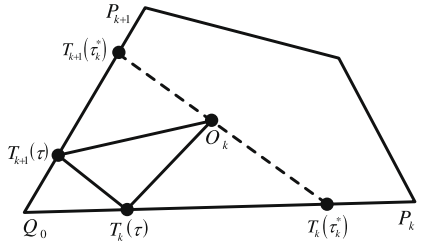


Fig. 3. Some notations.

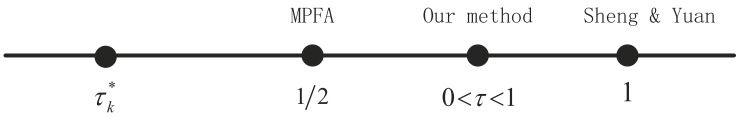


Fig. 4. MPFA: Ref. [1]; Sheng and Yuan: Ref. [28].

Proof. By (5.6) and (5.7), $\mathcal{A}_k$ and $\overline{\mathcal{A}}_k$ are invertible when $\tau \in (0,1)$ and $\tau \neq \tau_k^\star (k = 1,2,3,4)$. Now, given $U = 0$, from (5.2), we have $F = 0$ by the assumption $\mathrm{Det}\,\overline{\mathcal{M}}(\tau) \neq 0$, which leads to $\overline{U} = 0$ by (3.8) or (5.1). On the other hand, the assumptions in this lemma and (3.10) guarantee that the entries of $\mathcal{M}$ are all finite. If $\mathrm{Det}\mathcal{M}(\tau) = 0$, then we are able to get a nonzero vector $\overline{U}$ when $U = 0$, which is a contradiction and completes the proof. □

Lemma 5.3. *Let $\overline{\mathcal{M}}(\tau)$ be defined by (5.3) and (5.1). Then*

$$\mathrm{Det}\,\overline{\mathcal{M}}(\tau) = (\tau - 1)\overline{\mathcal{F}}_3(\tau), \tag{5.13}$$

where $\overline{\mathcal{F}}_3(\tau)$ is a polynomial of τ with degree at most 3.

Proof. By comparing (5.1) with (3.8) where $\mathcal{A}_k$ was defined in (3.20), we find

$$\bar{a}_{ij}^{(k)}(\tau) = \frac{1}{2S_{\Delta Q_0 P_k P_{k+1}}\kappa(O_k)} \vec{O_k T}_{k+i-1} \cdot \vec{Q_0 P}_{k+2-j}, \quad i,j = 1,2, \tag{5.14}$$

which are linear functions of τ due to the definition of T_k in (3.7). It follows that

$$\bar{a}_{1j}^{(k)}(\tau) - \bar{a}_{2j}^{(k)}(\tau) = \frac{1}{2S_{\Delta Q_0 P_k P_{k+1}}\kappa(O_k)} \left(\vec{O_k T}_k - \vec{O_k T}_{k+1} \right) \cdot \vec{Q_0 P}_{k+2-j} = \frac{(\tau - 1)\vec{P_k P}_{k+1} \cdot \vec{Q_0 P}_{k+2-j}}{2S_{\Delta Q_0 P_k P_{k+1}}\kappa(O_k)}, \quad j = 1,2.$$

Then,

$$-\bar{a}_{12}^{(k)}(\tau) + \bar{a}_{22}^{(k)}(\tau) + \bar{a}_{11}^{(k+1)}(\tau) - \bar{a}_{21}^{(k+1)}(\tau) = (\tau - 1)\sigma_k,$$

where

$$\sigma_k = -\frac{\vec{P_k P}_{k+1} \cdot \vec{Q_0 P}_k}{2S_{\Delta Q_0 P_k P_{k+1}}\kappa(O_k)} + \frac{\vec{P_{k+1} P}_{k+2} \cdot \vec{Q_0 P}_{k+2}}{2S_{\Delta Q_0 P_{k+1} P_{k+2}}\kappa(O_{k+1})}.$$

Summing the first three rows of $\overline{\mathcal{M}}(\tau)$ to its last row and taking out the common factor $\tau - 1$, we reach

$$\mathrm{Det}\,\overline{\mathcal{M}}(\tau) = (\tau - 1)\mathrm{Det}\,\overline{\mathcal{M}}_1(\tau)$$

with $\overline{\mathcal{M}}_1(\tau)$ given by

$$\overline{\mathcal{M}}_1(\tau) = \begin{pmatrix} \bar{a}_{11}^{(1)}(\tau) + \bar{a}_{22}^{(4)}(\tau) & -\bar{a}_{12}^{(1)}(\tau) & 0 & -\bar{a}_{21}^{(4)}(\tau) \\ -\bar{a}_{21}^{(1)}(\tau) & \bar{a}_{11}^{(2)}(\tau) + \bar{a}_{22}^{(1)}(\tau) & -\bar{a}_{12}^{(2)}(\tau) & 0 \\ 0 & -\bar{a}_{21}^{(2)}(\tau) & \bar{a}_{11}^{(3)}(\tau) + \bar{a}_{22}^{(2)}(\tau) & -\bar{a}_{12}^{(3)}(\tau) \\ \sigma_4 & \sigma_1 & \sigma_2 & \sigma_3 \end{pmatrix}.$$

By noting the fact that all $\bar{a}_{ij}^{(k)}(\tau)$ are linear functions of τ we get (5.13) and complete the proof. □

The following result is straightforward.

Lemma 5.4. *Assume that the coefficient $\kappa(x,y)$ is a constant κ_0, then under the assumption (**A**), there holds the decomposition*

$$\bar{a}_{ij}^{(k)}(\tau) = \frac{1}{\kappa_0}(-\bar{b}_{ij}^{(k)}\tau + \bar{c}_{ij}^{(k)}), \quad i,j = 1,2, \ k = 1,2,3,4, \tag{5.15}$$

where

$$\bar{b}_{ij}^{(k)} = \frac{\vec{Q_0 P}_{k+i-1} \cdot \vec{Q_0 P}_{k+2-j}}{2S_{\Delta Q_0 P_k P_{k+1}}}, \quad \bar{c}_{ij}^{(k)} = \frac{\vec{O_k P}_{k+i-1} \cdot \vec{Q_0 P}_{k+2-j}}{2S_{\Delta Q_0 P_k P_{k+1}}}. \tag{5.16}$$

Moreover,

$$\frac{\bar{b}_{11}^{(k+1)} + \bar{b}_{22}^{(k)}}{\bar{b}_{12}^{(k+1)}\bar{b}_{12}^{(k)}} = \frac{|Q_0 P_{k+2}|}{|Q_0 P_k|}\sin(\theta_k + \theta_{k+1}), \quad \frac{\bar{b}_{11}^{(k+1)} + \bar{b}_{22}^{(k)}}{\bar{b}_{21}^{(k+1)}\bar{b}_{21}^{(k)}} = \frac{|Q_0 P_k|}{|Q_0 P_{k+2}|}\sin(\theta_k + \theta_{k+1}), \tag{5.17}$$

where $\theta_k = \angle P_k Q_0 P_{k+1}$.

Lemma 5.5. *Under the same assumptions of Lemma 5.4, there exist two constants α and β, independent of τ, such that*

$$\mathrm{Det}\,\overline{\mathcal{M}}(\tau) = (\tau - 1)(\alpha\tau + \beta). \tag{5.18}$$

Proof. Set

$$\alpha_1 = \frac{\bar{b}_{11}^{(2)} + \bar{b}_{22}^{(1)}}{\bar{b}_{12}^{(2)}}, \quad \beta_1 = -\frac{\bar{b}_{12}^{(1)}}{\bar{b}_{21}^{(4)}}.$$

By (5.16) and (5.17), we find that

$$-\alpha_1 \bar{b}_{21}^{(3)} + \beta_1(\bar{b}_{11}^{(4)} + \bar{b}_{22}^{(3)}) = 0, \quad \alpha_1(\bar{b}_{11}^{(3)} + \bar{b}_{22}^{(2)}) - \beta_1 \bar{b}_{12}^{(3)} = \bar{b}_{21}^{(2)},$$

where we have used

$$\theta_1 + \theta_2 + \theta_3 + \theta_4 = 2\pi.$$

Multiplying respectively the last two columns of $\overline{\mathcal{M}}$ with α_1 and β_1, adding the results to the second column and through some straightforward calculations, we reach

$$\mathrm{Det}\,\overline{\mathcal{M}} = \frac{1}{\kappa_0}\begin{vmatrix} \bar{a}_{11}^{(1)} + \bar{a}_{22}^{(4)} & -\bar{c}_{12}^{(1)} - \beta_1 \bar{c}_{21}^{(4)} & 0 & -\bar{a}_{21}^{(4)} \\ -\bar{a}_{21}^{(1)} & \bar{c}_{11}^{(2)} + \bar{c}_{22}^{(1)} - \alpha_1 \bar{c}_{12}^{(2)} & -\bar{a}_{12}^{(2)} & 0 \\ 0 & -\bar{c}_{21}^{(2)} + \alpha_1(\bar{c}_{11}^{(3)} + \bar{c}_{22}^{(2)}) - \beta_1 \bar{c}_{12}^{(3)} & \bar{a}_{11}^{(3)} + \bar{a}_{22}^{(2)} & -\bar{a}_{12}^{(3)} \\ -\bar{a}_{12}^{(4)} & -\alpha_1 \bar{c}_{21}^{(3)} + \beta_1(\bar{c}_{11}^{(4)} + \bar{c}_{22}^{(3)}) & -\bar{a}_{21}^{(3)} & \bar{a}_{11}^{(4)} + \bar{a}_{22}^{(3)} \end{vmatrix}. \tag{5.19}$$

Now, set

$$\alpha_2 = -\frac{\bar{b}_{21}^{(1)}}{\bar{b}_{12}^{(2)}}, \quad \beta_2 = \frac{\bar{b}_{11}^{(1)} + \bar{b}_{22}^{(4)}}{\bar{b}_{21}^{(4)}}.$$

Performing a derivation similar to that of (5.19), we conclude that

$$\mathrm{Det}\,\overline{\mathcal{M}} = \frac{1}{\kappa_0^2}\begin{vmatrix} \bar{c}_{11}^{(1)} + \bar{c}_{22}^{(4)} - \beta_2 \bar{c}_{21}^{(4)} & -\bar{c}_{12}^{(1)} - \beta_1 \bar{c}_{21}^{(4)} & 0 & -\bar{a}_{21}^{(4)} \\ -\bar{c}_{21}^{(1)} - \alpha_2 \bar{c}_{12}^{(2)} & \bar{c}_{11}^{(2)} + \bar{c}_{22}^{(1)} - \alpha_1 \bar{c}_{12}^{(2)} & -\bar{a}_{12}^{(2)} & 0 \\ \alpha_2(\bar{c}_{11}^{(3)} + \bar{c}_{22}^{(2)}) - \beta_2 \bar{c}_{12}^{(3)} & d_1 & \bar{a}_{11}^{(3)} + \bar{a}_{22}^{(2)} & -\bar{a}_{12}^{(3)} \\ d_2 & -\alpha_1 \bar{c}_{21}^{(3)} + \beta_1(\bar{c}_{11}^{(4)} + \bar{c}_{22}^{(3)}) & -\bar{a}_{21}^{(3)} & \bar{a}_{11}^{(4)} + \bar{a}_{22}^{(3)} \end{vmatrix},$$

where

$$d_1 = -\bar{c}_{21}^{(2)} + \alpha_1\left(\bar{c}_{11}^{(3)} + \bar{c}_{22}^{(2)}\right) - \beta_1 \bar{c}_{12}^{(3)}, \quad d_2 = -\bar{c}_{12}^{(4)} - \alpha_2 \bar{c}_{21}^{(3)} + \beta_2(\bar{c}_{11}^{(4)} + \bar{c}_{22}^{(3)}).$$

Noting that $\alpha_1, \alpha_2, \beta_1, \beta_2, \bar{b}_{ij}^{(k)}$ and $\bar{c}_{ij}^{(k)}$ are all constant, independent of τ, and moreover, $\bar{a}_{ij}^{(k)}$ has the decomposition (5.15), we arrive at (5.18) by using Lemma 5.3 and choosing

$$\alpha = \frac{1}{\kappa_0^4}\begin{vmatrix} \bar{c}_{11}^{(1)} + \bar{c}_{22}^{(4)} - \beta_2 \bar{c}_{21}^{(4)} & -\bar{c}_{12}^{(1)} - \beta_1 \bar{c}_{21}^{(4)} & 0 & -\bar{b}_{21}^{(4)} \\ -\bar{c}_{21}^{(1)} - \alpha_2 \bar{c}_{12}^{(2)} & \bar{c}_{11}^{(2)} + \bar{c}_{22}^{(1)} - \alpha_1 \bar{c}_{12}^{(2)} & -\bar{b}_{12}^{(2)} & 0 \\ \alpha_2(\bar{c}_{11}^{(3)} + \bar{c}_{22}^{(2)}) - \beta_2 \bar{c}_{12}^{(3)} & d_1 & \bar{b}_{11}^{(3)} + \bar{b}_{22}^{(2)} & -\bar{b}_{12}^{(3)} \\ d_2 & -\alpha_1 \bar{c}_{21}^{(3)} + \beta_1(\bar{c}_{11}^{(4)} + \bar{c}_{22}^{(3)}) & -\bar{b}_{21}^{(3)} & \bar{b}_{11}^{(4)} + \bar{b}_{22}^{(3)} \end{vmatrix},$$

and

$$\beta = -\frac{1}{\kappa_0^4}\begin{vmatrix} \bar{c}_{11}^{(1)} + \bar{c}_{22}^{(4)} & -\bar{c}_{12}^{(1)} & 0 & -\bar{c}_{21}^{(4)} \\ -\bar{c}_{21}^{(1)} & \bar{c}_{11}^{(2)} + \bar{c}_{22}^{(1)} & -\bar{c}_{12}^{(2)} & 0 \\ 0 & -\bar{c}_{21}^{(2)} & \bar{c}_{11}^{(3)} + \bar{c}_{22}^{(2)} & -\bar{c}_{12}^{(3)} \\ -\bar{c}_{12}^{(4)} & 0 & -\bar{c}_{21}^{(3)} & \bar{c}_{11}^{(4)} + \bar{c}_{22}^{(3)} \end{vmatrix}. \quad \square$$

The main results of this section are summarized in the following theorem.

Theorem 5.1. $\mathrm{Det}\overline{\mathcal{M}}(\tau)$ *has at most three roots in (0, 1) and,* $\mathrm{Det}\,\mathcal{M}(\tau)$ *has at most three roots that are different from* $\tau_k^\star (k = 1, 2, 3, 4)$ *and belong to (0,1). Moreover, under the same assumptions of* Lemma 5.4, $\mathrm{Det}\overline{\mathcal{M}}(\tau)$ *has at most one zero in (0,1).*

Proof. The proof of the first part follows immediately from Lemma 5.3. For $\tau \in (0,1)$ and $\tau \neq \tau_k^\star (k = 1, 2, 3, 4)$, $\mathrm{Det}\,\mathcal{M}(\tau) = 0$ implies $\mathrm{Det}\overline{\mathcal{M}}(\tau) = 0$ by Lemma 5.2. The last result follows directly from Lemma 5.5, which concludes the proof. $\square$

Based on Theorem 5.1, we obtain a robust algorithm for the solution of linear system (3.9) or (5.2), which is necessary for our new nine-point scheme and certain physical space derived MPFA schemes, such as the O-method in [1]. Specifically, we first choose $\tau = 1/2$. If $\mathcal{M}$ or $\overline{\mathcal{M}}$ is singular, then we choose, arbitrarily, a different value in (0, 1), for example, $\tau = 3/4$ or

$\tau = 2/5$. Theoretically speaking, the nonsingular value can be obtained by sampling at most eight (resp. four) times for linear system (3.9) (resp. (5.2)).

6. Numerical examples

In this section, we present some numerical results to demonstrate the accuracy and the efficiency of the linearity preserving nine-point schemes. The notations of the schemes that are considered in this section are given in Table 1. We point out once again that, the generalized MPFA scheme with $\tau = 1/2$ is identical to the physical space derived MPFA O-method, originally suggested in [1] and analyzed in [18,19]. The later is derived from a hypotheses of linear variation in u and accordingly, is also derived from the linearity preserving method.

Example 6.1. Solve the problem

$$-\Delta u = (6 + 4x^2 + 16y^2)e^{x^2+2y^2}, \quad \text{in } \Omega = [0,1] \times [0,1] \tag{6.1}$$

with Dirichlet boundary condition $u(x,y) = -e^{x^2+2y^2}$ on $\partial\Omega$. The exact solution is $u(x,y) = -e^{x^2+2y^2}$.

The solution errors and edge normal flux errors are investigated in the discrete L_2 norms, which are defined by [4]

$$E_u(h) = \left(\sum_i S_i(u_{h,i} - u_i)^2\right)^{1/2},$$
$$E_q(h) = \left(\sum_j Q_j(q_{h,j} - q_j)^2 \Big/ \sum_j Q_j\right)^{1/2}.$$

Here S_i is the area of mesh cell i, Q_j is the volume associated with edge j and equal to the sum of the area of the two cells separated by the edge j. Further, $q = -\kappa\nabla u \cdot \boldsymbol{n}$ is the edge normal flux density. Subscript h refers to the discrete solution. The analytical solution u_i is computed at the cell centers, whereas the analytical cell edge flux q_j is evaluated by the midpoint rule.

We first use the uniform trapezoidal mesh shown in Fig. 5 to find out the convergence rates of the above nine schemes. The L_2 errors of the solution u and edge normal flux are presented in Tables 2 and 3 respectively, where in the last columns, the convergence rates of the solution and the flux are presented, which are obtained by a least square fit on the ones computed on each two successive meshes by the following formula

$$\frac{\log[E_\alpha(h_2)/E_\alpha(h_1)]}{\log(h_2/h_1)}, \quad \alpha = u, q,$$

where h_1, h_2 denote the mesh sizes of the two successive meshes, and $E_\alpha(h_1), E_\alpha(h_2)$ the corresponding L_2 errors of the solution or the flux. The results in Tables 2 and 3 show that the convergence rates of the four linearity preserving schemes (LPNPS1, LPNPS2, LPNPS-SY and GMPFA) are all of the second order for the solution and approximately $O(h^{3/2})$ for the flux, the same as those of MPFA, while NPS1 and NPS2 seem to have no convergence rate in this special case. CCDS has a convergence rate of $O(h^{1/2})$ for the solution and even lower convergence rate for the flux. As for LSOM, it has a second order convergence rate for the solution and first order convergence for the flux. Moreover, according to our experience, the computational costs of the weights in LPNPS2 is approximately one third of that in LPNPS-SY, which implies that, by the use of proper technique, it is possible for us to get much better results than those of the old nine-point schemes and at the same time, to keep the extra computational costs down to a reasonable level with much enhanced robustness.

Now we begin to investigate the performance of our linearity preserving schemes on some highly skewed and highly distorted meshes. We note that there exist in the literature several typical meshes such as the random mesh, the Shestakov mesh, the Kershaw mesh [17] and so on. As is done by many authors, we shall use these meshes to test our schemes. The description of the mesh and analysis of the corresponding numerical results (see Table 4) are given as follows:

Table 1
The notations for the schemes.

Notation	Description
NPS1	The nine-point scheme in Section 2 with equal weights
NPS2	The nine-point scheme in Section 2 with weights given by (3.4)
LPNPS1	The Linearity preserving scheme in Section 2 with weights given by (3.5)
LPNPS2	The Linearity preserving scheme in Section 2 with weights given by (3.16)
LPNPS-SY	The Linearity preserving scheme in section 2 with weights given in [28]
GMPFA	The generalized MPFA scheme in Section 3 with $\tau = 2/5$
MPFA	The MPFA scheme suggested in [1] and analyzed in [18,19]
CCDS	The cell-centered diffusion scheme in [6]
LSOM	The local support operator scheme in [23]

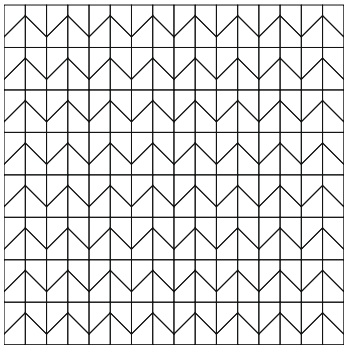


Fig. 5. Trapezoidal mesh.

Table 2
Comparison of solution errors on the uniform trapezoidal mesh.

N	4	8	16	32	64	R_u
NPS1	5.14×10^{-1}	4.01×10^{-1}	3.84×10^{-1}	3.81×10^{-1}	3.80×10^{-1}	0.109
NPS2	4.31×10^{-1}	2.97×10^{-1}	2.82×10^{-1}	2.81×10^{-1}	2.81×10^{-1}	0.154
LPNPS1	3.11×10^{-1}	8.96×10^{-2}	2.43×10^{-2}	6.19×10^{-3}	1.54×10^{-3}	1.914
LPNPS2	3.11×10^{-1}	8.96×10^{-2}	2.43×10^{-2}	6.20×10^{-3}	1.55×10^{-3}	1.913
LPNPS-SY	3.11×10^{-1}	8.96×10^{-2}	2.43×10^{-2}	6.19×10^{-3}	1.54×10^{-3}	1.914
GMPFA	2.07×10^{-1}	8.18×10^{-2}	2.35×10^{-2}	6.16×10^{-3}	1.56×10^{-3}	1.762
MPFA	3.20×10^{-1}	1.05×10^{-1}	2.85×10^{-2}	7.33×10^{-3}	1.85×10^{-3}	1.859
CCDS	1.34×10^{-1}	1.38×10^{-1}	9.48×10^{-2}	5.40×10^{-2}	2.86×10^{-2}	0.556
LSOM	5.28×10^{-1}	1.43×10^{-1}	3.62×10^{-2}	9.06×10^{-3}	2.27×10^{-3}	1.966

Table 3
Comparison of the flux errors on the uniform trapezoidal mesh.

N	4	8	16	32	64	R_q
NPS1	4.66×10^{0}	2.64×10^{0}	2.51×10^{0}	2.59×10^{0}	2.62×10^{0}	0.207
NPS2	4.54×10^{0}	2.15×10^{0}	1.80×10^{0}	1.84×10^{0}	1.87×10^{0}	0.320
LPNPS1	4.49×10^{0}	1.65×10^{0}	5.33×10^{-1}	1.65×10^{-1}	5.11×10^{-2}	1.615
LPNPS2	4.49×10^{0}	1.65×10^{0}	5.33×10^{-1}	1.65×10^{-1}	5.12×10^{-2}	1.614
LPNPS-SY	4.49×10^{0}	1.65×10^{0}	5.33×10^{-1}	1.65×10^{-1}	5.11×10^{-2}	1.615
GMPFA	2.73×10^{0}	9.15×10^{0}	2.83×10^{-1}	8.86×10^{-2}	3.09×10^{-2}	1.617
MPFA	2.33×10^{0}	7.40×10^{-1}	2.25×10^{-1}	7.17×10^{-2}	2.66×10^{-2}	1.613
CCDS	1.37×10^{0}	9.95×10^{-1}	8.40×10^{-1}	6.41×10^{-1}	4.63×10^{-1}	0.390
LSOM	1.49×10^{0}	6.91×10^{-1}	3.17×10^{-1}	1.36×10^{-1}	5.90×10^{-2}	1.166

- The random mesh is obtained by randomly disturbing the interior mesh vertices of the uniform mesh, see Fig. 6. The convergence rates of nearly all schemes drop a little, compared with those on the uniform trapezoidal mesh. Again, the four linearity preserving schemes have almost the same convergence rates as those of MPFA.
- A typical Shestakov mesh is shown in Fig. 7. The numerical results relevant to a sequence of Shestakov meshes are presented in the forth and fifth rows of Table 4. The performance of the linearity preserving schemes on this type of mesh is similar to that of MPFA.
- A Kershaw mesh is shown in Fig. 8. Structured quadrilateral mesh of this type, first proposed by Kershaw [17], is used by many authors to test their methods. One can see that this kind of mesh has a large distortion in vertical direction. It is interesting to note that, on this special type of distorted mesh, MPFA, GMPFA and CCDS perform much better than the rest six schemes.

Now, we investigate the performance of our schemes in a more interesting case where the diffusion coefficient is discontinuous. The following test problem is rebuilt from the last test example in [27].

Example 6.2. Solve the problem (2.1) with Dirichlet boundary condition (2.2) and $\Omega = [0,1] \times [0,1]$. The diffusion coefficient is discontinuous, i.e., $\kappa(x,y) = 1$ for $0 < x \leqslant 0.5$ and k for $0.5 < x < 1$. A solution that has a discontinuous tangential flux at the interface $x = 0.5$ is

Table 4
Convergence rates for the solution and the flux on distorted meshes.

Scheme	Random mesh		Shestakov mesh		Kershaw mesh	
	R_u	R_q	R_u	R_q	R_u	R_q
NPS1	−0.060	0.109	0.558	0.118	0.662	0.457
NPS2	0.209	0.361	0.872	0.419	0.954	0.763
LPNPS1	1.621	1.512	2.397	1.730	1.530	1.484
LPNPS2	1.631	1.503	2.372	1.724	1.701	1.472
LPNPS-SY	1.600	1.505	2.387	1.731	1.473	1.560
GMPFA	1.782	1.431	2.484	1.799	1.970	1.897
MPFA	1.848	1.420	2.630	1.842	1.961	1.916
CCDS	0.613	0.503	0.872	0.659	1.916	1.886
LSOM	1.834	0.993	2.552	1.473	1.459	1.313

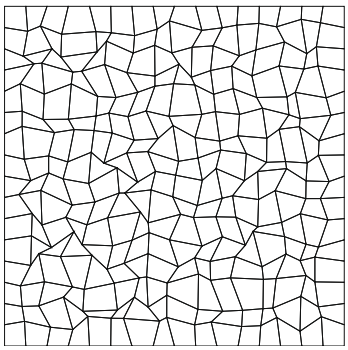


Fig. 6. Random mesh.

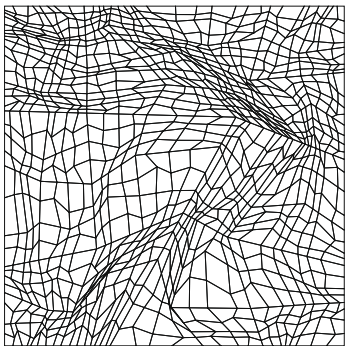


Fig. 7. Shestakov mesh.

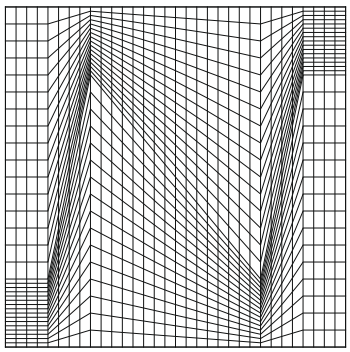


Fig. 8. Kershaw mesh.

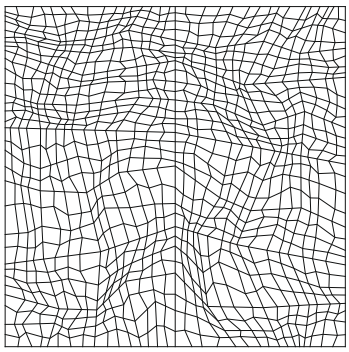
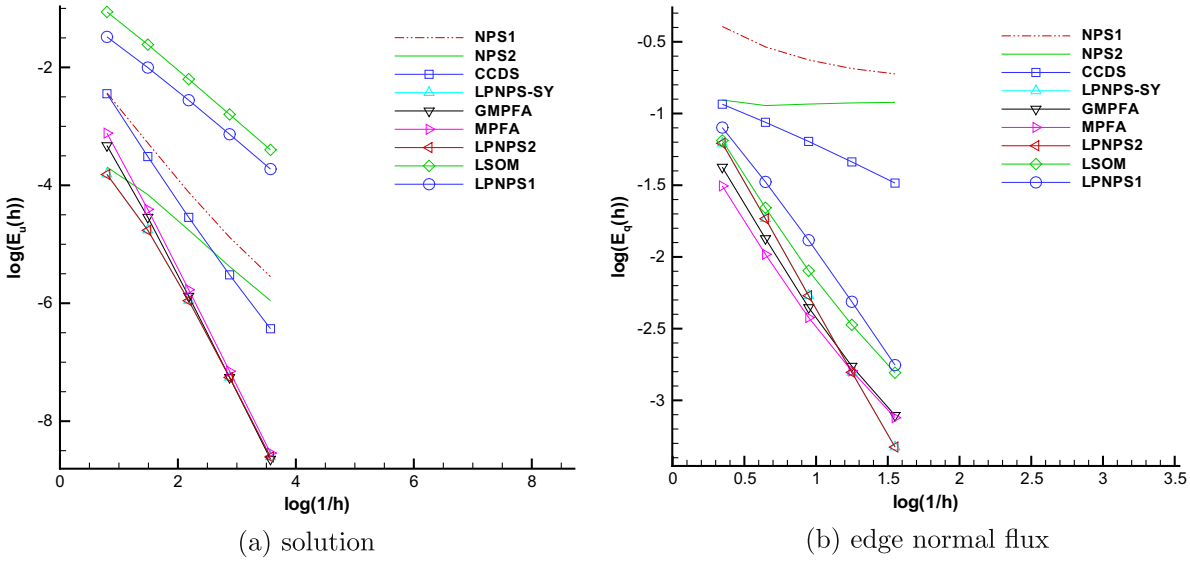
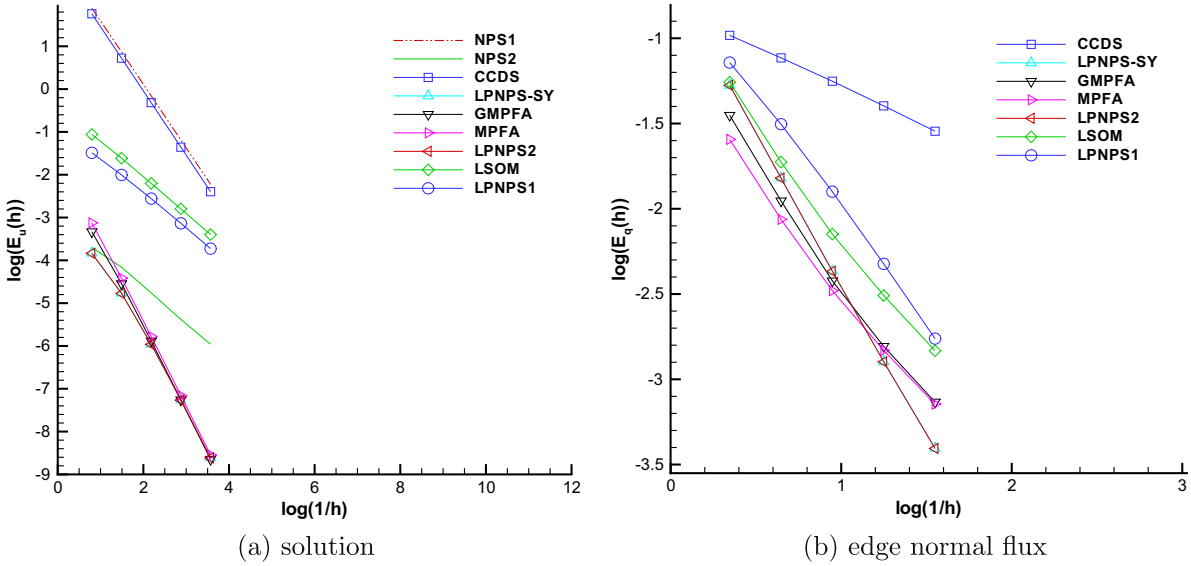


Fig. 9. Modified Shestakov mesh.



(a) solution (b) edge normal flux

Fig. 10. Convergence behavior on the uniform trapezoidal mesh ($k = 10^{-1}$).

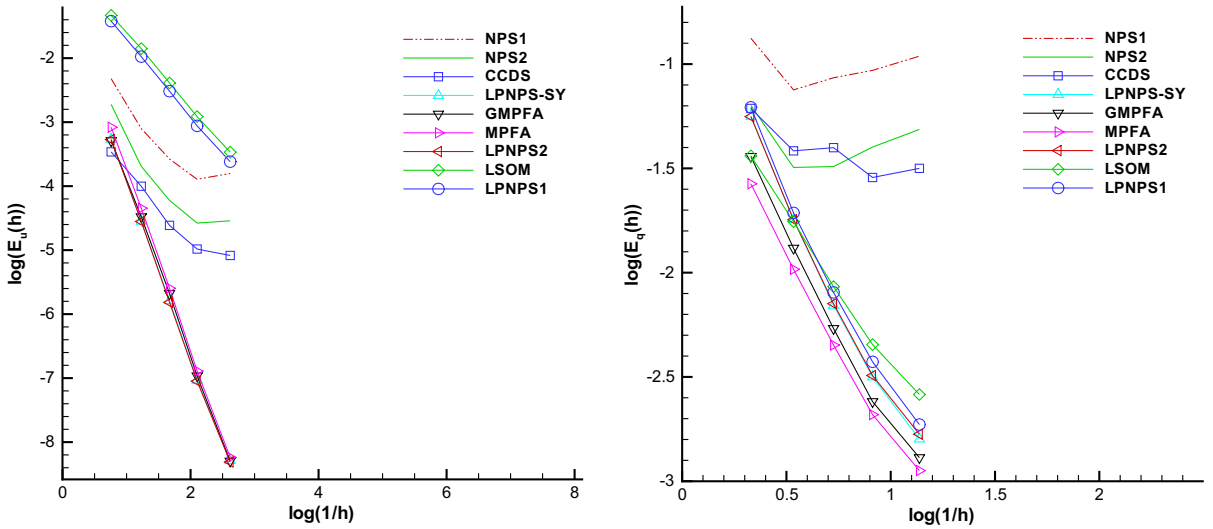


(a) solution (b) edge normal flux

Fig. 11. Convergence behavior on the uniform trapezoidal mesh ($k = 10^{-3}$).

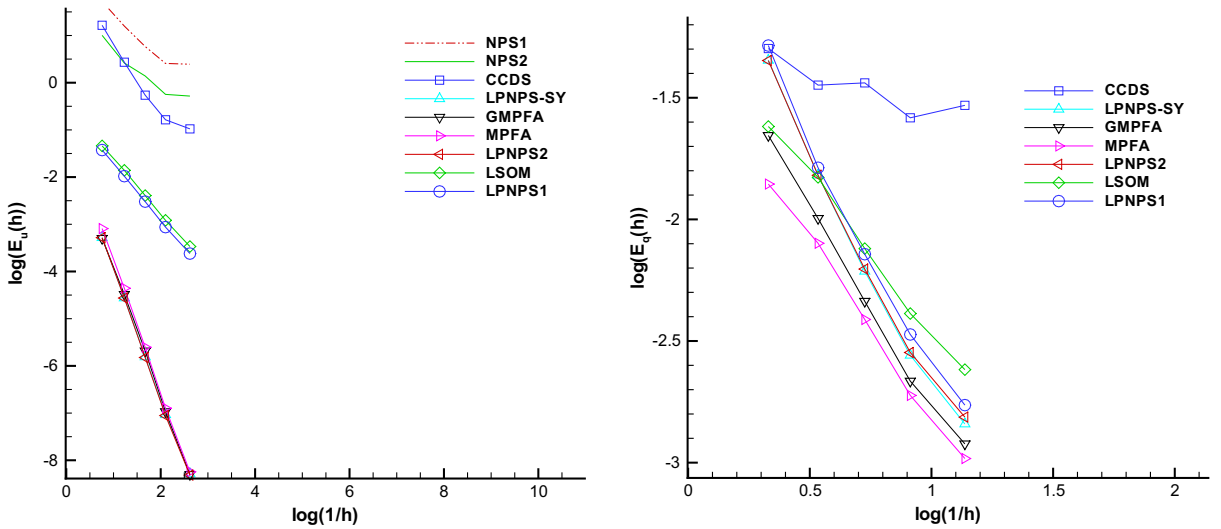
$$u(x,y) = \begin{cases} 1 + x + y + (x-0.5)^2 e^{x+y}, & 0 < x \leqslant 0.5, \\ \frac{3k-1}{2k} + \frac{x}{k} + y + (x-0.5)^2 e^{x+y}, & 0.5 < x < 1. \end{cases} \tag{6.2}$$

Obviously, this solution and its relevant normal flux at the interface are continuous. We first employ the uniform trapezoidal mesh and examine different values of k. the convergence rates are graphically depicted in Figs. 10 and 11 as log-log plots of the solution errors and flux errors versus the characteristic mesh size h. The actual convergence order is reflected by the slopes of the experimental error curves. The second mesh we employ is the modified Shestakov mesh shown in Fig. 9, where the material interface coincides with the mesh line. The convergence results are given in Figs. 12 and 13. We can see that the linearity preserving schemes performs very well on the discontinuity cases and have a comparable convergence rates with the famous MPFA and LSOM schemes. Besides, we can see from the Figs. 10–13 that the accuracy of LPNPS2 is almost the same as that of LPNPS-SY. Here, the results for NPS1 and NPS2 are not drawn in Fig. 11(b) and Fig. 13(b) due to large errors.



(a) solution (b) edge normal flux

Fig. 12. Convergence behavior on the modified Shestakov mesh ($k = 10^{-1}$).



(a) solution (b) edge normal flux

Fig. 13. Convergence behavior on the modified Shestakov mesh ($k = 10^{-3}$).

Contrast to the traditional MPFA schemes, the generalized MPFA scheme employs a dynamic continuity point with $\tau \in (0,1)$. We observe that convergence rates have been tested numerically in [2,13] for $\tau = 0.5$ and in [4] for a comparison of $\tau = 0.5$ with $\tau = 0.75$. The authors of [25] present the convergence results for the MPFA type scheme with $\tau = 0.5, 0.75, 0.8565, 0.95$. However, the above investigations are confined to certain fixed continuity point with $\tau \in [0.5, 1)$ due to the restriction of interaction region or dual cell. In the following two examples, we will study the numerical behavior of the generalized MPFA scheme with $\tau \in (0,1)$.

Example 6.3. Solve the boundary value problem (6.1) with Neumann boundary condition on the top boundary and Dirichlet boundary condition on the rest part of the boundary.

We investigate the accuracy for the 64×64 Kershaw mesh, random mesh, trapezoidal mesh and Shestakov mesh, respectively. Figs. 14 shows the solution error for different locations of the continuity point T_k, where the points between $\tau \in (0, 0.15)$ are not drawn due to large errors. Fig. 15 describes the flux errors. We can see that the minimum discrete L_2 error is reached around $\tau = 0.5$.

Example 6.4. We solve the same problem in Example 6.2. The top boundary is acted as a Neumann boundary and the Dirichlet boundary condition is imposed on the rest part of the boundary. Here we set $k = 10^{-3}$. This time three types of meshes,

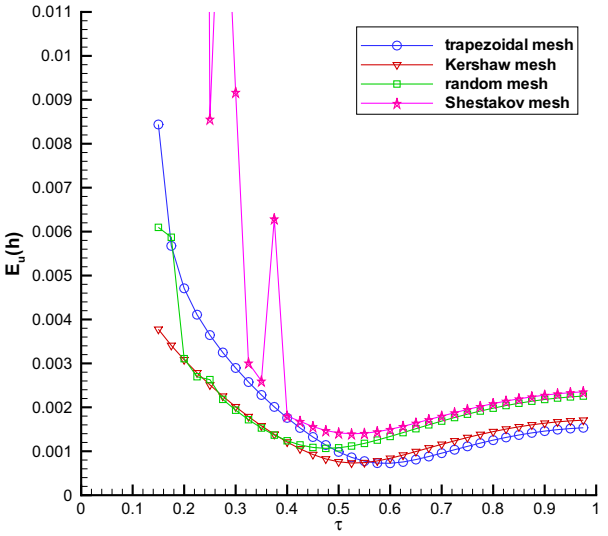


Fig. 14. L_2 error for the solution.

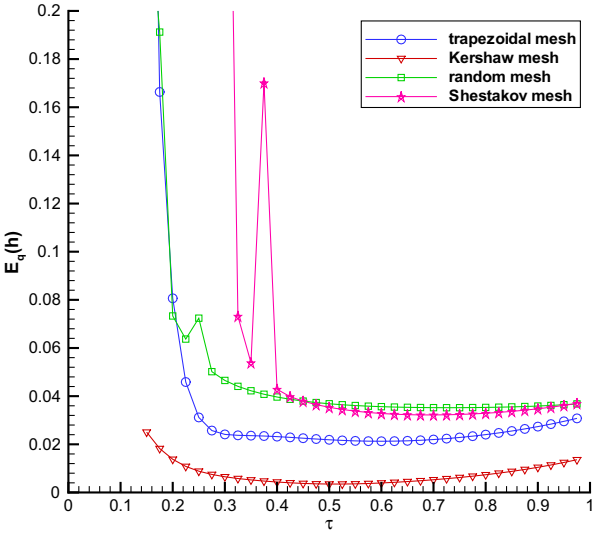


Fig. 15. L_2 error for the flux.

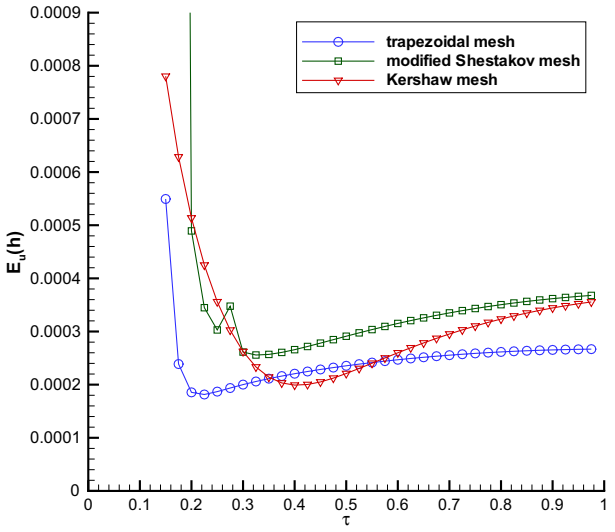


Fig. 16. L_2 error for the solution.

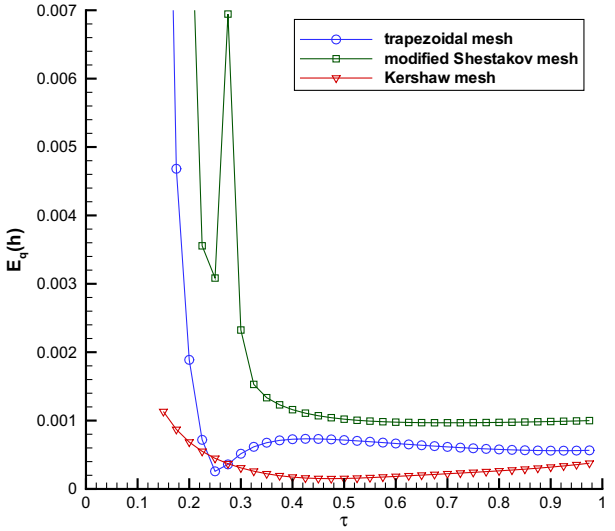


Fig. 17. L_2 error for the flux.

i.e., the 64×64 trapezoidal mesh, Kershaw mesh and the modified Shestakov mesh, are involved. The L_2 errors for the solution and the flux are shown in Figs. 16 and 17, respectively. In this special case, desirable results can be expected around $\tau = 0.4$.

7. Conclusions

We have discussed the problem of applying the linearity preserving method to improve the accuracy of a cell-centered nine-point scheme. Both the derivation of the nine-point scheme and the treatments of vertex unknowns are subjected to the so-called linearity preserving criterion. By investigating the relations between this scheme and some existing schemes, a generalized MPFA scheme, which is also a nine-point linearity preserving scheme, is suggested. Numerical results show that our linearity preserving nine-point schemes have almost second order accuracy on most highly distorted quadrilateral meshes.

In deriving the linearity preserving schemes, we employ an MPFA-type technique to introduce a continuity point on the cell edge. Contrast to the traditional MPFA method, the continuity point here is a dynamic one on the whole cell edge with a parameter τ which may have different values at different interior mesh vertices and can even be different for the four edges sharing the same mesh vertex. The introduction of the dynamic continuity point enables us to analyze the solvability of the

local linear system and to locate some of the singular points, which is not a trivial problem since the linear system depends not only on the local geometry of the mesh but also the diffusion coefficient. The discussion for the solvability of the local linear system contributes not only to our algorithms but also to some physical space derived MPFA algorithms. The problems of locating the rest three possible singular points and exploiting other advantages of the choice of dynamic continuity point constitute the topics of some future works.

Acknowledgment

This work was partially supported by the National Basic Research Program (2005CB321703), the National Natural Science Foundation of China (Nos. 10871030, 90718029) and a grant from the Laboratory of Computational Physics. The authors would like to thank the reviewers and editor for their valuable suggestions and comments.

References

[1] I. Aavatsmark, T. Barkve, Ø. Bøe, T. Mannseth, Discretization on non-orthogonal, quadrilateral grids for inhomogeneous, anisotropic media, J. Comput. Phys. 127 (1996) 2–14.
[2] I. Aavatsmark, G.T. Eigestad, Numerical convergence of the MPFA O-method and U-method for general quadrilateral grids, Int. J. Numer. Methods Fluids 51 (2006) 939–961.
[3] I. Aavatsmark, G.T. Eigestad, R.A. Klausen, M.F. Wheeler, I. Yotov, Convergence of a symmetric MPFA method on quadrilateral grids, Comput. Geosci. 11 (2007) 333–345.
[4] I. Aavatsmark, G.T. Eigestad, B.T. Mallison, J.M. Nordbotten, A compact multipoint flux approximation method with improved robustness, Numer. Methods Part. Diff. Eq. 24 (2008) 1329–1360.
[5] M.M. Basko, J. Maruhn, An. Tauschwitz, An efficient cell-centered diffusion scheme for quadrilateral grids, J. Comput. Phys. 228 (2009) 2175–2193.
[6] J. Breil, P.-H. Maire, A cell-centered diffusion scheme on two-dimensional unstructured meshes, J. Comput. Phys. 224 (2007) 785–823.
[7] L.N. Chang, G.W. Yuan, Cell-centered finite volume methods with flexible stencils for diffusion equations on general nonconforming meshes, Comput. Methods Appl. Mech. Eng. 198 (2009) 1638–1646.
[8] J. Cheng, C.-W. Shu, A third order conservative Lagrangian type scheme on curvilinear meshes for the compressible Euler equations, Commun. Comput. Phys. 4 (2008) 1008–1024.
[9] Q.-Y. Chen, J. Wan, Y. Yang, R.T. Mifflin, Enriched multi-point flux approximation for general grids, J. Comput. Phys. 227 (2008) 1701–1721.
[10] Y.B. Chen, S. Jiang, An optimization-based rezoning for ALE methods, Commun. Comput. Phys. 4 (2008) 1216–1244.
[11] L. Ding, J.M. Wu, et al, Simulation of Z-pinch implosion using MARED code, High Power Laser Particle Beans 20 (2008) 212–218.
[12] M.G. Edwards, C.F. Rogers, Finite volume discretization with imposed flux continuity for the general tensor pressure equation, Comput. Geosci. 2 (1998) 259–290.
[13] G.T. Eigestad, R.A. Klausen, On the convergence of the multi-point flux approximation O-method: numerical experiments for discontinuous permeability, Numer. Methods Part. Diff. Eq. 21 (2005) 1079–1098.
[14] S.W. Fu, H.Q. Fu, L.J. Shen, A nine-point difference scheme for the two-dimensional equations of heat conduction with three temperatures, J. Numer. Methods Comput. Appl. 20 (1999) 237–240;
S.W. Fu, H.Q. Fu, L.J. Shen, Translation in Chinese, J. Numer. Math. Appl. 22 (2000) 45–48.
[15] C.H. Han, T. Kang, J.M. Wu, A linearity preserving difference scheme for the numerical solution of diffusion equation on distorted meshes, J. Commun. Univ. China Sci. Technol. 15 (2008) 40–44.
[16] W.Z. Huang, A.M. Kappen, A study of cell-center finite volume methods for diffusion equations, Mathematics Research Report 98-10-01, Department of Mathematics, University of Kansas, Laurence KS, 1998.
[17] D. Kershaw, Differencing of the diffusion equation in Lagrangian hydrodynamic codes, J. Comput. Phys. 39 (1981) 375–395.
[18] R.A. Klausen, A.F. Stephansen, Mimetic MPFA, in: Proceedings of the 11th European Conference on the Mathematics of Oil Recovery, Bergen, Norway, A12, EAGE, 2008.
[19] R.A. Klausen, R. Winther, Robust convergence of multi point flux approximation on rough grids, Numer. Math. 104 (2006) 317–337.
[20] D.Y. Li, H.S. Shui, M. Tang, J. Tang, On the finite difference scheme of two-dimensional parabolic equation in a non-rectangular mesh, J. Numer. Methods Comput. Appl. 1 (1980) 217–224.
[21] K. Lipnikov, M. Shashkov, I. Yotov, Local flux mimetic finite difference methods, Numer. Math. 112 (2009) 115–152.
[22] J.M. Morel, J.E. Dendy Jr., M.L. Hall, S.W. White, A cell-centered Lagrangian-mesh diffusion differencing scheme, J. Comput. Phys. 103 (1992) 286–299.
[23] J.E. Morel, R.M. Roberts, M.J. Shashkov, A local support-operators diffusion discretization scheme for quadrilateral $r-z$ meshes, J. Comput. Phys. 144 (1998) 17–51.
[24] J.M. Nordbotten, I. Aavatsmark, G.T. Eigestad, Monotonicity of control volume methods, Numer. Math. 106 (2007) 255–288.
[25] M. Pal, M.G. Edwards, A.R. Lamb, Convergence study of a family of flux-continuous finite-volume schemes for the general tensor pressure equation, Int. J. Numer. Methods Fluids 51 (2006) 1177–1203.
[26] W.B. Pei, The construction of simulation algorithms for laser fusion, Commun. Comput. Phys. 2 (2007) 255–270.
[27] M. Shashkov, S. Steinberg, Solving diffusion equations with rough coefficients in rough grids, J. Comput. Phys. 129 (1996) 383–405.
[28] Z.Q. Sheng, G.W. Yuan, A nine point scheme for the approximation of diffusion operators on distorted quadrilateral meshes, SIAM J. Sci. Comput. 30 (2008) 1341–1361.
[29] M.F. Wheeler, I. Yotov, A multipoint flux mixed finite element method, SIAM J. Numer. Anal. 44 (2006) 2082–2106.
[30] J.M. Wu, S.W. Fu, L.J. Shen, A difference scheme with high resolution for the numerical solution of a nonlinear diffusion equation, J. Numer. Methods Comput. Appl. 24 (2003) 116–128;
J.M. Wu, S.W. Fu, L.J. Shen, Translation in Chinese, J. Numer. Math. Appl. 25 (2003) 79–94.
[31] J.M. Wu, Linearly exact method and the difference scheme for diffusion equation on quadrilateral meshes, Annual Report, Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Beijing, China, 2005, pp. 156–169.
[32] G.W. Yuan, X.D. Hang, Tangential flux on grid edge in nine-point schemes for diffusion equations, Chinese J. Comput. Phys. 25 (2008) 7–14.
[33] G.W. Yuan, Z.Q. Sheng, Calculating the vertex unknowns of nine point scheme on quadrilateral meshes for diffusion equation, Sci. China, Ser. A: Math. 51 (2008) 1–13.
[34] G.W. Yuan, Z.Q. Sheng, Monotone finite volume schemes for diffusion equations on polygonal meshes, J. Comput. Phys. 227 (2008) 6288–6312.