Commun. Comput. Phys.
doi: 10.4208/cicp.120914.311214a

Vol. **18**, No. 2, pp. 297-320
August 2015

A Constrained Finite Element Method Based on Domain Decomposition Satisfying the Discrete Maximum Principle for Diffusion Problems

Xingding Chen[1,*] and Guangwei Yuan[2]

[1] *Department of Mathematics, School of Science, Beijing Technology and Business University, Beijing 100048, P.R. China.*
[2] *LCP, Institute of Applied Physics and Computational Mathematics, P.O. Box 8009, Beijing 100088, P.R. China.*

Received 12 September 2014; Accepted (in revised version) 31 December 2014

Abstract. In this paper, we are concerned with the constrained finite element method based on domain decomposition satisfying the discrete maximum principle for diffusion problems with discontinuous coefficients on distorted meshes. The basic idea of domain decomposition methods is used to deal with the discontinuous coefficients. To get the information on the interface, we generalize the traditional Neumann-Neumann method to the discontinuous diffusion tensors case. Then, the constrained finite element method is used in each subdomain. Comparing with the method of using the constrained finite element method on the global domain, the numerical experiments show that not only the convergence order is improved, but also the nonlinear iteration time is reduced remarkably in our method.

AMS subject classifications: 65N12, 65N30

Key words: Discrete maximum principle, discontinuous coefficients, domain decomposition, constrained finite element method.

1 Introduction

There are many applications of diffusion equations, such as geology components, image dispose and so on. In some multi-material problems, the diffusion tensors are discontinuous, and the computational meshes are usually distorted. So, in practical computations, we have to solve the diffusion equations with discontinuous coefficients on distorted meshes. A good numerical scheme should be not only stable and convergent,

*Corresponding author. *Email addresses:* chenxd@lsec.cc.ac.cn (X. D. Chen), ygw8009@sina.com (G. W. Yuan)

 ©2015 Global-Science Press

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

but also possesses the mathematical property of the physical system, such as positivity-preserving or satisfying the discrete maximum principle (DMP). The maximum principle means, if the source term is non-positive, then the solution attains its maximum on the boundary. A numerical scheme that does not generate spurious extrema in the interior of the domain is said to satisfy the discrete maximum principle, which makes it possible to prove the uniform convergence of an approximate solution to the exact one. In the sufficient conditions of the DMP, M-matrix plays an important role [2, 4]. If the inverse of the coefficient matrix is non-negative, then the solution of the algebraic system is also non-negative for any non-negative right-hand side. A numerical scheme enjoys this property is called positivity-preserving. We should point out that, the DMP criterion is more stringent than the positivity-preserving since it requires that the row sums of the coefficient matrix be zero for all interior nodes.

As we know, it is very difficult to construct a numerical scheme satisfying the discrete maximum principle. Based on the finite volume discretization, the continuous edge flux scheme [10], which needs high regularity, can solve the diffusion problems with smooth coefficients effectively, but fails for the discontinuous coefficients. The multi-point flux approximation (MPFA) methods [11–13] and the mimetic finite difference (MFD) methods [14, 15] are monotone for the shape-regular meshes, but when the diffusion tensors are anisotropic or the meshes are distorted, these methods do not satisfy the positivity-preserving property. The diffusion scheme, which is addressed in [24], is positivity-preserving on various distorted meshes for both smooth and non-smooth solutions. But, from the structure of the algebraic system, we can see that this scheme does not satisfy the discrete maximum principle.

For the finite element methods, the sufficient conditions of the DMP are imposed by severe restrictions on the choice of basis functions and on the geometric properties of the mesh. For example, when the triangulation is acute or non-obtuse, the piecewise-linear finite element solutions of the Poisson equation satisfying the DMP [3]. When the quadrilaterals are non-narrow, the bilinear finite element solutions satisfy the DMP. However, these geometric restrictions fail in the case of high-order finite elements, singularly perturbed convection-diffusion equations and anisotropic diffusion problems. A nonlinear Galerkin finite element method [1] is proposed for isotropic Laplace equation on distorted meshes. Based on repair techniques and constrained optimizations, the methods addressed in [5–7] enforce the linear finite element solution and the mixed element solution satisfying the DMP. Two approaches are addressed in [5]. The first approach is based on the repair technique, which is a posteriori correction of the discrete solution. The second approach is based on the constrained optimization, in which the linear constraints are introduced. In these two approaches, only the global discrete energy of the solution is preserved. We should point out that when the total available energy is less than the total needed one, this repair technique will result in a dead circle and has to terminate, and the solution of the constrained optimization is very expensive as the number of unknowns is increased. In [6], two non-negative mixed finite element formulations for tensorial diffusion equations based on constrained optimization techniques are proposed. The first for-

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

mulation is based on the Raviart-Thomas spaces, and the other is based on the variational multiscale formulation. The main idea of [6] is to augment a constraint on nodal values to be non-negative to the discrete variational formulation. From the KKT optimal conditions, one can see these optimization-based formulations only have global mass balance property. Besides, when the three-node triangular meshes are used to solve the diffusion problems with smooth coefficients, the additional CPU time to obtain non-negative solution increases dramatically as mesh refinements. The authors in [7] developed a finite element formulation for anisotropic diffusion equations with decay on general computational grids for lower-order finite elements. This formulation only preserves the global mass balance property, and the CPU time increases dramatically when the meshes are refined. Recently, a constrained finite element method is described in [8] to enforce the solution satisfying the DMP. This method preserves the local mass balance property, and the convergence orders are about the same as the standard finite element method when the coefficients are smooth. But, when this method is used for the problem with discontinuous coefficients on distorted triangular mesh, not only the convergence order loses, but also the nonlinear iteration converges very slowly.

In this paper, we will combine the domain decomposition method with the constraint finite element method [8] to make the finite element solutions satisfy the DMP on distorted meshes. We hope this new method will not degrade the convergence order for the diffusion problems with discontinuous coefficients, and the total CPU time of the nonlinear iteration in our method is less than that of the original constrained finite element method. It is well-known that, when the diffusion coefficients are discontinuous, the gradient recovery is always inaccurate along the discontinuous lines. In order to improve the accuracy, we firstly decompose the computational domain into several subdomains along the discontinuous lines to guarantee that the diffusion coefficients are smooth in each subdomain. Then, a modified Neumann-Neumann method is used to get the information on the interfaces. Finally, we implement the constrained finite element method [8] in each subdomain to get the solution, which satisfies the discrete maximum principle.

The outline of this paper is as follows. In Section 2, we give the model problem and the continuous maximum principle, which is emulated at the discrete level. Based on the domain decomposition, we generalize the traditional Neumann-Neumann method to the discontinuous diffusion tensors case in Section 3, and give the theoretical analysis in Section 4. The constrained finite element method [8], which is used in each subdomain, is simply described in Section 5. Numerical experiments are performed in Section 6 which followed by a summary and conclusions in Section 7.

2 The model problem and the maximum principle

Consider the stationary diffusion problem

$$\begin{cases} -\nabla\cdot(D\nabla\bar{u}) = q, & \text{in } \Omega, \\ \bar{u} = g, & \text{on } \partial\Omega, \end{cases} \tag{2.1}$$

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

where Ω is an open bounded polygonal set of $\mathbb{R}^2$ with Lipschitz continuous boundary $\partial\Omega$, D is a piecewise constant diffusion tensor (possibly anisotropic), and q is a source function. The maximum and minimum principle can be formulated as follows.

Theorem 2.1. *The solution $\bar{u}\in C^2(\Omega)\cap C(\bar{\Omega})$ of problem* (2.1) *attains its maxima (minima) on the boundary $\partial\Omega$ if q is non-positive (non-negative) in Ω*

$$q\leq 0 \implies \max_{\mathbf{x}\in\bar{\Omega}}\bar{u}(\mathbf{x})=\max_{\mathbf{x}\in\partial\Omega}g(\mathbf{x}),\ \ \forall\mathbf{x}\in\Omega, \tag{2.2}$$

$$q\geq 0 \implies \min_{\mathbf{x}\in\bar{\Omega}}\bar{u}(\mathbf{x})=\min_{\mathbf{x}\in\partial\Omega}g(\mathbf{x}),\ \ \forall\mathbf{x}\in\Omega. \tag{2.3}$$

There are two useful corollaries of Theorem 2.1.

Corollary 2.1. *If $q=0$, then the solution $\bar{u}\in C^2(\Omega)\cap C(\bar{\Omega})$ of problem* (2.1) *is bounded by*

$$\min_{\mathbf{x}\in\partial\Omega}g(\mathbf{x})\leq\bar{u}(\mathbf{x})\leq\max_{\mathbf{x}\in\partial\Omega}g(\mathbf{x}),\ \ \forall\mathbf{x}\in\Omega. \tag{2.4}$$

Corollary 2.2. *The solution $\bar{u}\in C^2(\Omega)\cap C(\bar{\Omega})$ of problem* (2.1) *preserves the sign of the boundary data $g\in C(\partial\Omega)$ if q and g are non-positive (non-negative) in Ω*

$$q\leq 0,\ g\leq 0 \implies \bar{u}\leq 0 \quad in\ \Omega, \tag{2.5}$$

$$q\geq 0,\ g\geq 0 \implies \bar{u}\geq 0 \quad in\ \Omega. \tag{2.6}$$

This criterion is commonly referred to as non-negativity or positivity preservation. It ensures that positive boundary data and sources cannot create negative concentrations or temperatures.

3 Domain decomposition and the Neumann-Neumann method

For simplicity, we consider the two subdomains problems, which can be generalized to multi-subdomain problems without essential difficulty. Let the domain Ω be decomposed into two non-overlapping subdomains Ω_1 and Ω_2 along the discontinuous line Γ, which means

$$\Omega_1\cap\Omega_2=\emptyset,\quad \bar{\Omega}_1\cup\bar{\Omega}_2=\bar{\Omega},\quad \Gamma=\partial\Omega_1\cap\partial\Omega_2.$$

As usual, we assume the restriction D_k of D is smooth in each subdomain Ω_k $(k=1,2)$. In order to get the solution on the interface Γ, we will generalize the traditional Neumann-Neumann method to the discontinuous diffusion tensors case.

The traditional Neumann-Neumann method, which is constructed for Poisson equation with smooth coefficients, can be described as follows: we start from an initial guess u_Γ^0 on the interface Γ. A step of the Neumann-Neumann algorithm consists in first solving Dirichlet problems on each subdomain Ω_k with data u_Γ^0 on Γ, and then solve a problem on each subdomain with Neumann data on Γ, which is chosen as the difference of the

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

normal derivatives of the solutions of the two Dirichlet problems. The values on Γ of the solutions of these Neumann problems are then employed to correct the initial u_Γ^0 and find the new data u_Γ^1, repeat the iteration until converge. The details of the traditional Neumann-Neumann method can be found in chapter 1 of [18].

Now, we generalize the traditional Neumann-Neumann algorithm to the diffusion equation (2.1) with discontinuous coefficients.

Algorithm 3.1: Modified algorithm

Given u_Γ^0 as an initial guess, ε is the iteration terminate tolerance, $\mathbf{n_k}(k=1,2)$ is the unit outward normal of Ω_k on the interface Γ, $\mathbf{n_1}=-\mathbf{n_2}$. When u_Γ^n is gotten, compute u_Γ^{n+1} by the following way:

Step 1. Find $u_k^{n+\frac{1}{2}}\in H^1(\Omega_k)$, such that

$$\begin{cases} -\nabla\cdot(D_k\nabla u_k^{n+\frac{1}{2}})=q, & \text{in } \Omega_k,\\ u_k^{n+\frac{1}{2}}=g, & \text{on } \partial\Omega_k\backslash\Gamma \quad (k=1,2),\\ u_k^{n+\frac{1}{2}}=u_\Gamma^n, & \text{on } \Gamma. \end{cases}$$

Step 2. Find $\phi_k^{n+1}\in H^1(\Omega_k)$, such that

$$\begin{cases} -\nabla\cdot(D_k\nabla\phi_k^{n+1})=0, & \text{in } \Omega_k,\\ \phi_k^{n+1}=0, & \text{on } \partial\Omega_k\backslash\Gamma \quad (k=1,2),\\ D_k\nabla\phi_k^{n+1}\cdot\mathbf{n_k}=D_1\nabla u_1^{n+\frac{1}{2}}\cdot\mathbf{n_1}+D_2\nabla u_2^{n+\frac{1}{2}}\cdot\mathbf{n_2}, & \text{on } \Gamma. \end{cases}$$

Step 3. Choose a suitable parameter θ, compute

$$u_\Gamma^{n+1}=u_\Gamma^n-\theta(\phi_1^{n+1}+\phi_2^{n+1}).$$

Step 4. When $\|u_\Gamma^{n+1}-u_\Gamma^n\|<\varepsilon$, the iteration terminates; if not, go to step 1.

In Section 4, we will give the convergence analysis of Algorithm 3.1.

4 The theoretical analysis

In this section, we present the theoretical analysis of Algorithm 3.1. Firstly, the important *Steklov-Poincaré* operator on the interface Γ is introduced. Then, we prove Algorithm 3.1 is equivalent to a preconditioned Richardson iteration. Finally, some criterions to choose the parameter θ are given.

4.1 The Steklov-Poincaré operator

We assume $u\in H^1(\Omega)$ is the weak solution of the model problem (2.1), set

$$\mathcal{L}u=-\nabla\cdot(D\nabla u), \quad u_k=u|_{\Omega_k}\ (k=1,2).$$

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

then u_k satisfies the following system

$$\begin{cases} \mathcal{L}u_k = q, & \text{in } \Omega_k, \\ u_k = g, & \text{on } \partial\Omega_k \cap \partial\Omega, \\ u_1 = u_2, & \text{on } \Gamma, \\ D_1 \nabla u_1 \cdot \mathbf{n}_1 = D_2 \nabla u_2 \cdot \mathbf{n}_1, & \text{on } \Gamma. \end{cases} \tag{4.1}$$

The third and forth equations of the system (4.1) are called the interface conditions, which guarantee the continuity of the solution and the continuity of the normal derivatives across the interface Γ.

Set $\lambda = u|_\Gamma$, then u_k $(k=1,2)$ satisfies the following problem in each subdomain Ω_k, respectively.

$$\begin{cases} \mathcal{L}u_k = q, & \text{in } \Omega_k, \\ u_k = g, & \text{on } \partial\Omega_k \cap \partial\Omega, \\ u_k = \lambda, & \text{on } \Gamma. \end{cases} \tag{4.2}$$

The difficulty to solve the problem (4.2) is that the value of λ is unknown. In the following, we use the interface condition

$$D_1 \nabla u_1(\lambda) \cdot \mathbf{n}_1 = D_2 \nabla u_2(\lambda) \cdot \mathbf{n}_1 \tag{4.3}$$

to derive the interface equation of λ.

According to the principle of superposition, the solution of (4.2) can be written as

$$u_k = R_k \lambda + T_k f,$$

where $T_k f$ satisfies

$$\begin{cases} \mathcal{L}T_k f = q, & \text{in } \Omega_k, \\ T_k f = 0, & \text{on } \Gamma, \\ T_k f = g, & \text{on } \partial\Omega_k \backslash \Gamma, \end{cases} \tag{4.4}$$

and the harmonic extension $R_k\lambda$ satisfies

$$\begin{cases} \mathcal{L}R_k \lambda = 0, & \text{in } \Omega_k, \\ R_k \lambda = \lambda, & \text{on } \Gamma, \\ R_k \lambda = 0, & \text{on } \partial\Omega_k \backslash \Gamma. \end{cases} \tag{4.5}$$

Combining (4.4), (4.5) with (4.3), we get

$$(D_1 \nabla R_1 - D_2 \nabla R_2)\lambda \cdot \mathbf{n}_1 = (D_2 \nabla T_2 - D_1 \nabla T_1) f \cdot \mathbf{n}_1, \quad \text{on } \Gamma. \tag{4.6}$$

Set $\mathcal{X} = (D_2 \nabla T_2 - D_1 \nabla T_1) f \cdot \mathbf{n}_1$, which is independent of λ. Then, the system (4.6) can be written as

$$\mathcal{S}\lambda = \mathcal{X}, \tag{4.7}$$

where

$$\mathcal{S} = \mathcal{S}_1 + \mathcal{S}_2, \quad \mathcal{S}_k = D_k \nabla(R_k) \cdot \mathbf{n}_k, \quad k = 1,2.$$

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

The so-defined operator $\mathcal{S}$ is called the *Steklov-Poincaré* operator. When the diffusion tensor D_k is symmetric positive definite, it is easy to prove the operator $\mathcal{S}_k : H^{\frac{1}{2}}(\Gamma) \to H^{-\frac{1}{2}}(\Gamma)(k=1,2)$ is also symmetric positive definite.

Remark 4.1. Generally speaking, it is very difficult to compute the inverse of $\mathcal{S}$. But, it is easy to get the inverse of $\mathcal{S}_k$. For example, find $\lambda \in H^{\frac{1}{2}}(\Gamma)$, such that

$$\mathcal{S}_k\lambda = f, \quad f \in H^{-\frac{1}{2}}(\Gamma).$$

To get the solution λ, we can solve the Neumann boundary value problem in Ω_k: find $u_k \in V_k$, such that

$$a_k(u_k,v) = \int_\Gamma f v ds + \int_{\partial\Omega\setminus\Gamma} g v ds, \ \ \forall v \in V_k.$$

where $a_k(\cdot,\cdot)$ is the bilinear form

$$a_k(u_k,v) = \int_{\Omega_k} D_k \nabla u_k \cdot \nabla v dx$$

and V_k is the finite element space on Ω_k. Then, set $\lambda = u_k|_\Gamma$.

4.2 The preconditioned Richardson iteration

Consider the interface equation

$$\mathcal{S}\lambda = \mathcal{X}. \tag{4.8}$$

As we mentioned above, it is easy to compute $\mathcal{S}_k^{-1}$ $(k=1,2)$, which can be derived by solving a Neumann boundary value problem in Ω_k. So, we can use $\mathcal{S}_k$ as a preconditioner to construct a preconditioned iteration for (4.8).

Given λ^0 as an initial guess, the preconditioned Richardson iteration can be written as

$$\lambda^{n+1} - \lambda^n = \theta M^{-1}(\mathcal{X} - \mathcal{S}\lambda^n), \tag{4.9}$$

where M^{-1} is the preconditioner, θ is the relax factor. In the following, we will give the relationship between Algorithm 3.1 and the preconditioned Richardson iteration (4.9).

Theorem 4.1. *The modified Neumann-Neumann Algorithm 3.1 is equivalent to the preconditioned Richardson iteration* (4.9) *where* $M^{-1} = \mathcal{S}_1^{-1} + \mathcal{S}_2^{-1}$.

Proof. Assume $\bar{u}$ is the exact solution of the diffusion problem (2.1), set the errors

$$\mu^n = \bar{u}|_\Gamma - \lambda^n, \quad e_k^n = \bar{u} - u_k^{n+\frac{1}{2}} (k=1,2).$$

Consider the iterative steps in Algorithm 3.1, we have

$$\begin{cases} \mathcal{L}e_k^n = 0, & \text{in } \Omega_k, \\ e_k^n = 0, & \text{on } \partial\Omega_k\setminus\Gamma, \\ e_k^n = \mu^n, & \text{on } \Gamma, \end{cases} \tag{4.10}$$

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

and

$$\begin{cases} \mathcal{L}\phi_k^{n+1}=0, & \text{in } \Omega_k, \\ \phi_k^{n+1}=0, & \text{on } \partial\Omega_k\backslash\Gamma, \\ D_k\nabla\phi_k^{n+1}\cdot\mathbf{n}_k=-D_1\nabla e_1^n\cdot\mathbf{n}_1-D_2\nabla e_2^n\cdot\mathbf{n}_2, & \text{on } \Gamma. \end{cases} \tag{4.11}$$

Also,

$$\mu^{n+1}-\mu^n=\theta(\phi_1^{n+1}+\phi_2^{n+1}).$$

According to the definition of $\mathcal{S}_k$, we get

$$\mathcal{S}_1\phi_1^{n+1}|_\Gamma=-\mathcal{S}\mu^n,\quad \mathcal{S}_2\phi_2^{n+1}|_\Gamma=-\mathcal{S}\mu^n,$$

then

$$\mu^{n+1}-\mu^n=\theta(-\mathcal{S}_1^{-1}\mathcal{S}\mu^n-\mathcal{S}_2^{-1}\mathcal{S}\mu^n),$$

namely

$$\mu^{n+1}-\mu^n=-\theta(\mathcal{S}_1^{-1}+\mathcal{S}_2^{-1})\mathcal{S}\mu^n.$$

Using $\lambda=\bar{u}|_\Gamma$ and $\mathcal{S}\lambda=\mathcal{X}$, we obtain

$$\bar{u}|_\Gamma-\lambda^{n+1}-(\bar{u}|_\Gamma-\lambda^n)=-\theta(\mathcal{S}_1^{-1}+\mathcal{S}_2^{-1})\mathcal{S}(\bar{u}|_\Gamma-\lambda^n),$$

then

$$-\lambda^{n+1}+\lambda^n=-\theta(\mathcal{S}_1^{-1}+\mathcal{S}_2^{-1})(\mathcal{X}-\mathcal{S}\lambda^n).$$

Therefore

$$\lambda^{n+1}-\lambda^n=\theta(\mathcal{S}_1^{-1}+\mathcal{S}_2^{-1})(\mathcal{X}-\mathcal{S}\lambda^n).$$

This completes the proof. □

The following theorems give the convergence results of the preconditioned Richardson iteration (4.9). We can refer to [18] for the proofs.

Theorem 4.2. *The preconditioned iterative process* (4.9) *is convergent under the inner product* $(\mathcal{S}\cdot,\cdot)$*, so long as* $M-\frac{1}{2}\theta\mathcal{S}$ *is positive definite under the inner product* $(\cdot,\cdot)$*.*

Theorem 4.3. *Assume the eigenvalues* λ_i *of* $M^{-1}\mathcal{S}$ *satisfies* $\lambda_i\in[\alpha,\beta]$*,* $\alpha>0$*. If* θ *satisfies*

$$0<\theta<\min_i\frac{2}{\lambda_i}\leq\frac{2}{\beta},$$

then, the iterative process (4.9) converges. Moreover, the optimal choice of the parameter θ *is* $\hat{\theta}=\frac{2}{\alpha+\beta}$*, and the convergence rate is* $R\approx\frac{2}{cond(M^{-1}\mathcal{S})}=\frac{2\alpha}{\beta}$*, where* $cond(M^{-1}\mathcal{S})$ *is the condition number of* $M^{-1}\mathcal{S}$*.*

Remark 4.2. As we mentioned above, if the eigenvalues of the preconditioned iterative operator are derived, we can choose the optimal parameter θ. If the eigenvalues are difficult to get, we can get θ by some other ways, such as the minimum residual method, see chapter 6 in [23].

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

5 The constrained finite element method

After the solution u_Γ on the interface is gotten, we can use the constrained finite element method in each subdomain with a whole Dirichlet boundary condition to get the solution of the model problem (2.1). Following [8], we implement the constrained finite element method in each subdomain Ω_k $(k=1,2)$ as follows.

1. **Renumber the freedoms**

   We discrete the model problem (2.1) by the standard finite element method in each subdomain $\Omega_k(k=1,2)$. Let $X_1^k, X_2^k, \cdots, X_n^k$ denote the interior nodes in Ω_k, and $X_{n+1}^k, \cdots, X_m^k$ denote the nodes on the boundary $\partial\Omega_k$. Combining with the Dirichlet boundary condition, in which the information on the interface Γ is gotten by the modified Neumann-Neumann method, the algebraic system in each subdomain Ω_k can be written as

   $$A_k u_k = b_k,$$

   where A_k is the matrix form of the bilinear form $a_k(\cdot,\cdot)$, and b_k is the right-hand side. Namely

   $$\begin{pmatrix} A_{\Omega_k\Omega_k} & A_{\Omega_k\partial\Omega_k} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} u_{\Omega_k} \\ u_{\partial\Omega_k} \end{pmatrix} = \begin{pmatrix} b_{\Omega_k} \\ g_k \end{pmatrix},$$

   where $u_{\Omega_k}=[u_1^k,\cdots,u_n^k]^T$ is the vector of interior unknowns in Ω_k, $u_{\partial\Omega_k}=[u_{n+1}^k,\cdots,u_m^k]^T$ is the vector of unknowns on the boundary $\partial\Omega_k$. From the above system, we get

   $$\begin{cases} A_{\Omega_k\Omega_k}\, u_{\Omega_k} + A_{\Omega_k\partial\Omega_k}\, u_{\partial\Omega_k} = b_{\Omega_k}, \\ u_{\partial\Omega_k} = g_k, \end{cases}$$

   then

   $$A_{\Omega_k\Omega_k}\, u_{\Omega_k} = b_{\Omega_k} - A_{\Omega_k\partial\Omega_k} g_k. \tag{5.1}$$

2. **Algebraic splitting**

   Decompose the stiffness matrix $A_k=\{a_{ij}^k\}$ into the good part A_k^- and the bad part A_k^+,

   $$A_k = A_k^- + A_k^+,$$

   where the elements $\{a_{ij}^{k,+}\}$ of A_k^+ are defined by

   $$\begin{aligned} a_{ij}^{k,+} &= \max\{0, a_{ij}^k\}, \quad \forall j \neq i, \\ a_{ii}^{k,+} &= -\sum_{j\neq i} a_{ij}^{k,+}, \quad i,j=1,\cdots,n. \end{aligned}$$

   The elements $\{a_{ij}^{k,-}\}$ of the good part A_k^- are obtained by

   $$a_{ij}^{k,-} = a_{ij}^k - a_{ij}^{k,+}.$$

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

Remark 5.1. From (5.1), we can see only the first n rows of matrix A_k, namely, $(A_{\Omega_k\Omega_k}\ A_{\Omega_k\partial\Omega_k})$, need to be implemented in the algebraic splitting.

3. Slope limiting

To weaken the flux of the bad part A_k^+, a correct factor α_{ij}^k is introduced, define

$$\bar{f}_{ij}^{k,+}=\alpha_{ij}^k a_{ij}^{k,+}(u_i^k-u_j^k),\quad i\neq j,\quad i,j=1,\cdots,n.$$

To enforce the solution satisfying the discrete maximum principle, we apply the slope limiting by the following formula

$$\bar{f}_{ij}^{k,+}=a_{ij}^{k,+}\bar{S}_{ij}^k,\quad \bar{S}_{ij}^k=\alpha_{ij}^k(u_i^k-u_j^k),$$

where u_i^k, u_j^k denote the unknowns on the node i and j in Ω_k, respectively. In order to find the right value of $\bar{S}_{ij}^k$, the approximation of solution gradients at nodes i and j are needed. Let $\{\psi_i^k\}$ be a set of piecewise linear basis functions of the finite element method, a continuous approximation to nodal gradients can be obtained by using the lumped-mass L_2-projection

$$\hat{\nabla}u_i^k=\frac{1}{m_i^k}\sum_l c_{il}^k u_l^k, \tag{5.2}$$

where $m_i^k=(1,\psi_i^k)_{\Omega_k}$ is a diagonal entry of the lumped-mass matrix and $c_{il}^k=(\psi_i^k,\nabla\psi_l^k)_{\Omega_k}$.

For the arbitrary node points (i,j), the distance s_{ij}^k between u_i^k and u_j^k is

$$s_{ij}^k=u_i^k-u_j^k=(X_i^k-X_j^k)\cdot\hat{\nabla}u_i^k.$$

Introducing the maximum and minimum nodal values in Ω_k over the stencil of i

$$u_i^{k,\max}=\max\{u_i^k,u_i^k+\max_{j\neq i}(u_j^k-u_i^k)\},$$
$$u_i^{k,\min}=\min\{u_i^k,u_i^k+\min_{j\neq i}(u_j^k-u_i^k)\},$$

the extrapolated slope s_{ij}^k can be estimated in terms of the steepest gradients

$$\mu_{ij}^k(u_i^{k,\min}-u_i^k)\leq s_{ij}^k\leq\mu_{ij}^k(u_i^{k,\max}-u_i^k),$$

where $\mu_{ij}^k=\frac{1}{m_i^k}\sum_{l\neq i}|c_{il}^k\cdot(X_i^k-X_j^k)|$. Then, we can compute $\bar{S}_{ij}^k$ in the following way:

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

- The nodes i,j are both interior nodes

$$\bar{S}_{ij}^k=\begin{cases}\min\{2\mu_{ij}^k(u_i^{k,\max}-u_i^k),u_i^k-u_j^k,2\mu_{ji}^k(u_j^k-u_j^{k,\min})\}, & u_i^k>u_j^k,\\ \max\{2\mu_{ij}^k(u_i^{k,\min}-u_i^k),u_i^k-u_j^k,2\mu_{ji}^k(u_j^k-u_j^{k,\max})\}, & u_i^k<u_j^k.\end{cases}$$

- The node i is a interior node, and the node j is a boundary node

$$\bar{S}_{ij}^k=\begin{cases}\min\{2\mu_{ij}^k(u_i^{k,\max}-u_i^k),u_i^k-u_j^k\}, & u_i^k>u_j^k,\\ \max\{2\mu_{ij}^k(u_i^{k,\min}-u_i^k),u_i^k-u_j^k\}, & u_i^k<u_j^k.\end{cases}$$

Define

$$\bar{f}_i^{k,+}=\sum_{j\neq i}\bar{f}_{ij}^{k,+},\quad i=1,\cdots,n,$$

then, the bad flux $\bar{f}_{\Omega_k}^+$ can be written as

$$\bar{f}_{\Omega_k}^+=(\bar{f}_1^{k,+},\bar{f}_2^{k,+},\cdots,\bar{f}_n^{k,+}).$$

4. **Defect correction**

The last step of the constrained finite element method [8] is to solve the following system in each subdomain Ω_k $(k=1,2)$

$$\begin{pmatrix}u_{\Omega_k}^{(l+1)}\\ u_{\partial\Omega_k}^{(l+1)}\end{pmatrix}=\begin{pmatrix}u_{\Omega_k}^{(l)}\\ u_{\partial\Omega_k}^{(l)}\end{pmatrix}+\begin{pmatrix}\bar{A}_{\Omega_k\Omega_k}^{(l)} & \bar{A}_{\Omega_k\partial\Omega_k}^{(l)}\\ \mathbf{0} & \mathbf{I}\end{pmatrix}^{-1}\begin{pmatrix}\bar{r}_{\Omega_k}^{(l)}\\ \mathbf{0}\end{pmatrix},$$

where

$$\bar{r}_{\Omega_k}^{(l)}=b_{\Omega_k}-A_{\Omega_k\Omega_k}^-u_{\Omega_k}^{(l)}-A_{\Omega_k\partial\Omega_k}^-\tilde{g}+\bar{f}_{\Omega_k}^{+(l)},\quad l=1,\cdots,L.$$

To avoid the computation of the inverse of the matrix, we introduce an auxiliary linear problem

$$\begin{pmatrix}\bar{A}_{\Omega_k\Omega_k}^{(l)} & \bar{A}_{\Omega_k\partial\Omega_k}^{(l)}\\ \mathbf{0} & \mathbf{I}\end{pmatrix}\begin{pmatrix}v_{\Omega_k}^{(l+1)}\\ v_{\partial\Omega_k}^{(l+1)}\end{pmatrix}=\begin{pmatrix}\bar{r}_{\Omega_k}^{(l)}\\ \mathbf{0}\end{pmatrix},$$

using

$$u_{\Omega_k}^{(l+1)}=u_{\Omega_k}^{(l)}+v_{\Omega_k}^{(l+1)},\quad l=0,\cdots,L$$

to update the solution. Moreover, the auxiliary linear system can be written as

$$\bar{A}_{\Omega_k\Omega_k}^{(l)}\,v_{\Omega_k}^{(l+1)}=\bar{r}_{\Omega_k}^{(l)}.$$

The following positivity-preserving scheme is constructed for the above system.

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

- When $l=0$, choose

$$u_{\Omega_k}^{(0)}=\mathbf{0},\quad u_{\partial\Omega_k}^{(0)}=g_k,\quad \bar{A}_k^{(0)}=A_k^-.$$

- When $l=1,\cdots,L$, choose

$$\bar{A}_k^{(l)}=A_k^-+diag\{\sigma_i\},\quad \sigma_i=2\sum_{j\neq i}\mu_{ij}^k a_{ij}^{k,+}.$$

So, we can get the solution in each subdomain.

Remark 5.2. In this section, we only simply describe the constrain finite element method from the algebraic implementation. We can refer to [8] and the references within for the details of this method.

6 Numerical experiments

In this section, we test the ability of the proposed method for problem (2.1) with discontinuous diffusion tensors. We denote the scheme using the constrained finite element method on the global domain in [8] by Method 1, and denote our method by Method 2.

The computational domain $\Omega=[0,1]\times[0,1]$. The distorted triangular meshes are constructed from the uniform triangular meshes with mesh size h by random perturbation of internal nodes (x,y)

$$x:=x+\alpha\eta_x h,\quad y:=y+\alpha\eta_y h,$$

where η_x and η_y are random numbers with values in the range from -0.5 to 0.5. The parameter $\alpha\in(0,1]$ quantifies the degree of distortion. The default value $\alpha=0.4$ is adopted to introduce sufficiently strong grid deformations without tangling, and assume nodes that belong to the discontinuous line are shifted in the y-direction only. It is pertinent to emphasize that the distortion is performed on each refinement level.

In the numerical experiments, we only test the two subdomains problems. But, our method can be extended to multi-subdomain problems without essential difficulty. The difference between the numerical solution u and the exact solution $\bar{u}$ is measured in norms

$$||\bar{u}-u||_2=\sqrt{\sum_{k=1,2}\sum_{i=1}^{n}m_i^k|\bar{u}(X_i^k)-u_i^k|^2},$$
$$||\bar{u}-u||_\infty=\max_{k=1,2}\max_{1\le i\le n}|\bar{u}(X_i^k)-u_i^k|,$$

where X_i^k denotes the mesh node in the domain Ω_k, $m_i^k=(1,\psi_i^k)_{\Omega_k}$ denotes a diagonal coefficients of the lumped matrix or, equivalently, the area of the control volume associated with the mesh node X_i^k. The linear triangular elements are used, and the terminate criterion of the modified Neumann-Neumann method is

$$\|u_\Gamma^{n+1}-u_\Gamma^n\|\le 1.0e-8.$$

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

When the residual $\bar{r}_{\Omega_k} \leq 1.0e-8$, the iteration of the constrained finite element method in each subdomain terminates.

6.1 Test problem 1

The diffusion tensor D is defined as

$$D(x,y)=\begin{cases} D_1, & x<0.5, \\ D_2, & x>0.5, \end{cases} \quad D_1=\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad D_2=\begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}.$$

This homogeneous diffusion tensor has a jump in the value and direction across the line $x=0.5$. Define $g=0$, the analytical solution of the problem (2.1) is given by

$$u(x,y)=\begin{cases} 1-2y^2+4xy+2y+6x, & x\leq 0.5, \\ b_2y^2+c_2xy+d_2x+e_2y+f_2, & x>0.5. \end{cases} \tag{6.1}$$

Substitute (6.1) into (2.1) yields the following values of the involved coefficients

$$b_2=-2, \quad c_2=\frac{4(D_2(1,2)+1)}{D_2(1,1)}, \quad d_2=\frac{6-4D_2(1,2)}{D_2(1,1)},$$
$$e_2=\frac{4D_2(1,1)-2D_2(1,2)-2}{D_2(1,1)}, \quad f_2=\frac{4D_2(1,1)+2D_2(1,2)-3}{D_2(1,1)}.$$

We decompose the domain Ω into two subdomain Ω_1 and Ω_2 along the discontinuous line $x=0.5$, and choose $\theta=0.02$ in the modified Neumann-Neumann method. The convergence results are presented in Table 1 and Table 2, respectively. From the convergence results, we can see both of the two methods enforce the finite element solutions satisfying the discrete maximum principle. The convergence order in Method 1 oscillates obviously, while more than one-order accuracy is achieved in Method 2.

From the tables, we can see both the original constrained finite element method and the new method with multiple subdomains have a convergence order less than two. This is caused by the slope limiting procedure in Section 5, which is used to modify the stiffness matrix to enforce the solution satisfying the discrete maximum principle. This slope-limited Galerkin scheme employs the lumped-mass projection (5.2) to approximate the nodal gradients. In the case of a heterogeneous diffusion tensor, the gradient is discontinuous but the lumped-mass projection still samples data from both sides of an internal interface. Then, the gradient recovery is inaccurate, which will degrade the convergence order of the method. In the case of sufficiently smooth coefficients, the constrained finite element method preserves the accuracy and consistency of the underlying discretization.

For the modified Neumann-Neumann method occupies very little time in the computation, we only test the CPU time of the nonlinear iteration in the constrained finite element method. The CPU time of the nonlinear iteration of the two methods is presented in Table 3 and Fig. 1. Comparing with Method 1, the total nonlinear iteration time of the two subdomains in Method 2 is reduced remarkably. The numerical solution profiles and the colormaps are shown in Fig. 2.

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

Table 1: Test problem 1. The convergence study for Method 1, $u_{\min}=1.0$, $u_{\max}=6.5$.

h	1/16	1/32	1/64	1/128	1/256
$\|\|u-u_h\|\|_2$	6.15E-3	1.80E-3	1.94E-3	6.47E-4	3.58E-4
convergence order	-	1.77	-0.10	1.58	0.85
$\|\|u-u_h\|\|_\infty$	3.19E-2	1.55E-2	1.93E-2	8.05E-3	5.03E-3
convergence order	-	1.04	-0.31	1.26	0.67

Table 2: Test problem 1. The convergence study for Method 2, $u_{\min}=1.0$, $u_{\max}=6.5$.

h	1/16	1/32	1/64	1/128	1/256
$\|\|u-u_h\|\|_2$	6.10E-3	3.67E-3	1.45E-3	4.94E-4	1.63E-4
convergence order	-	0.73	1.33	1.55	1.58
$\|\|u-u_h\|\|_\infty$	5.08E-2	4.05E-2	1.76E-2	7.25E-3	3.02E-3
convergence order	-	0.32	1.20	1.27	1.26

Table 3: Test problem 1. The CPU time of nonlinear iterations for Method 1 (in Ω) and Method 2 (in Ω_1,Ω_2).

h	1/16	1/32	1/64	1/128	1/256
cpu time(s) Ω	1.07	28.23	586.17	6790.34	116997.55
cpu time(s) Ω_1	0.24	5.19	87.74	1348.43	21274.88
cpu time(s) Ω_2	0.32	6.71	113.63	1754.38	27017.45

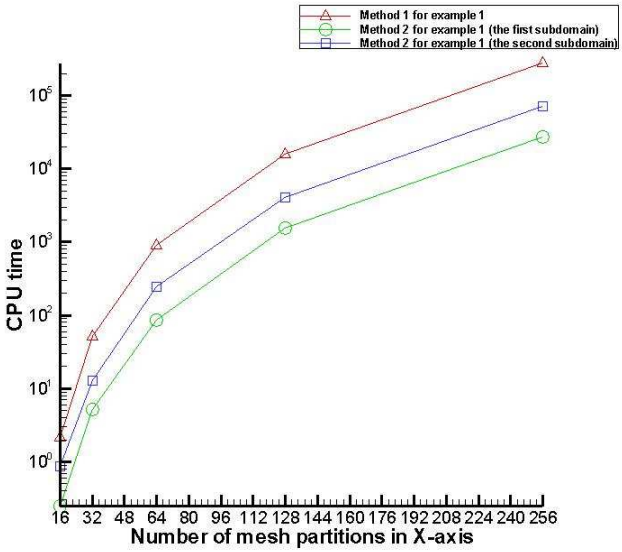


Figure 1: Test problem 1. The CPU time of nonlinear iterations for the two methods.

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

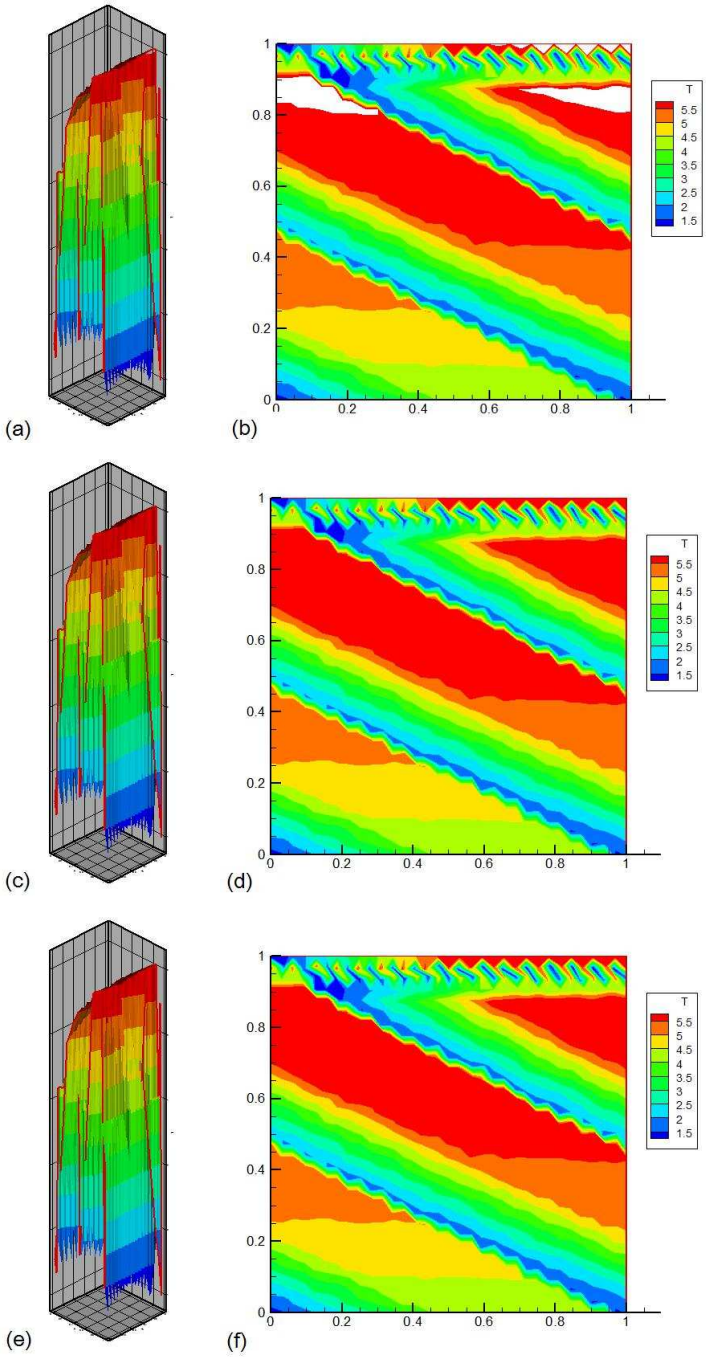


Figure 2: Test problem 1. Solution profiles (left) and solution colormaps (right) of (a),(b) original FEM; (c),(d) Method 1; (e),(f) Method 2. Mesh scale is 64×64. The regions that violate the DMP are indicated in white.

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

6.2 Test problem 2

The diffusion tensor D is defined as follows

$$D(x,y)=\begin{cases} D_1, & x<0.5, \\ D_2, & x>0.5, \end{cases} \quad D_1=\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad D_2=\begin{pmatrix} 10 & 3 \\ 3 & 1 \end{pmatrix}.$$

We use the same exact solution $u(x,y)$ in (6.1), $g=0$. Choose $\theta=0.02$ in the modified Neumann-Neumann method. The convergence results are presented in Table 4 and Table 5, respectively. From the convergence results, we can see both of the two methods enforce the finite element solutions satisfying the discrete maximum principle. The convergence order in Method 1 oscillates obviously, while more than one-order accuracy is achieved in Method 2.

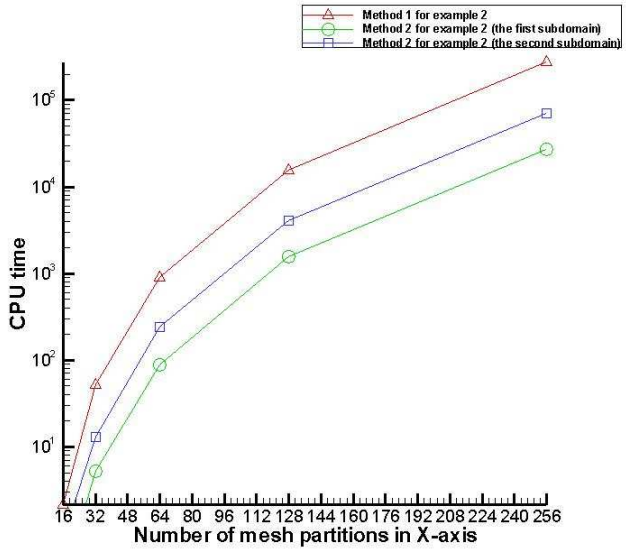


Figure 3: Test problem 2. The CPU time of nonlinear iterations for the two methods.

The CPU time of the nonlinear iteration of the two methods is presented in Table 6 and Fig. 3. Comparing with Method 1, the total nonlinear iteration time in Method 2 is reduced remarkably. The numerical solution profiles and the colormaps are shown in Fig. 4.

Table 4: Test problem 2. The convergence study for Method 1, $u_{\min}=1.0$, $u_{\max}=6.5$.

h	1/16	1/32	1/64	1/128	1/256
$\|\|u-u_h\|\|_2$	2.30E-3	2.84E-3	7.95E-4	5.09E-4	2.69E-4
convergence order	-	-0.30	1.84	0.66	0.89
$\|\|u-u_h\|\|_\infty$	2.01E-2	2.52E-2	1.11E-2	6.03E-3	2.93E-3
convergence order	-	-0.33	1.17	0.88	1.03

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

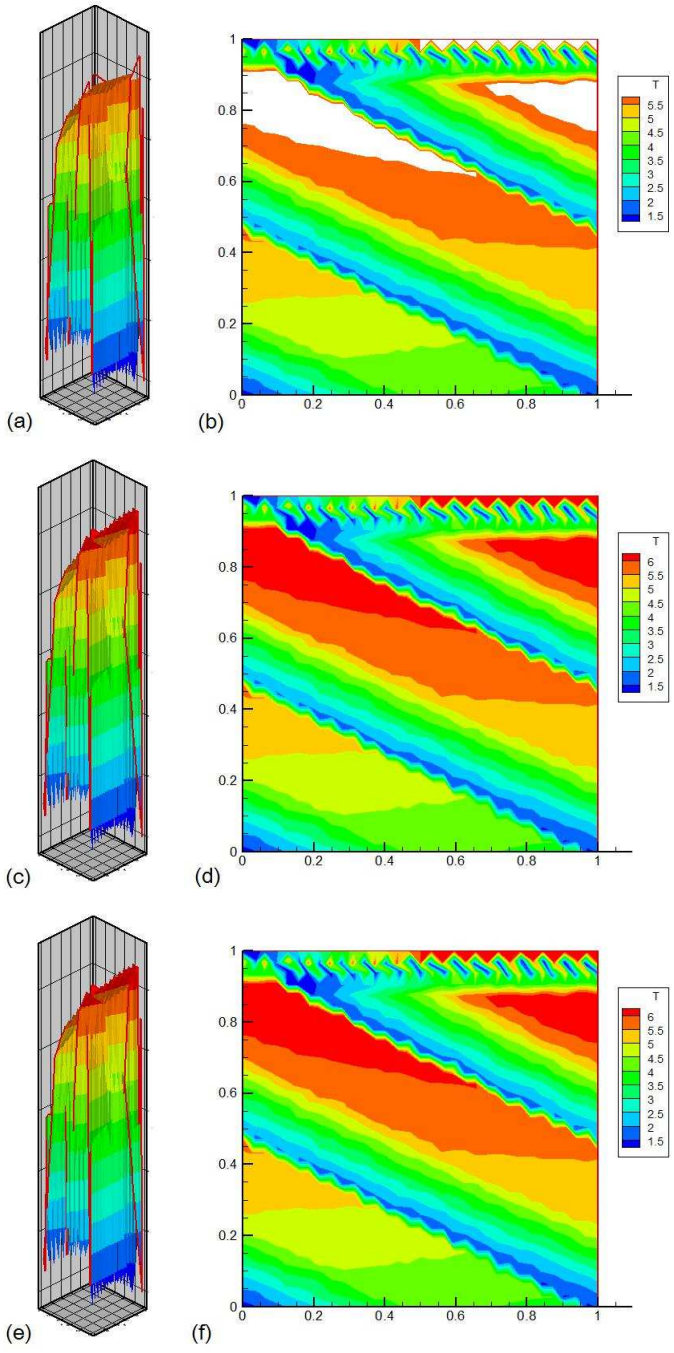


Figure 4: Test problem 2. Solution profiles (left) and solution colormaps (right) of (a),(b) original FEM; (c),(d) Method 1; (e),(f) Method 2. Mesh scale is 64×64. The regions that violate the DMP are indicated in white.

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

Table 5: Test problem 2. The convergence study for Method 2, $u_{\min}=1.0$, $u_{\max}=6.5$.

h	1/16	1/32	1/64	1/128	1/256
$\|\|u-u_h\|\|_2$	6.48E-3	2.82E-3	1.26E-3	5.01E-4	2.32E-4
convergence order	-	1.19	1.15	1.30	1.13
$\|\|u-u_h\|\|_\infty$	5.68E-2	2.79E-2	1.16E-2	5.28E-3	2.41E-3
convergence order	-	1.02	1.26	1.14	1.12

Table 6: Test problem 2. The CPU time of nonlinear iterations for Method 1 (in Ω) and Method 2 (in Ω_1,Ω_2).

h	1/16	1/32	1/64	1/128	1/256
cpu time(s) Ω	2.14	51.35	902.70	15725.22	273936.57
cpu time(s) Ω_1	0.25	5.17	86.82	1560.04	27176.24
cpu time(s) Ω_2	0.87	12.92	242.38	4056.10	70658.24

6.3 Test problem 3

The diffusion tensor D is defined as follows

$$D(x,y)=\begin{cases} D_1, & x<0.5, \\ D_2, & x>0.5, \end{cases} \quad D_1=\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad D_2=\begin{pmatrix} 100 & 0 \\ 0 & 1 \end{pmatrix}.$$

We use the same exact solution $u(x,y)$ in (6.1), $g=0$. Choose $\theta=0.02$ in the modified Neumann-Neumann method, The results of convergence study are presented in Table 7 and Table 8, respectively. The CPU time of the nonlinear iteration of the two methods is presented in Table 9 and Fig. 5. The numerical solution profiles and the colormaps are shown in Fig. 6.

Table 7: Test problem 3. The convergence study for Method 1, $u_{\min}=1.0$, $u_{\max}=6.5$.

h	1/16	1/32	1/64	1/128	1/256
$\|\|u-u_h\|\|_2$	6.16E-3	1.80E-3	1.92E-3	6.36E-4	3.42E-4
convergence order	-	1.77	-0.09	1.59	0.89
$\|\|u-u_h\|\|_\infty$	3.17E-2	1.56E-2	1.91E-2	7.72E-3	4.21E-3
convergence order	-	1.02	-0.29	1.30	0.87

Table 8: Test problem 3. The convergence study for Method 2, $u_{\min}=1.0$, $u_{\max}=6.5$.

h	1/16	1/32	1/64	1/128	1/256
$\|\|u-u_h\|\|_2$	4.65E-3	1.46E-3	5.65E-4	2.59E-4	1.04E-4
convergence order	-	1.67	1.36	1.12	1.31
$\|\|u-u_h\|\|_\infty$	6.47E-2	2.87E-2	1.18E-2	5.43E-3	2.46E-3
convergence order	-	1.17	1.28	1.11	1.14

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

Table 9: Test problem 3. The CPU time of nonlinear iterations for Method 1 (in Ω) and Method 2 (in Ω_1,Ω_2).

h	1/16	1/32	1/64	1/128	1/256
cpu time(s) Ω	2.29	68.56	1274.17	13669.64	168136.57
cpu time(s) Ω_1	0.24	5.14	86.82	1345.85	21816.22
cpu time(s) Ω_2	0.72	13.03	210.62	3336.93	52456.53

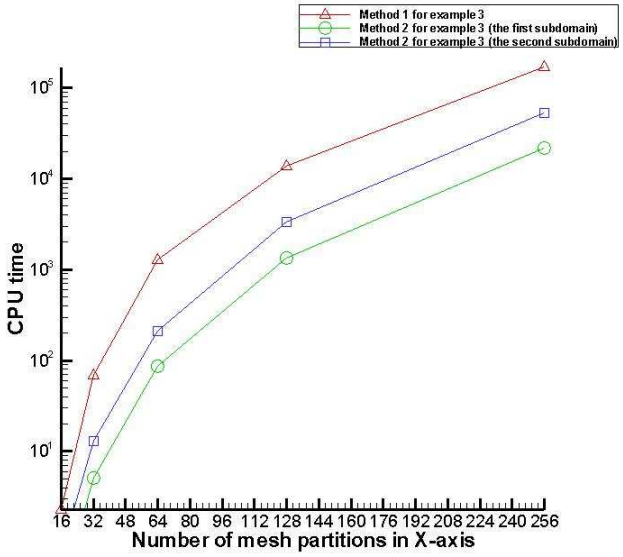


Figure 5: Test problem 3. The CPU time of nonlinear iterations for the two methods.

6.4 Test problem 4

Consider the multi-subdomain problem. Assume $g=0$, and the forcing term is

$$q=\begin{cases} 1, & \text{if } (x,y)\in[7/18,11/18]^2, \\ 0, & \text{otherwise.} \end{cases}$$

The domain Ω is partitioned into four square subdomains $\Omega_i, i=1,\cdots,4$. The diffusion tensor is given by

$$D(x,y)=\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix}\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

The parameters k_1,k_2 and θ are different in the four subdomains, see Fig. 7.

The numerical solution profiles and the colormaps are shown in Fig. 8. The CPU time of the nonlinear iteration of the two methods is presented in Table 10. From the numerical experiments, we can see that both of the two Methods satisfy the discrete maximum principle and the total CPU time of the nonlinear iteration in Method 2 is reduced remarkably.

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

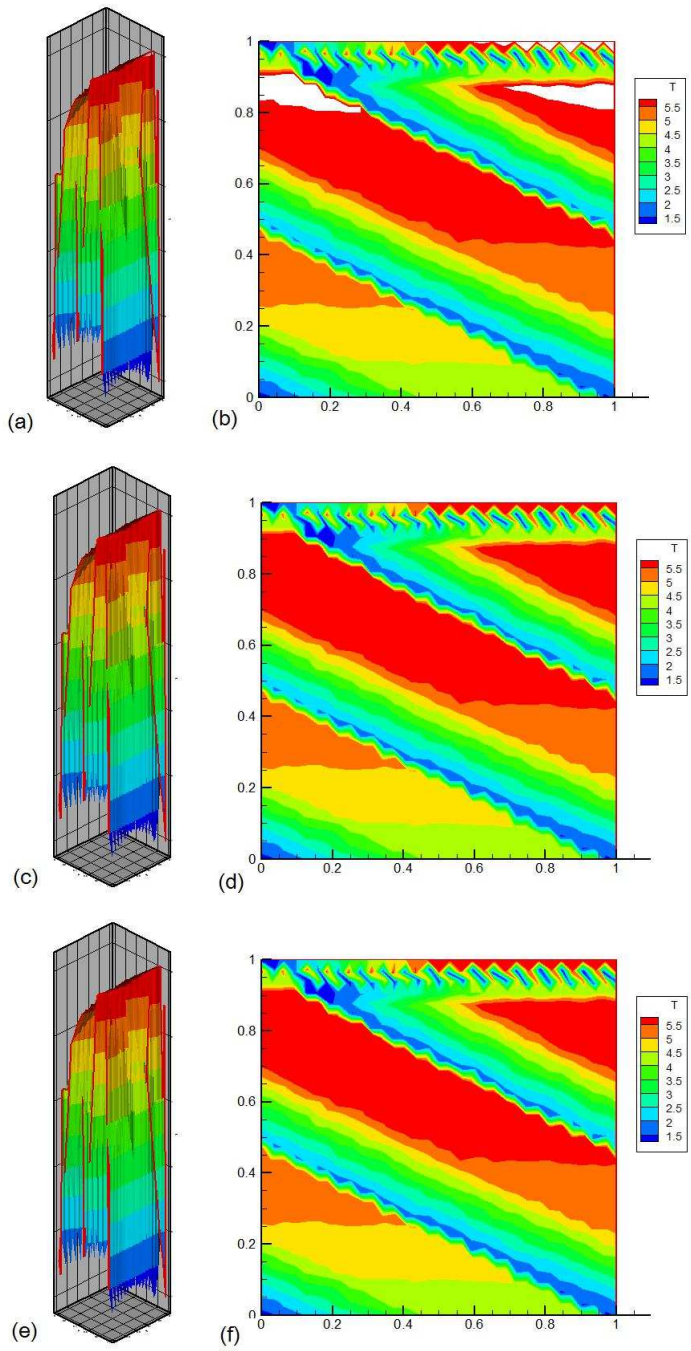


Figure 6: Test problem 3. Solution profiles (left) and solution colormaps (right) of (a),(b) original FEM; (c),(d) Method 1; (e),(f) Method 2. Mesh scale is 64×64. The regions that violate the DMP are indicated in white.

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

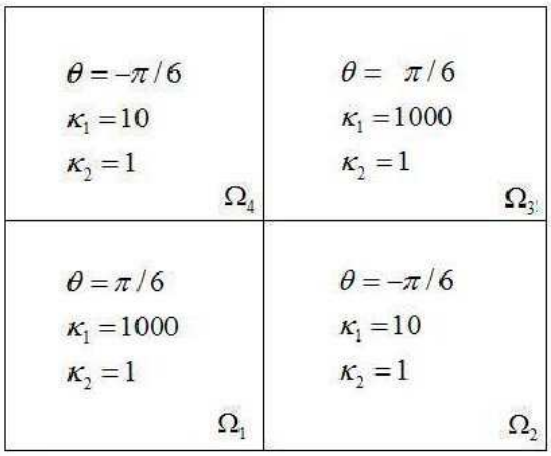


Figure 7: Test problem 4. The parameters in different subdomains.

Table 10: Test problem 4. The CUP time of nonlinear iterations for Method 1 (in Ω) and Method 2 (in Ω_1, Ω_2, Ω_3, Ω_4).

h	1/16	1/32	1/64	1/128
cpu time(s) Ω	23.65	1073.99	50884.65	1274222.10
cpu time(s) Ω_1	1.49	26.62	521.04	5028.03
cpu time(s) Ω_2	0.28	4.83	75.40	706.00
cpu time(s) Ω_3	1.49	26.62	521.04	5028.03
cpu time(s) Ω_4	0.28	4.83	75.40	706.00

Remark 6.1. In the numerical experiments, we simply choose $\theta = 0.02$ in the modified Neumann-Neumann method. Comparing with the nonlinear iteration in the constrained finite element method, the computational cost of the modified Neumann-Neumann method is very little. So, this simple choice of the relax parameter θ will not influence the efficiency of our method.

Remark 6.2. In the numerical experiments, the discontinuity of the coefficients is known in advance. We handle the discontinuity by aligning the interfaces of the subdomains with the coefficient discontinuity. For the situations of the discontinuity is unknown, the accuracy of locating the interfaces will influence the accuracy of our method. As we know, the lumped-mass L_2 projection in the slope-limited scheme samples data from both sides of an internal interface. In the domain decomposition method, we divide the whole domain aligning with the interfaces. In the case of a heterogeneous diffusion tensor, if the location of the interface is inaccurate, the gradient is still discontinuous in the subdomains. When the constrained finite element method is used in each subdomain, the gradient recovery is inaccurate, which will degrade the accuracy of the method. There

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

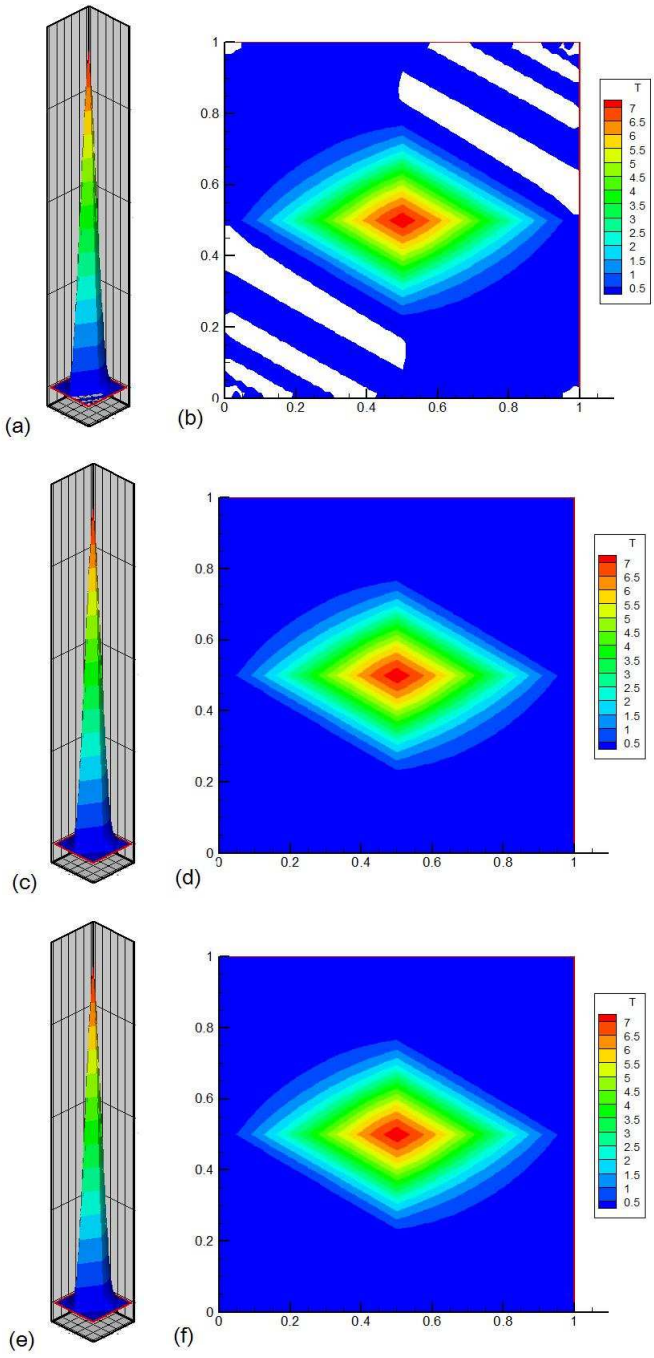


Figure 8: Test problem 4. Solution profiles (left) and solution colormaps (right) of (a),(b) original FEM; (c),(d) Method 1; (e),(f) Method 2. Mesh scale is 64×64. The regions that violate the DMP are indicated in white.

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

are some techniques to locate the interface, such as the Volume of Fluid (VOF) methods and the Level Set methods. The VOF methods can capture very sharp interface, but the interface reconstruction is complex. In the Level Set methods, the reconstruction of the interface is not needed, and these methods can simulate the smooth interface well. Besides, the particle Level Set method is suited for the case of large gradient interface deformation. One can choose a proper method to locate the interfaces according to the practical computation.

7 Summary and conclusions

In this paper, we combine the constrained finite element method [8] with the domain decomposition method to enforce the solutions satisfying the discrete maximum principle. To get the information on the interface, we address the modified Neumann-Neumann algorithm for the discontinuous diffusion tensors. Then, the constrained finite element method is used in each subdomain to get the solution. The numerical experiments show that our method not only achieve more than one-order accuracy, but also reduce the CPU time remarkably. Moreover, the basic idea of our method can be generalized to multi-subdomain problems without essential difficulty. By the way, we only consider the stationary diffusion problems in the present paper. How to make the solutions of time-dependent diffusion problems satisfy the discrete maximum principle is our future work.

Acknowledgments

The first author was supported by General Program of Science and Technology Development Project of Beijing Municipal Education Commission KM201310011006, Program of the Young People of Outstanding Ability for the Construction of the Teachers Procession YETP1445, the National Natural Science Foundation 11401015, 91130015. The second author was supported by the National Natural Science Foundation 11171036.

References

[1] E. Burman and A. Ern, Discrete maximum priciple for Galerkin approximation of the Laplace operator on arbitrary meshes, Comptes Rendus Mathematique Academie des Sciences. Paris, 338 (2004), 641-646.
[2] P. G. Ciarlet, Discrete maximum principle for finite-difference operators, Aeq. Math. 4 (1970), 338-352.
[3] P. G. Ciarlet and P. A. Raviart, Maximum principle and convergence for the finite element method, Comput. Methods Appl. Mech. Eng. 2 (1973), 17-31.
[4] R. S. Varga, Matrix Iterative Analysis, Prentice-Hall, Englewood Cliffs, 1962.
[5] R. Liska and M. Shashkov, Enforcing the discrete maximum principle for linear finite element solutions of second-order elliptic problems, Comm. Comput. Phys., 3 (2008), 852-877.

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press

[6] K. Nakshatrala and A. Valocchi, Non-negative mixed finite element formulation for a tensorial diffusion equation, J. Comput. Phy., 228 (2009), 6726-6752.
[7] H. Nagarajan and K. Nakshatrala, Enforcing the non-negativity constraint and maximum principles for diffusion with decay on general computational grids, Int. J. Numer. Meth. Flu., 67 (2011), 820-847.
[8] D. Kuzmin, M. Shashkov and D. Svyatskiy, A constrained finite element method satisfying the discrete maximum priciple for anisotropic diffusion problems, J. Comput. Phys., 228 (2009), 3448-3463.
[9] D. Y. Li, H. S. Shui and M. J. Tang, On the finite difference scheme of two-dimensional parabolic equation in a non-recangular mesh, Int. J. Numer. Meth. Comput. Appl., 1 (1980), 217-224.
[10] W. Z. Huang and A. Kappen, A study of cell-center finite volume methods for diffusion equations, Mathematics Research Report, University of Kansas, Lawrence KS66045, 98-10-01.
[11] I. Aavatsmark, An introduction to multipoint flux approximations for quadrilateral grids, Computational Geosciences; 6 (2002), 405-432.
[12] H. Friis and M. Ewards, A family of mp finite-volume schemes with full presseure support for general tensor oressure equation on cell-centered triangular grids, J. Comput.Phys., 230 (2011), 205-231.
[13] J. Nordbotten and G. Eigestad, Discritization on quadrilateral grids with improved monotoncity properties, J. Comput. Phys., 203 (2005), 744-760.
[14] K. Lipnikov, M. Shashkov and D. Svyatskiy, The mimetic finite difference discretization of diffusion problem on unstructured polyhedral meshes, J. Comput. Phys., 211 (2006), 473-491.
[15] K. Lipnikov, M. Shashkov and I. Yotov, Local flux mimetic finite difference methods, Numer. Math., 112 (2009), 115-152.
[16] M. H. Protter and H. F. Weinberger, Maximum principlrs in Differential Equations, Prentice-Hall, 1967.
[17] J. Karatson and S. Korotov, Discrete maximum principles for finite element solutions of nonlinear elliptic problems with mixed boundary conditions, Numer. Math. 99 (2005), 669-698.
[18] A. Toselli and O. Widlund, Domain Decomposition Methods-Algorithms And Theory, Springer, Berlin, 2004.
[19] D. Kuzmin and M. Moller, Algebraic flux correction I. Scalar conservation laws, in: D. Kuzmin, R. Lohner, S. Turek(Eds.), Flux-Corrected Transport: Principles Algorithms and Applications, Springer Berlin, 2005, 155-206.
[20] D. Kuzmin, On the design of general-purpose flux limiters for implicit FEM with a consistent mass matrix. I. Scalar convection, J. Comput. Phys. 219 (2006), 513-531.
[21] Z. Q. Sheng and G. W. Yuan, A nine point scheme for the approxiamtion of diffusion operators on distorted quadrilateral meshes, SIAM J. Sci. Comput., 30 (2008), 1341-1361.
[22] G. W. Yuan and Z. Q. Sheng, Analysis of accuracy of a finite volume scheme for diffusion equations on distorted meshes, J. Comput. Phy., 224 (2007), 1170-1189.
[23] T. Lv, J. M. Shi and Z. B. Lin, Domain Decomposition Methods-The New Technique For The Numerical Analysis Of Partial Differential Equations, Science Press, China, 1992.
[24] G. W. Yuan and Z. Q. Sheng, Monotone finite volume schemes for diffusion equations on polygonal meshes, J. Comput. Phy., 227 (2008), 6288-6312.

https://doi.org/10.4208/cicp.120914.311214a Published online by Cambridge University Press