Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2012, Article ID 187634, 23 pages
doi:10.1155/2012/187634

Research Article

The Finite Volume Formulation for 2D Second-Order Elliptic Problems with Discontinuous Diffusion/Dispersion Coefficients

Stefano Ferraris, Ivan Bevilacqua, Davide Canone, Davide Pognant, and Maurizio Previati

Dipartimento di Economia e Ingegneria Agraria, Forestale e Ambientale (DEIAFA) Sez. Idraulica,
Via Leonardo da Vinci 44, 10095 Grugliasco, Italy

Correspondence should be addressed to Stefano Ferraris, stefanof32@gmail.com

Received 18 July 2011; Revised 2 November 2011; Accepted 17 November 2011

Academic Editor: Moran Wang

We propose a finite volume method for the numerical resolution of two-dimensional steady diffusion problems with possibly discontinuous coefficients on unstructured polygonal meshes. Our numerical method is cellcentered, secondorder accurate on smooth solutions and based on a special numerical treatment of the diffusion/dispersion coefficients that makes its application possible also when such coefficients are discontinuous. Numerical experiments confirm the convergence of the numerical approximation and show a good behavior on a set of benchmark problems in two space dimensions.

1. Introduction

In the last decades, finite volume methods have been greatly successful in solving engineering models of flows in porous media on complex geometrical domain because the finite volume formulation works on general polygonal and polyhedral meshes. This great mesh flexibility is now combined with a strong theoretical foundation, that is, convergence analysis and error estimates are available on the simplest mathematical models [1]. In this work, we address a version of the finite volume method that is popularly known as *"the diamond scheme"* and was originally presented for the advection-diffusion equation in two dimensions in [2–4] and successively extended to convection-dominated problems in [5, 6] and nonlinear flow problems in partially saturated porous media [7].

The numerical treatment of both diffusion/dispersion flux and of the partially saturated flow is based on averaging one-sided gradients and diffusion (conductivity) tensors. Many papers addressed the issue of conductivity averaging both in finite differences and in finite elements. A seminal work [8] started with a finite difference application to the unsaturated flow equation. Later on they, dealt with the problem in a finite volume scheme [9]. Recent reviews on this topic can be found in [10–12]. They also made performance comparisons using different averages. The latter paper shows in conclusion that their scheme, namely, an averaging schemes based on Darcian mean principle used in the framework of either vertex-centered or cell-centered approach compare favorably to other methods for a range of test cases. They compared their method with the one of [11]. Aside of the numerical comparisons, in [13] it is remarked that the usage of the harmonic average can be motivated on the basis of physical arguments.

Another reason for dealing with the tensor averaging issue is related to the scale problems. Typically, the Darcy equation is written at a scale much smaller than the scale of practical interest [14, 15]. This gives boost to new numerical techniques to able to deal with the heterogeneity of hydrodynamic parameters. One possibility is to obtain either analytically or empirically explicit equations for the scale of interest, eliminating other scales in the problem [16].

Several methods have been recently developed, such as, the heterogeneous multiscale method (HMM) [16] and the multiscale finite volume element method [17]. Babuška in the 70s motivated the multiscale finite element method [18–20]. Multiscale methods have been proposed also in the case of saturated flows in heterogeneous porous media and also applied the multiscale finite volume element method to the Richards equation [21, 22]. In [14], a multiscale method based on the framework of HMM was recently extended to solve the Richards equation with random coefficients.

Another feature of their numerical scheme is the formula estimating the macroscopic flux in which the unsaturated hydraulic conductivity can be calculated as a diagonal tensor. In particular, it is worth noting that a finite volume high-order accurate approximation of the pressure head field also allows one to achieve a better resolution in the approximation of other important fields like the components of the hydraulic conductivity tensor at the mesh vertices [7]. For the sake of simplicity, in this work it was considered the steady flow in a layered porous medium in the presence of a source term. It is the Poisson equation which is suitable to test new schemes which will be applied in further work to the more general unsaturated transient cases.

We propose a novel technique that automatically adjusts when the diffusion tensor is discontinuous across a mesh interface shared by two adjacent cells. This technique is general and can be easily implemented in any finite volume scheme that has an explicit numerical flux and may result in a particularly efficient strategy for the numerical resolution of problems where both the diffusive/dispersive and convective phenomena are simultaneously significant. In fact, in such problems high-resolution finite volume methods are normally coupled with mixed finite elements following the criterion of choosing the best available technique in accordance with the nature of the equation to be discretized. In fact, the $\mathbb{RT}_0 - \mathbb{P}_0$ method is more suitable to the numerical discretization of the diffusive part of the model, and the finite volume method gives an accurate and stable discretization of the convective part of the model, even in presence of strong convective fields. This approach was proved successful in the numerical modeling of oil reservoir problems [23] and of groundwater flow and transport of contaminants in porous media, combared with [24–26]. Moreover, other different engineering areas may benefit from this new technique, such as [27–38].

The outline of the paper is as follows. In Section 2, we introduce the mathematical model and discuss the formulation of the finite volume methods based on vertex reconstruction. In Section 3, we present the numerical treatment of the diffusion tensor. In Section 4, we confirm the theoretical results with numerical experiments. In Section 5 we offer final remarks and conclusions.

2. The Mathematical Model and the Finite Volume Formulation

Let Ω be a polygonal domain with boundary Γ.

We consider the steady diffusion problem for the scalar solution field u given by

$$-\operatorname{div}(\Lambda \nabla u) = f \quad \text{in } \Omega, \tag{2.1}$$

$$u = g \quad \text{on } \Gamma, \tag{2.2}$$

where Λ is the diffusion tensor describing the material properties, f is the forcing term and g the boundary function that defines nonhomogeneous Dirichlet conditions on the boundary Γ. Under suitable regularity assumptions on f, g, Λ, and Ω, it turns out that the diffusion problem is mathematically wellposed, a unique solution exists [39] and such a solution is continuously dependent on the model data. In particular, due to the elliptic nature of the model, the diffusion tensor $\Lambda(\mathbf{x})$ is normally represented by a strictly positive definite matrix for every $\mathbf{x}$ in Ω. The components of the diffusion tensor may be discontinuous in the computational domain; in such a case and without loss of generality, we assume that the mesh is conforming with each discontinuity of Λ.

The numerical approximation to (2.1)–(2.2) is performed on a sequence of polygonal partitions $\{\Omega_h\}_h$ of the domain Ω. For any mesh Ω_h, the subscripted label h is the mesh size and is defined by

$$h = \max_{\mathsf{K} \in \Omega_h} h_{\mathsf{K}}, \qquad h_{\mathsf{K}} = \sup_{\mathbf{x}', \mathbf{x}'' \in \mathsf{K}} \left| \mathbf{x}' - \mathbf{x}'' \right|, \tag{2.3}$$

where h_{K} is the diameter of the polygonal cell $\mathsf{K} \in \Omega_h$. We also label the numerical solution u_h calculated on the mesh Ω_h by h. Also,

(i) a generic *mesh vertex* is denoted by V and its coordinate vector by $\mathbf{x}_{\mathsf{V}}$. We will also find it convenient to introduce a local numbering of the vertices, for example, V_i, and to ease notation to denote the vertex position of the i-th vertex by $\mathbf{x}_i$ (instead of $\mathbf{x}_{\mathsf{V}_i}$);

(ii) a generic *cell interface* or a *boundary edge* is denoted by σ, its center (i.e., its edge midpoint) by $\mathbf{x}_\sigma$, and its measure (edge length) by $|\sigma|$;

(iii) a generic *polygonal cell* is denoted by K its measure (area) by $|\mathsf{K}|$, its center of gravity by $\mathbf{x}_{\mathsf{K}}$, and its boundary by $\partial \mathsf{K}$.

The orientation of each mesh interface σ is reflected by its unit normal vector $\mathbf{n}_\sigma$, which is fixed once and for all. For any mesh cell K and any face σ of the polygonal boundary $\partial \mathsf{K}$, we define the unit normal vector $\mathbf{n}_{\mathsf{K}\sigma}$ that points out of K and we also use the notation $\mathbf{N}_{\mathsf{K}\sigma} = |\sigma| \mathbf{n}_{\mathsf{K}\sigma}$.

Let the algebraic vector $u_h = (u_{\mathsf{K}})_{\mathsf{K}\in\Omega_h}$ be the *numerical solution*, where each u_{K} approximates the cell average of the scalar solution u over the cell K. The finite volume scheme for u_h on the computational mesh Ω_h reads

$$-\sum_{\sigma\in\partial\mathsf{K}} \Lambda_\sigma \nabla_\sigma u_h \cdot \mathbf{N}_{\mathsf{K}\sigma} = f_{\mathsf{K}}|\mathsf{K}| \quad \forall\mathsf{K} \in \Omega_h, \tag{2.4}$$

where Λ_σ is an evaluation of Λ at face σ taken from the side of K; ∇_σ is the discrete face gradient built inside the diamond cell centered at face σ; $\mathbf{N}_{\mathsf{K},\sigma}$ is the geometric vector perpendicular to σ, pointing from K to L, and with lenght $|\mathbf{N}_{\mathsf{K}\sigma}| = |\sigma|$; f_{K} is the cell average of the right-hand side term f over the cell K:

$$f_{\mathsf{K}} := \frac{1}{|\mathsf{K}|}\int_{\mathsf{K}} f(\mathbf{x})dV. \tag{2.5}$$

Let Λ_{K} and Λ_{L} be first-order approximations of the diffusion tensor Λ within the control volumes K and L. For example, we can take $\Lambda_{\mathsf{K}} := \Lambda(\mathbf{x}_{\mathsf{K}})$ and $\Lambda_{\mathsf{L}} := \Lambda(\mathbf{x}_{\mathsf{L}})$, where $\mathbf{x}_{\mathsf{K}}$ and $\mathbf{x}_{\mathsf{L}}$ are internal points (not necessarily the barycenters) of K and L, respectively. Now, we can consider either $\Lambda_\sigma = \Lambda_{\mathsf{K}}$ inside K and $\Lambda_\sigma = \Lambda_{\mathsf{L}}$ inside L or a suitable mean of these two tensors. In particular, we can deal with the *arithmetic mean*:

$$\Lambda_\sigma = \frac{1}{2}(\Lambda_{\mathsf{K}} + \Lambda_{\mathsf{L}}), \tag{2.6}$$

or with the *harmonic mean*:

$$\Lambda_\sigma^{-1} = \frac{1}{2}\left(\Lambda_{\mathsf{K}}^{-1} + \Lambda_{\mathsf{L}}^{-1}\right). \tag{2.7}$$

Both approaches deserve a special care when Λ is discontinuous across σ and a proper definition of the discrete gradient $\nabla_\sigma u_h$ is also needed in order to preserve the property of flux conservation.

To define the numerical gradient, a special control volume is built around this interface, which has a quadrilateral shape in two dimensions and is named *"diamond cell"*. The geometry of a diamond cell is shown in Figure 1, which plots the diamond cell D centered at the mesh face σ with vertices $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ and shared by the cells with centers $\mathbf{x}_{\mathsf{K}}$ and $\mathbf{x}_{\mathsf{L}}$. The diamond cell D can also be seen as the union of the subdiamonds D_{K} and D_{L}, which are the triangular cells sharing σ as common base and having, respectively, vertices $\mathbf{x}_{\mathsf{K}}$ and $\mathbf{x}_{\mathsf{L}}$, the centers of gravity of cells K and L. The four vectors $\mathbf{N}_{\mathsf{K}i}$, $\mathbf{N}_{\mathsf{L}i}$, $\mathbf{N}_{\mathsf{L}i+1}$, and $\mathbf{N}_{\mathsf{K}i+1}$ shown in Figure 1 are respectively orthogonal to the four boundary faces $\sigma_{\mathsf{K}i}$, $\sigma_{\mathsf{L}i}$, $\sigma_{\mathsf{L}i+1}$, and $\sigma_{\mathsf{K}i+1}$, and their lenght is equal to the lenght of the corresponding face. Instead, when σ is a boundary face, thus defined by $\sigma = \partial\mathsf{K} \cap \Gamma$ (Γ being the boundary of the computational domain Ω), the diamond cell associated to σ coincides with D_{K}, that is, it is the triangle defined by σ and the vertex $\mathbf{x}_{\mathsf{K}}$.

The numerical diffusive flux is built by using a constant approximation of the solution gradient and an evaluation of the diffusion tensor Λ within the diamond cell, as discussed before. Let us give the formulas for an internal mesh face σ, that we suppose to be shared by

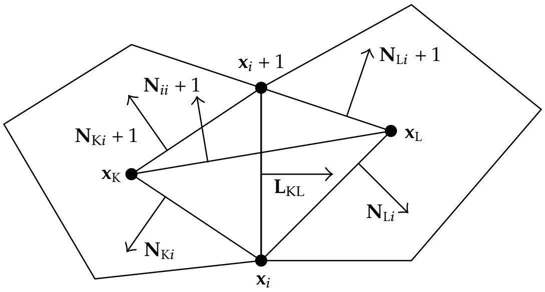

$\mathbf{x}_i + 1$
$\mathbf{N}_{\mathrm{L}i} + 1$
$\mathbf{N}_{ii} + 1$
$\mathbf{N}_{\mathrm{K}i} + 1$
$\mathbf{x}_\mathrm{L}$
$\mathbf{x}_\mathrm{K}$
$\mathbf{L}_{\mathrm{KL}}$
$\mathbf{N}_{\mathrm{L}i}$
$\mathbf{N}_{\mathrm{K}i}$
$\mathbf{x}_i$

Figure 1: Geometry of the *2D diamond* cell.

two cells K and L. All these formulas can be easily adapted to the case of a boundary face by taking $\mathbf{x}_\mathsf{L}$ equal to the center of gravity of face σ.

Using the Green-Gauss theorem yields:

$$\nabla_\sigma u_h \approx \frac{1}{|\mathsf{D}|}\int_\mathsf{D} \nabla u(\mathbf{x})dV = \frac{1}{|\mathsf{D}|}\int_{\partial \mathsf{D}} u(\mathbf{x})\mathbf{n}(\mathbf{x})dS = \frac{1}{|\mathsf{D}|}\sum_{\sigma\in\partial\mathsf{D}}\int_\sigma u(\mathbf{x})\mathbf{n}(\mathbf{x})dS, \tag{2.8}$$

where $\mathbf{n}$ is the unit vector orthogonal to $\sigma \in \partial\mathsf{D}$ and pointing out of D. If the restriction of u to the face σ of $\partial\mathsf{D}$ is an affine function, the boundary integral on $\partial\mathsf{D}$ in (2.8) only depends on the values of u at the vertexes of D and the constant vector provided by the formulas that we derive below must coincide with the gradient of u.

The Gauss-Green theorem applied to the solution gradient ∇u integrated on the 2D diamond cell D of Figure 1 yields:

$$\begin{aligned}\sum_{\sigma\in\partial\mathsf{D}}\int_\sigma u(\mathbf{x})\mathbf{n}(\mathbf{x})dS = \frac{1}{2}\big((u(\mathbf{x}_\mathsf{K}) + u(\mathbf{x}_i))\mathbf{N}_{\mathsf{K}i} + (u(\mathbf{x}_i) + u(\mathbf{x}_\mathsf{L}))\mathbf{N}_{\mathsf{L}i} \\ +(u(\mathbf{x}_\mathsf{L}) + u(\mathbf{x}_{i+1}))\mathbf{N}_{\mathsf{L}i+1} + (u(\mathbf{x}_{i+1}) + u(\mathbf{x}_\mathsf{K}))\mathbf{N}_{\mathsf{K}i+1}\big).\end{aligned} \tag{2.9}$$

Let us introduce the vectors $\mathbf{N}_\mathsf{KL}$ and $\mathbf{N}_{ii+1}$, which are, respectively, orthogonal to the edge connecting $\mathbf{x}_\mathsf{K}$ to $\mathbf{x}_\mathsf{L}$ and $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$, and whose lenght are equal to the length of these edges. Using the geometric relations $\mathbf{N}_\mathsf{KL} = -(\mathbf{N}_{\mathsf{K}i} + \mathbf{N}_{\mathsf{L}i})$ and $\mathbf{N}_\mathsf{KL} = -(\mathbf{N}_{\mathsf{K}i} + \mathbf{N}_{\mathsf{L}i})$ into (2.9) and rearranging the terms yields:

$$\int_{\partial\mathsf{D}} \nabla u(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})dS = \frac{1}{2}\big((u(\mathbf{x}_{i+1}) - u(\mathbf{x}_i))\mathbf{N}_\mathsf{KL} + (u(\mathbf{x}_\mathsf{L}) - u(\mathbf{x}_\mathsf{K}))\mathbf{N}_{ii+1}\big). \tag{2.10}$$

To derive the gradient formula, we replace the function values $u(\mathbf{x}_\mathsf{K})$ and $u(\mathbf{x}_\mathsf{L})$ with the cell unknowns u_K and u_L, respectively, and the function values $u(\mathbf{x}_i)$ and $u(\mathbf{x}_{i+1})$ with the corresponding nodal unknowns u_i and u_{i+1}. We obtain

$$(u_h) := \frac{1}{2|\mathsf{D}|}\left(\nabla_\mathsf{KL}(u_\mathsf{L} - u_\mathsf{K})\mathbf{N}_\mathsf{KL} + (u_{i+1} - u_i)\mathbf{N}_{ii+1}\right). \tag{2.11}$$

which is a constant approximation of the solution gradient ∇u within the quadrilateral cell D.

With any vertex $\mathbf{x}_i$ of the mesh Ω_h, we associate the reconstructed vertex value $u_i = \sum_{\mathsf{K}\in\Omega_i}\omega_{\mathsf{K}i}u_{\mathsf{K}}$, where $\Omega_i = \{\mathsf{K} : \mathbf{x}_i \in \mathsf{K}\}$ denotes the subset of the mesh cells which share the vertex $\mathbf{x}_i$. The interpolation weights $\omega_{\mathsf{K}i}$ are assumed to verify the consistency relations [3]:

$$\sum_{\mathsf{K}\in\Omega_i}\omega_{\mathsf{K}i} = 1, \qquad \sum_{\mathsf{K}\in\Omega_i}\omega_{\mathsf{K}i}(\mathbf{x}_i - \mathbf{x}_{\mathsf{K}}) = 0. \tag{2.12}$$

Alternative constructions of the interpolation weights are available in the literature, see, for instance, [40]. The main advantage offered by these alternative choices concerns the control of positivity of the weights, which turns out to be significant when we are aimed at developing a numerical method with a maximum or minimum principle. However, as this topic is out of the scope of the current work, we will not pursue it anymore.

The interpolation weights $\omega_{\mathsf{K}i}$ are obtained by solving the reconstruction problem that approximates the cell-averaged data set $\{(\mathbf{x}_{\mathsf{K}},\ u_{\mathsf{K}})_{\mathsf{K}\in\Omega_i}\}$ by the affine function:

$$\tilde{u}_i(\mathbf{x}) = \alpha + \boldsymbol{\beta}\cdot(\mathbf{x}_{\mathsf{K}} - \mathbf{x}_i) \quad \text{for } \mathbf{x}\in\mathcal{V}_i \tag{2.13}$$

on the covolume $\mathcal{V}_i = \bigcup_{\mathsf{K}\in\Omega_i}\mathsf{K}$ and in a least square sense, compared to [3, 41]. The reconstructed value at vertex $\mathbf{x}_i$ is now given by taking $u_i = \tilde{u}_i(\mathbf{x}_i) = \alpha$. The coefficients $(\alpha,\boldsymbol{\beta})$ are the minimizers of the least squares functional:

$$\mathcal{J}_i(\alpha,\boldsymbol{\beta}) = \sum_{\mathsf{K}\in\Omega_i}(\alpha + \boldsymbol{\beta}\cdot(\mathbf{x}_{\mathsf{K}} - \mathbf{x}_i) - u_{\mathsf{K}})^2. \tag{2.14}$$

Imposing the zero gradient condition, that is, $\nabla_{\alpha,\beta}\mathcal{J}(\alpha,\boldsymbol{\beta}) = 0$, yields a linear system for the coefficients $(\alpha,\boldsymbol{\beta})$, whose solution returns the interpolation weights. For completeness of exposition, we briefly review the derivation of the weight formula given in [41]. Let $\boldsymbol{\beta} = (\beta_x,\beta_y)$; a straightforward calculation yields:

$$\begin{aligned}
\frac{\partial\mathcal{J}_i}{\partial\alpha} &= 2\sum_{\mathsf{K}\in\Omega_i}(\alpha + \boldsymbol{\beta}\cdot(\mathbf{x}_{\mathsf{K}} - \mathbf{x}_i) - u_{\mathsf{K}}) = 0,\\
\frac{\partial\mathcal{J}_i}{\partial\beta_x} &= 2\sum_{\mathsf{K}\in\Omega_i}(\alpha + \boldsymbol{\beta}\cdot(\mathbf{x}_{\mathsf{K}} - \mathbf{x}_i) - u_{\mathsf{K}})(x_{\mathsf{K}} - x_i) = 0,\\
\frac{\partial\mathcal{J}_i}{\partial\beta_y} &= 2\sum_{\mathsf{K}\in\Omega_i}(\alpha + \boldsymbol{\beta}\cdot(\mathbf{x}_{\mathsf{K}} - \mathbf{x}_i) - u_{\mathsf{K}})(y_{\mathsf{K}} - y_i) = 0.
\end{aligned} \tag{2.15}$$

Let m_i be the number of cells of Ω_i; we reformulate the linear system written above as

$$\sum_{\mathsf{K}\in\Omega_i} m_i\alpha + \sum_{\mathsf{K}\in\Omega_i} (x_\mathsf{K} - x_i)\beta_x + \sum_{\mathsf{K}\in\Omega_i} (y_\mathsf{K} - y_i)\beta_y = \sum_{\mathsf{K}\in\Omega_i} u_\mathsf{K}, \tag{2.16}$$

$$\sum_{\mathsf{K}\in\Omega_i} (x_\mathsf{K} - x_i)\ \alpha + \sum_{\mathsf{K}\in\Omega_i} (x_\mathsf{K} - x_i)^2\beta_x + \sum_{\mathsf{K}\in\Omega_i} (x_\mathsf{K} - x_i)(y_\mathsf{K} - y_i)\beta_y = \sum_{\mathsf{K}\in\Omega_i} (x_\mathsf{K} - x_i)u_{\mathsf{K},} \tag{2.17}$$

$$\sum_{\mathsf{K}\in\Omega_i} (y_\mathsf{K} - y_i)\ \alpha + \sum_{\mathsf{K}\in\Omega_i} (x_\mathsf{K} - x_i)(y_\mathsf{K} - y_i)\beta_x + \sum_{\mathsf{K}\in\Omega_i} (y_\mathsf{K} - y_i)^2\beta_y = \sum_{\mathsf{K}\in\Omega_i} (y_\mathsf{K} - y_i)u_\mathsf{K}. \tag{2.18}$$

For convenience, we consider a suitable local numbering of the cells of $\mathcal{V}_i$, for example, $\mathcal{V}_i = \mathsf{K}_1 \cup \mathsf{K}_2 \cup \ldots \mathsf{K}_{m_i}$, and we introduce the two matrices:

$$\begin{gathered}\mathsf{A} = \begin{pmatrix} m_i & \sum_{\mathsf{K}\in\Omega_i} (x_\mathsf{K} - x_i) & \sum_{\mathsf{K}\in\Omega_i} (y_\mathsf{K} - y_i) \\ \sum_{\mathsf{K}\in\Omega_i} (x_\mathsf{K} - x_i) & \sum_{\mathsf{K}\in\Omega_i} (x_\mathsf{K} - x_i)^2 & \sum_{\mathsf{K}\in\Omega_i} (x_\mathsf{K} - x_i)(y_\mathsf{K} - y_i) \\ \sum_{\mathsf{K}\in\Omega_i} (y_\mathsf{K} - y_i) & \sum_{\mathsf{K}\in\Omega_i} (x_\mathsf{K} - x_i)(y_\mathsf{K} - y_i) & \sum_{\mathsf{K}\in\Omega_i} (y_\mathsf{K} - y_i)^2 \end{pmatrix}, \\ \mathsf{B} = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ x_{\mathsf{K}_1} - x_i & x_{\mathsf{K}_2} - x_i & \ldots & x_{\mathsf{K}_{m_i}} - x_i \\ y_{\mathsf{K}_1} - y_i & y_{\mathsf{K}_2} - y_i & \ldots & y_{\mathsf{K}_{m_i}} - y_i \end{pmatrix},\end{gathered} \tag{2.19}$$

and the column vector $\mathbf{u} = (u_1, u_2, \ldots, u_{m_i})^T$, which collects the solution averages to be imposed. Using such definitions, the linear system (2.16)–(2.18) becomes

$$\mathsf{A}\begin{pmatrix}\alpha \\ \beta_x \\ \beta_y\end{pmatrix} = \mathsf{B}\mathbf{u}, \quad \text{which implies that } \begin{pmatrix}\alpha \\ \beta_x \\ \beta_y\end{pmatrix} = \mathsf{A}^{-1}\mathsf{B}\mathbf{u}, \tag{2.20}$$

since matrix A is, obviously, symmetric and positive definite and, thus, nonsingular. The coefficient α that provides the vertex value is given by

$$\alpha = (1,0,0)\begin{pmatrix}\alpha \\ \beta_x \\ \beta_y\end{pmatrix} = (1,0,0)\mathsf{A}^{-1}\mathsf{B}\mathbf{u}. \tag{2.21}$$

We require that such a coefficient be an average of the data in $\mathbf{u}$ with coefficients $\boldsymbol{\omega} = (\omega_{\mathsf{K}_1 i}, \omega_{\mathsf{K}_2 i}, \ldots, \omega_{\mathsf{K}_{m_i} i})^T$, that is, $\alpha = \boldsymbol{\omega}^T \mathbf{u}$. By comparison with (2.21), we have the final weight formula:

$$\boldsymbol{\omega} = \mathsf{B}^T \mathsf{A}^{-T} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}. \tag{2.22}$$

Conditions (2.12) can be checked by a direct calculations.

The functional in (2.14) can be suitably modified to take into account different kind of boundary conditions, for example, Neumann or Robin, compared to[41]. The values u_i at the vertexes $\mathbf{x}_i \in \Gamma$ on the Dirichlet boundary are constrained to the boundary data, for instance $u_i = 0$ for a homogeneous condition. We also mention that this choice of weights provides a very robust technique, compared to the study of locking effect due to the accuracy of the reconstruction [42].

3. Treatment of Discontinuous Diffusion Tensors

We discuss, here, how to treat the case of a diffusion tensor that is discontinuous across an interface σ shared by the cells K and L, that is, $\sigma = \mathsf{K} \mid \mathsf{L}$. Since we suppose that Λ is discontinuous across σ and the normal flux of the exact solution is continuous across σ, the normal component of the solution gradient must be discontinuous. Consequently, a numerical approximation of the diffusive flux across σ like

$$\int_{\sigma} \Lambda \nabla(u) \cdot \mathbf{n}\, dS \approx \Lambda_{\mathsf{K}\sigma} \nabla_{\sigma}(u_h) \cdot \mathbf{N}_{\mathsf{K}\sigma} \tag{3.1}$$

cannot be consistent whenever $\Lambda_{\mathsf{K}\sigma}$ is some kind of average between Λ_{K} and Λ_{L} and the constant vector $\nabla_{\sigma}(u_h)$ approximates $\nabla(u)$ *over all the diamond cell* hull $(\mathbf{x}_{\mathsf{K}}, \mathbf{x}_{\mathsf{L}}, \sigma)$. To tackle this problem, we consider two distinct approximations of the solution gradient within the subcells D_{K} and D_{L}, denoted by $\nabla_{\mathsf{K}\sigma}(u_h)$ and $\nabla_{\mathsf{L}\sigma}(u_h)$, respectively, and impose directly the flux conservation. To derive an expression for such *one-sided* gradients, we introduce an additional unknown u_σ along face σ and we apply the Gauss-Green Theorem as for the derivation of $\nabla_{\mathsf{KL}} u_h$ in the previous section. Also, we recall that $\mathbf{N}_{\mathsf{K}\sigma} = \mathbf{N}_{\mathsf{KL}}$ and $|\mathbf{N}_{\mathsf{K}\sigma}| = |\sigma|$ and we use a similar notation for the normal vector from the other side of σ, so that $\mathbf{N}_{\mathsf{L}\sigma} = -\mathbf{N}_{\mathsf{KL}}$ and $|\mathbf{N}_{\mathsf{L}\sigma}| = |\sigma|$. The two one-sided gradient formulas are

$$\begin{aligned} \nabla_{\mathsf{K}\sigma}(u_h) &:= \frac{1}{2|D_{\mathsf{K}}|}((u_\sigma - u_{\mathsf{K}})\mathbf{N}_{\mathsf{K}\sigma} + (u_{i+1} - u_i)\mathbf{N}_{ii+1}), \\ \nabla_{\mathsf{L}\sigma}(u_h) &:= \frac{1}{2|D_{\mathsf{L}}|}((u_\sigma - u_{\mathsf{L}})\mathbf{N}_{\mathsf{L}\sigma} - (u_{i+1} - u_i)\mathbf{N}_{ii+1}). \end{aligned} \tag{3.2}$$

Now, we search for a tensor $\Lambda_{\mathsf{K}\sigma}$, which is an average of the diffusion tensors Λ_{k} and Λ_{L}, and a gradient vector ∇_{KL}, which is an average of the gradient vectors $\nabla_{\mathsf{K}\sigma}(u_h)$ and

$\nabla_{\mathsf{L}\sigma}(u_h)$, that makes it possible to express directly the flux conservation as

$$\Lambda_{\mathsf{K}\sigma}\nabla_{\mathsf{KL}}(u_h)\cdot\mathbf{n}_\sigma := \Lambda_{\mathsf{K}}\nabla_{\mathsf{K}\sigma}(u_h)\cdot\mathbf{n}_\sigma = \Lambda_{\mathsf{L}}\nabla_{\mathsf{L}\sigma}(u_h)\cdot\mathbf{n}_\sigma. \tag{3.3}$$

Since we expect that the normal component of the vectors $\nabla_{\mathsf{K}\sigma}(u_h)$ and $\nabla_{\mathsf{L}\sigma}(u_h)$ be the same, we impose directly this condition by requiring that a scalar coefficient φ exists such that

$$\nabla_{\mathsf{L}\sigma}(u_h) = \nabla_{\mathsf{K}\sigma}(u_h) + \varphi\mathbf{n}_\sigma, \tag{3.4}$$

(recall that $\mathbf{n}_\sigma = \mathbf{n}_{\mathsf{KL}}$). We also assume that $\nabla_{\mathsf{KL}}(u_h)$ be a weighted average of $\nabla_{\mathsf{K}\sigma}\ (u_h)$ and $\nabla_{\mathsf{L}\sigma}\ (u_h)$. Let us be given two nonnegative coefficients μ_{K} and μ_{L} such that

$$\mu_{\mathsf{K}} + \mu_{\mathsf{L}} = 1, \tag{3.5}$$

and such that

$$\nabla_{\mathsf{KL}}(u_h) = \mu_{\mathsf{K}}\nabla_{\mathsf{K}\sigma}(u_h) + \mu_{\mathsf{L}}\nabla_{\mathsf{L}\sigma}(u_h). \tag{3.6}$$

To this purpose, a number of possible choices have been proposed and widely investigated in the literature for μ_{K} and μ_{L}. For example, a popular option is to take the volume of the two subdiamonds D_{K} and D_{L} normalized by the total volume of the diamond, for example, $\mu_{\mathsf{K}} = |D_{\mathsf{K}}|/|\mathsf{D}|$ and $\mu_{\mathsf{L}} = |D_{\mathsf{L}}|/|\mathsf{D}|$. The diffusion tensor can be chosen as the simple average of Λ_{K} and Λ_{L}, for example, $\Lambda_{\mathsf{KL}} = \mu_{\mathsf{K}}\Lambda_{\mathsf{K}} + \mu_{\mathsf{L}}\Lambda_{\mathsf{L}}$, or as the harmonic average, for example, $\Lambda_{\mathsf{KL}}^{-1} = \mu_{\mathsf{K}}\Lambda_{\mathsf{K}}^{-1} + \mu_{\mathsf{L}}\Lambda_{\mathsf{L}}^{-1}$. Much more attention must be paid when Λ is discontinuous across σ as none of the previous choices may provide the correct flux information across the common interface of cells K and L. Our aim in the next developments is at determining a value for φ and of the average diffusion tensor Λ_{KL} in terms of Λ_{K} and Λ_{L} that ensures flux conservation (3.3) for *any given pair* of coefficients μ_{K} and μ_{L}. Our derivation is similar to that considered in [43], and as pointed out therein, the resulting matrix is the same obtained for other purposes in the field of upscaling of conductivity, either by means of the large average techniques [44] or using the homogenization theory in the case of layered materials [45].

To this purpose, we first derive an expression for φ. We multiply (3.4) by $\Lambda_{\mathsf{L}}\mathbf{n}_\sigma$ and use flux conservation to obtain

$$\mathbf{n}_\sigma\cdot\Lambda_{\mathsf{L}}\nabla_{\mathsf{L}\sigma}(u_h) = \mathbf{n}_\sigma\cdot\left(\Lambda_{\mathsf{L}}\nabla_{\mathsf{K}\sigma}(u_h) + \varphi\Lambda_{\mathsf{L}}\mathbf{n}_\sigma\right) = \mathbf{n}_\sigma\cdot\Lambda_{\mathsf{K}}\nabla_{\mathsf{K}\sigma}(u_h), \tag{3.7}$$

from which we have

$$\mathbf{n}_\sigma\cdot(\Lambda_{\mathsf{K}} - \Lambda_{\mathsf{L}})\nabla_{\mathsf{K}\sigma}(u_h) = \varphi\mathbf{n}_\sigma\cdot\Lambda_{\mathsf{L}}\mathbf{n}_\sigma. \tag{3.8}$$

Likewise, we multiply (3.4) by $\Lambda_{\mathsf{K}}\mathbf{n}_\sigma$ and use flux conservation to obtain

$$\mathbf{n}_\sigma\cdot\Lambda_{\mathsf{K}}\nabla_{\mathsf{L}\sigma}(u_h) = \mathbf{n}_\sigma\cdot\left(\Lambda_{\mathsf{K}}\nabla_{\mathsf{K}\sigma}(u_h) + \varphi\Lambda_{\mathsf{K}}\mathbf{n}_\sigma\right) = \mathbf{n}_\sigma\cdot\left(\Lambda_{\mathsf{L}}\nabla_{\mathsf{L}\sigma}(u_h) + \varphi\Lambda_{\mathsf{K}}\mathbf{n}_\sigma\right), \tag{3.9}$$

from which we have

$$\mathbf{n}_\sigma\cdot(\Lambda_{\mathsf{K}} - \Lambda_{\mathsf{L}})\nabla_{\mathsf{L}\sigma}(u_h) = \varphi\mathbf{n}_\sigma\cdot\Lambda_{\mathsf{K}}\mathbf{n}_\sigma. \tag{3.10}$$

We multiply (3.8) by μ_{K} and (3.10) by μ_{L}, we sum the resulting relations and use (3.5) to have

$$\mathbf{n}_\sigma \cdot (\Lambda_{\mathsf{K}} - \Lambda_{\mathsf{L}})\big(\mu_{\mathsf{K}} \nabla_{\mathsf{K}\sigma}(u_h) + \mu_{\mathsf{L}} \nabla_{\mathsf{L}\sigma}(u_h)\big) = \varphi \mathbf{n}_\sigma \cdot \big(\mu_{\mathsf{K}} \Lambda_{\mathsf{L}} + \mu_{\mathsf{L}} \Lambda_{\mathsf{K}}\big)\mathbf{n}_\sigma. \tag{3.11}$$

Solving this last equation for φ gives us the formula:

$$\varphi = \frac{\mathbf{n}_\sigma \cdot (\Lambda_{\mathsf{K}} - \Lambda_{\mathsf{L}})\nabla_{\mathsf{KL}}(u_h)}{\mathbf{n}_\sigma \cdot \big(\mu_{\mathsf{K}} \Lambda_{\mathsf{L}} + \mu_{\mathsf{L}} \Lambda_{\mathsf{K}}\big)\mathbf{n}_\sigma}. \tag{3.12}$$

Now, we derive an expression for Λ_{KL}. In view of (3.5) and (3.3), it holds that

$$\mathbf{n}_\sigma \cdot \Lambda_{\mathsf{KL}} \nabla_{\mathsf{KL}}(u_h) = \mathbf{n}_\sigma \cdot \big(\mu_{\mathsf{K}} \Lambda_{\mathsf{K}} \nabla_{\mathsf{K}\sigma}(u_h) + \mu_{\mathsf{L}} \Lambda_{\mathsf{L}} \nabla_{\mathsf{L}\sigma}(u_h)\big). \tag{3.13}$$

First, in (3.13), we substitute the expression for $\nabla_{\mathsf{L}\sigma}(u_h)$ provided by (3.4), we collect the factor $\nabla_{\mathsf{K}\sigma}(u_h)$ and we obtain:

$$\mathbf{n}_\sigma \cdot \Lambda_{\mathsf{KL}} \nabla_{\mathsf{KL}}(u_h) = \mathbf{n}_\sigma \cdot \big(\mu_{\mathsf{K}} \Lambda_{\mathsf{K}} + \mu_{\mathsf{L}} \Lambda_{\mathsf{L}}\big) \nabla_{\mathsf{K}\sigma}(u_h) + \varphi \mu_{\mathsf{K}} \mathbf{n}_\sigma \cdot \Lambda_{\mathsf{L}} \mathbf{n}_\sigma. \tag{3.14}$$

Second, in (3.13), we substitute the expression for $\nabla_{\mathsf{K}\sigma}(u_h)$ provided by (3.4), we collect the factor $\nabla_{\mathsf{L}\sigma}(u_h)$ and we obtain

$$\mathbf{n}_\sigma \cdot \Lambda_{\mathsf{KL}} \nabla_{\mathsf{KL}}(u_h) = \mathbf{n}_\sigma \cdot \big(\mu_{\mathsf{K}} \Lambda_{\mathsf{K}} + \mu_{\mathsf{L}} \Lambda_{\mathsf{L}}\big) \nabla_{\mathsf{L}\sigma}(u_h) - \varphi \mu_{\mathsf{L}} \mathbf{n}_\sigma \cdot \Lambda_{\mathsf{K}} \mathbf{n}_\sigma. \tag{3.15}$$

Third, we multiply (3.14) by μ_{L}, (3.15) by μ_{K}, we add the two resulting equations, we use again (3.5) and (3.6), and we obtain

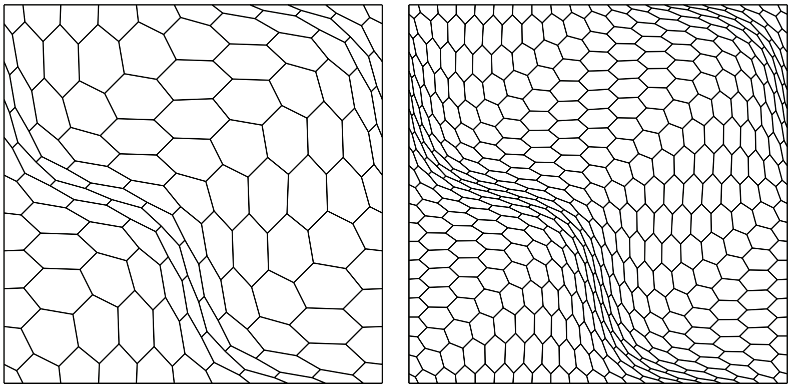
$$\mathbf{n}_\sigma \cdot \Lambda_{\mathsf{KL}} \nabla_{\mathsf{KL}}(u_h) = \mathbf{n}_\sigma \cdot \big(\mu_{\mathsf{K}} \Lambda_{\mathsf{K}} + \mu_{\mathsf{L}} \Lambda_{\mathsf{L}}\big) \nabla_{\mathsf{KL}}(u_h) + \varphi \mu_{\mathsf{K}} \mu_{\mathsf{L}} \mathbf{n}_\sigma \cdot (\Lambda_{\mathsf{L}} - \Lambda_{\mathsf{K}}) \mathbf{n}_\sigma. \tag{3.16}$$

Finally, in (3.16) we substitute the scalar factor φ given by (3.12), we collect the vectors $\mathbf{n}_F$ and $\nabla_{\mathsf{KL}}(u_h)$ and we obtain:

$$\mathbf{n}_\sigma \cdot \Lambda_{\mathsf{KL}} \nabla_{\mathsf{KL}}(u_h) = \mathbf{n}_\sigma \cdot \left(\mu_{\mathsf{K}} \Lambda_{\mathsf{K}} + \mu_{\mathsf{L}} \Lambda_{\mathsf{L}} + \mu_{\mathsf{K}} \mu_{\mathsf{L}} + \frac{\mathbf{n}_\sigma \cdot (\Lambda_{\mathsf{L}} - \Lambda_{\mathsf{K}}) \mathbf{n}_\sigma}{\mathbf{n}_\sigma \cdot \big(\mu_{\mathsf{K}} \Lambda_{\mathsf{K}} + \mu_{\mathsf{L}} \Lambda_{\mathsf{L}}\big) \mathbf{n}_\sigma} (\Lambda_{\mathsf{K}} - \Lambda_{\mathsf{L}}) \right) \nabla_{\mathsf{KL}}(u_h). \tag{3.17}$$

Equation (3.17) suggests us to set

$$\Lambda_{\mathsf{KL}} = \mu_{\mathsf{K}} \Lambda_{\mathsf{K}} + \mu_{\mathsf{L}} \Lambda_{\mathsf{L}} + \beta_{\mathsf{KL}} (\Lambda_{\mathsf{K}} - \Lambda_{\mathsf{L}}), \tag{3.18}$$



(a)

(b)

Figure 2: First and second mesh of mesh family $\mathcal{M}_1$. This mesh family is used in test case 1.

where

$$\beta_{\mathrm{KL}} = \frac{\mathbf{n}_\sigma \cdot (\Lambda_{\mathsf{L}} - \Lambda_{\mathsf{K}})\mathbf{n}_\sigma}{\mathbf{n}_\sigma \cdot (\Lambda_{\mathsf{K}}/\mu\mathsf{K} + \Lambda_{\mathsf{L}}/\mu\mathsf{L})\mathbf{n}_\sigma}. \tag{3.19}$$

4. Numerical Experiments

In this section, we present and discuss a set of numerical results to show the convergence behavior of the finite volume method when we compute the diffusive flux using the technique described in the previous sections. In all the test case that we present in this section, we compare the behavior of the new numerical treatment that we consider in this paper with the weighted average, which is the standard approach in the method [5, 41, 42]. In fact, from the formulation given in equation (3.18), it follows that the average diffusion tensor λ_{KL} is equal to the weighted average $\mu_{\mathsf{K}}\Lambda_{\mathsf{K}} + \mu_{\mathsf{L}}\Lambda_{\mathsf{L}}$ plus a correction term. This correction term is specifically designed to take care of possible discontinuities in the diffusion coefficients.

The finite volume formulation based on the least squares reconstruction of vertex values leads to a linear system for the cell-average unknowns whose coefficients matrix is generally unsymmetric although displaying a symmetric nonzero pattern. The positive definiteness of such system is still an open issue and this fact is also the major difficulty for the development of a full theory of convergence of such method. Theoretical results that prove coercivity are available only for meshes of slightly deformed quadrilaterals [2–4].

We solve such linear system using the MA41 routine of the HSL software collection, which implements an unsymmetric multifrontal sparse LU factorization technique especially designed for matrices with a symmetric nonzero pattern and unsymmetric values. Different software packages like UMFPACK and standard Krylov methods like BiCG-Stab and GMRES can be used in alternative. Efficiency issue is beyond the scope of our investigation but more details and comparison of performance when implementing different linear algebra solvers are found in the benchmark of the FVCA-6 Conference held in Prague, Czech Republic, in June 2011, [46].

We solve (2.1)-(2.2) on the domain $\Omega =]0,1[\times]0,1[$ for the data specified in the three test cases reported below. For all calculations, we measure the following relative errors:

Table 1: Data of mesh family $\mathcal{M}_1$; l is the mesh label, $\mathcal{N}_\mathsf{K}$ is the number of cells, $\mathcal{N}_\sigma$ is the number of mesh edges, $\mathcal{N}_\mathsf{V}$ is the number of mesh nodes, and h is the mesh size parameter.

l	$\mathcal{N}_\mathsf{K}$	$\mathcal{N}_\sigma$	$\mathcal{N}_\mathsf{V}$	h
1	121	400	280	$2.008\ 10^{-1}$
2	441	1400	960	$1.071\ 10^{-1}$
3	1681	5200	3520	$5.422\ 10^{-2}$
4	6561	20000	13440	$2.719\ 10^{-2}$
5	25921	78400	52480	$1.361\ 10^{-2}$

error on the solution:

$$\mathcal{E}_u = \left(\frac{\sum_{\mathsf{K}\in\Omega_h} |\mathsf{K}|\ |u(\mathbf{x}_\mathsf{K}) - u_\mathsf{K}|^2}{\sum_{\mathsf{K}\in\Omega_h} |\mathsf{K}|\ |u(\mathbf{x}_\mathsf{K})|^2} \right)^{1/2}, \tag{4.1}$$

error on the gradient:

$$\mathcal{E}_{\nabla u} = \left(\frac{\sum_{\sigma\in\Omega_h} |\sigma|\ |\nabla u(\mathbf{x}_\sigma) - \nabla_{\mathsf{KL}} u_h|^2}{\sum_{\mathsf{K}\in\Omega_h} |\sigma|\ |\nabla u(\mathbf{x}_\sigma)|^2} \right)^{1/2}. \tag{4.2}$$

Test Case 1

This test case is taken from [47] and is devoted to confirm that the new treatment of tensor coefficients and the standard weighted arithmetic average show the same behavior when the diffusion coefficients are regular, for example, constant or smoothly varying functions of position, even if small anisotropies along the principal direction of diffusion are present.The exact solution that we want to approximate is given by

$$u(x, y) = \sin(2\pi x)\sin(2\pi y) + x^3 + xy^2. \tag{4.3}$$

We consider two different constant diffusion tensor. The first one, called Λ_{iso}, is isotropic, while the second one, called Λ_{ani}, is anisotropic.The two diffusion tensors are given by:

$$\Lambda_{\text{iso}} = \begin{pmatrix} 1 & 0.25 \\ 0.25 & 1 \end{pmatrix}, \qquad \Lambda_{\text{ani}} = \begin{pmatrix} 1 & 0.1 \\ 0.1 & 0.25 \end{pmatrix}. \tag{4.4}$$

We solve this test case on mesh family $\mathcal{M}_1$. Each mesh of such family is built as follows. First, we remap the position $(\hat{x}, \hat{y})$ of the nodes of an $n \times n$ uniform partition by the smooth coordinate transformation:

$$\begin{aligned} x &= \hat{x} + \left(\frac{1}{10}\right)\sin(2\pi\hat{x})\sin(2\pi\hat{y}), \\ y &= \hat{y} + \left(\frac{1}{10}\right)\sin(2\pi\hat{x})\sin(2\pi\hat{y}). \end{aligned} \tag{4.5}$$

The corresponding mesh of $\mathcal{M}_1$ is built from this "primal" mesh by splitting each quadrilateral cell into two triangles and connecting the barycenters of adjacent triangular cells by a straight segment. The mesh construction is completed at the boundary Γ by connecting the barycenters of the triangular cells close to Γ to the midpoints of the boundary edges and these latters to the boundary vertices of the "primal" mesh. The first and the second mesh of mesh family $\mathcal{M}_1$ are displayed in Figure 2; mesh data are given in Table 1. Approximation errors for the isotropic diffusion tensor Λ_{iso} are shown in Figure 6; approximation errors for the anisotropic tensor Λ_{ani} are shown in Figure 7. In both plots, we also show the exact slopes proportional to h and h^2. From Figures 6 and 7, we deduce that in the case of constant diffusion tensors the performance of the new diffusion average proposed in this work and the weighted arithmetic average are almost the same. This behavior is confirmed both in the case of isotropic diffusion tensor and in the case of an anisotropic diffusion tensor.

Test Case 2

In this test case, we show the behavior of the new technique that is proposed in this paper when the diffusion coefficients are discontinuous along an internal line of the computational domain. To such purpose, we split $\Omega = \cup_{i=1}^{2}\Omega_i$ where

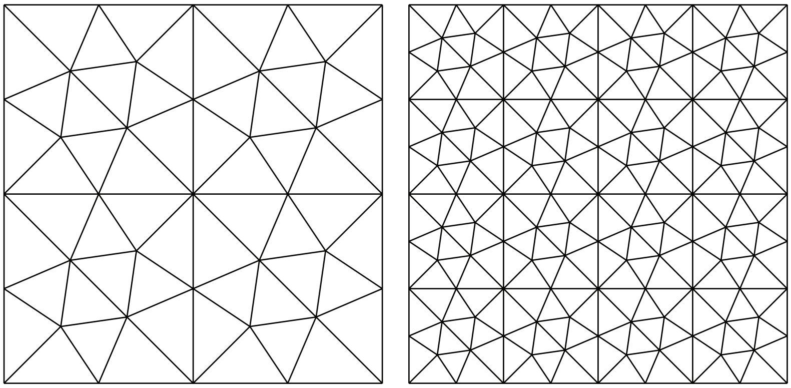
$$\begin{aligned} \Omega_1 &= \left\{(x,y) \in \Omega : 0 \le x \le 1,\ 0 \le y \le \frac{1}{2}\right\}, \\ \Omega_2 &= \left\{(x,y) \in \Omega : 0 \le x \le 1,\ \frac{1}{2} < y \le 1\right\}. \end{aligned} \tag{4.6}$$

The diffusion tensor is discontinuous across the horizontal line $y = 1/2$ and is given by

$$\Lambda(x,y) = \begin{cases} \lambda_1\widetilde{\Lambda} \text{ for any } x \in [0,1],\ y \in \left[0, \dfrac{1}{2}\right] \\ \lambda_2\widetilde{\Lambda} \text{ for any } x \in [0,1],\ y \in \left[\dfrac{1}{2}, 1\right], \end{cases} \tag{4.7}$$

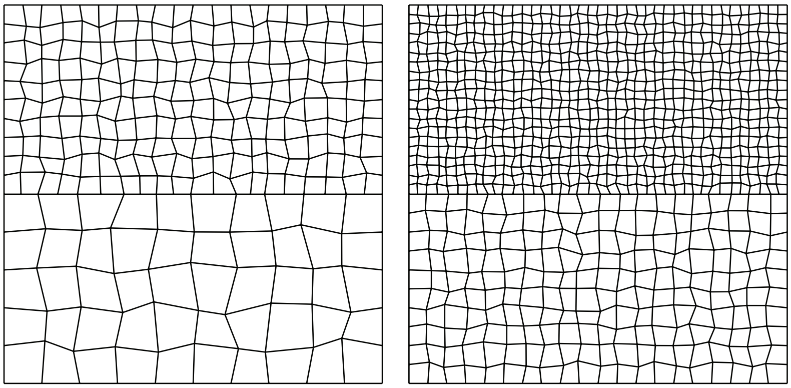
where

$$\widetilde{\Lambda} = \begin{pmatrix} 1 + x^2 + y^2 & -\left(x - \dfrac{1}{2}\right)\left(y - \dfrac{1}{2}\right) \\ -\left(x - \dfrac{1}{2}\right)\left(y - \dfrac{1}{2}\right) & 1 + x^2 + y^2 \end{pmatrix}, \tag{4.8}$$



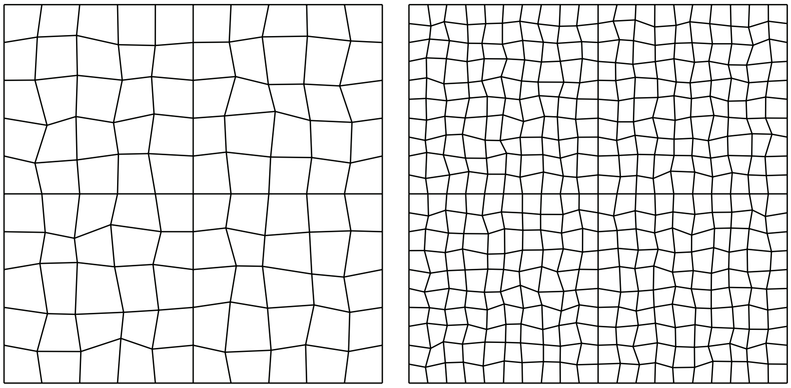
(a) (b)

Figure 3: First and second mesh of mesh family $\mathscr{M}_2$. This mesh family is used in test case 2.



(a) (b)

Figure 4: First and second mesh of mesh family $\mathscr{M}_3$. This mesh family is used in test case 2.



(a) (b)

Figure 5: First and second mesh of mesh family $\mathscr{M}_4$. This mesh family is used in test case 3.

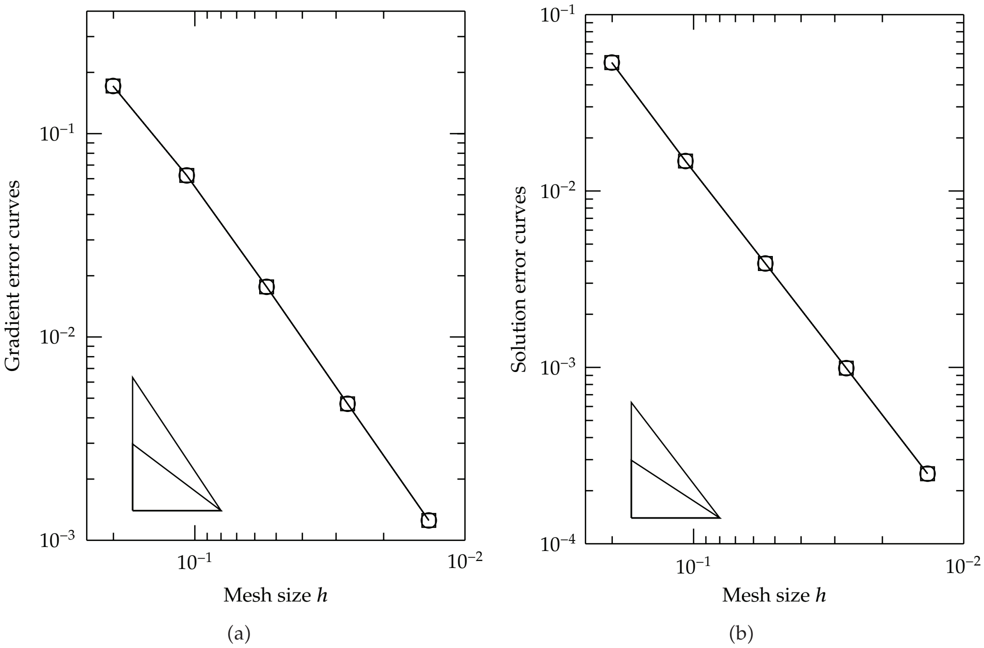


Figure 6: Test case 1: error curves for the gradient approximation (left plot) and the solution approximation (right plot). Problem (2.1)-(2.2) is solved using mesh family $\mathcal{M}_1$ and the isotropic diffusion tensor Λ_{iso} in (4.4).The errors obtained by average (3.18)-(3.19) are marked by circles, the errors obtained by using the weighted arithmetic average are marked by squares. Note that the two error curves are superimposed. In the bottom-left corners of both plots, we show the exact slopes proportional to h and h^2.

and $\lambda_1 = 1$, $\lambda_2 = 10$. The exact solution is continuous across the line $y = 1/2$ and is designed to ensure flux conservation, that is, continuity of the normal component of the flux field. The exact solution is given by

$$u(x,y) = \begin{cases} 1 + x + 3\,y^3 & \text{for any } x \in [0,1], \quad y \in \left[0, \dfrac{1}{2}\right] \\ 3\,\dfrac{\lambda_2 - \lambda_1}{2^3\,\lambda_2} + x + 3\,\dfrac{\lambda_1}{\lambda_2}\,y^3 & \text{for any } x \in [0,1], \quad y \in \left[\dfrac{1}{2}, 1\right]. \end{cases} \tag{4.9}$$

This test case is solved on the two mesh families described below.

Mesh Family $\mathcal{M}_2$

This mesh sequence is the first of the mesh collection of the two-dimensional benchmark of the conference “Finite Volumes for Complex Applications—V” held in Aussois (France) in 2008. The first mesh and the first refined mesh of this mesh suite are shown in Figure 3; mesh data are reported in Table 2.

Mesh Family $\mathcal{M}_3$

Below $y = 1/2$, we consider a regular mesh formed by $2n \times n$ squares, while above $y = 1/2$ we consider a regular mesh formed by $4n \times 2n$ squares. All the mesh nodes except those located at the boundaries and those located at the internal discontinuity line $y = 1/2$ are then displaced by perturbing their position to a random position inside a square box centered at that original node position. The sides of this square box are aligned with the coordinate axis and their length is equal to $0.8/n$ (note that $1/n$ is the distance between two consecutive nodes in each direction). Instead, the nodes lying at $y = 1/2$ are allowed to change only the abscissa in order not to modify the shape of the interface line while the nodes at the boundary are not displaced. We treat a nonmatching mesh as a conformal polygonal mesh modifying the shape of the polygons that are immediately below the line $y = 1/2$: these polygons are treated as degenerate pentagons with two parallel consecutive edges. The first mesh is built by taking $n = 10$ and mesh refinement is implemented by doubling the value of n at each refinement step, thus implying that the mesh size parameter is approximately halved when passing from one mesh to the next one in the refinement process.The first mesh and the first refined mesh of this mesh suite are shown in Figure 4; mesh data are reported in Table 3.

The numerical results for the gradient and the solution approximation on mesh family $\mathcal{M}_2$ are shown in the two plots of Figure 8 and on mesh family $\mathcal{M}_3$ in the two plots of Figure 9. In the bottom-left corner of both plots, we show the exact slopes proportional to $h^{1/2}$ (gradient errors) and to h and h^2 (solution errors). The solution gradient has a discontinuous normal component across such lines, and, due to this lack of regularity, the convergence rate cannot be expected to be better than that of a first-order scheme. This behavior is reflected in both left plots of Figures 8 and 9. Even if the convergence rate of the gradient is the same in the two cases, the approximation errors of the formulation using the new average are smaller than those obtained when using the standard weighted arithmetic average. Concerning the solution approximation, the results are even more spectacular because adopting the new average allows us to recover the second-order convergence rate, thus confirming the superior behavior of the new method.

Test Case 3

In this test case we aim at confirming the behavior of test case 2 for a more difficult case in which the diffusion tensor has intersecting discontinuities with an internal cross point.To such purpose, we split the computational domain as $\Omega = \cup_{i=1}^{4}\Omega_i$ where

$$\begin{aligned}
\Omega_1 &= \left\{ (x,y) \in \Omega : 0 \le x \le \frac{1}{2},\ 0 \le y \le \frac{1}{2} \right\},\\
\Omega_2 &= \left\{ (x,y) \in \Omega : \frac{1}{2} \le x \le 1,\ 0 \le y \le \frac{1}{2} \right\},\\
\Omega_3 &= \left\{ (x,y) \in \Omega : \frac{1}{2} \le x \le 1,\ \frac{1}{2} < y \le 1 \right\},\\
\Omega_4 &= \left\{ (x,y) \in \Omega : 0 \le x \le \frac{1}{2},\ \frac{1}{2} < y \le 1 \right\}.
\end{aligned} \tag{4.10}$$

Table 2: Data of mesh family $\mathcal{M}_2$; l is the mesh label, $\mathcal{N}_{\mathrm{K}}$ is the number of cells, $\mathcal{N}_\sigma$ is the number of mesh edges, $\mathcal{N}_{\mathrm{V}}$ is the number of mesh nodes, and h is the mesh size parameter.

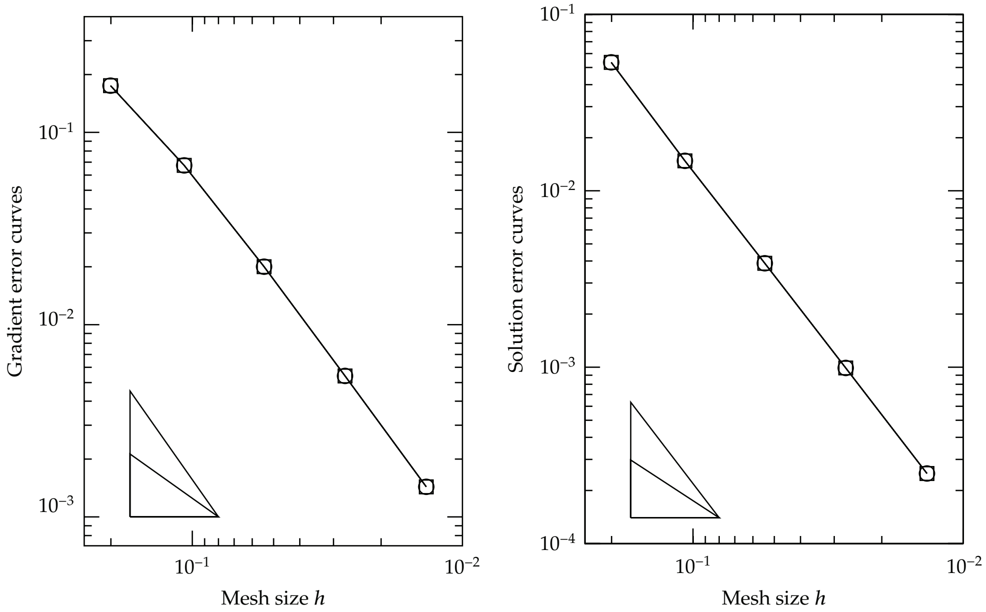
l	$\mathcal{N}_{\mathrm{K}}$	$\mathcal{N}_\sigma$	$\mathcal{N}_{\mathrm{V}}$	h
1	250	535	286	$1.865\ 10^{-1}$
2	1000	2070	1071	$9.051\ 10^{-2}$
3	4000	8140	4141	$4.693\ 10^{-2}$
4	16000	32280	16281	$2.407\ 10^{-2}$
5	64000	128560	64561	$1.215\ 10^{-2}$

Table 3: Data of mesh family $\mathcal{M}_3$; l is the mesh label, $\mathcal{N}_{\mathrm{K}}$ is the number of cells, $\mathcal{N}_\sigma$ is the number of mesh edges, $\mathcal{N}_{\mathrm{V}}$ is the number of mesh nodes, and h is the mesh size parameter.

l	$\mathcal{N}_{\mathrm{K}}$	$\mathcal{N}_\sigma$	$\mathcal{N}_{\mathrm{V}}$	h
1	56	92	37	$2.500\ 10^{-1}$
2	224	352	129	$1.250\ 10^{-1}$
3	896	1376	481	$6.250\ 10^{-2}$
4	3584	5440	1857	$3.125\ 10^{-2}$
5	14336	21632	7297	$1.563\ 10^{-2}$
6	57344	86272	28929	$7.813\ 10^{-3}$
7	229376	344576	115201	$3.906\ 10^{-3}$

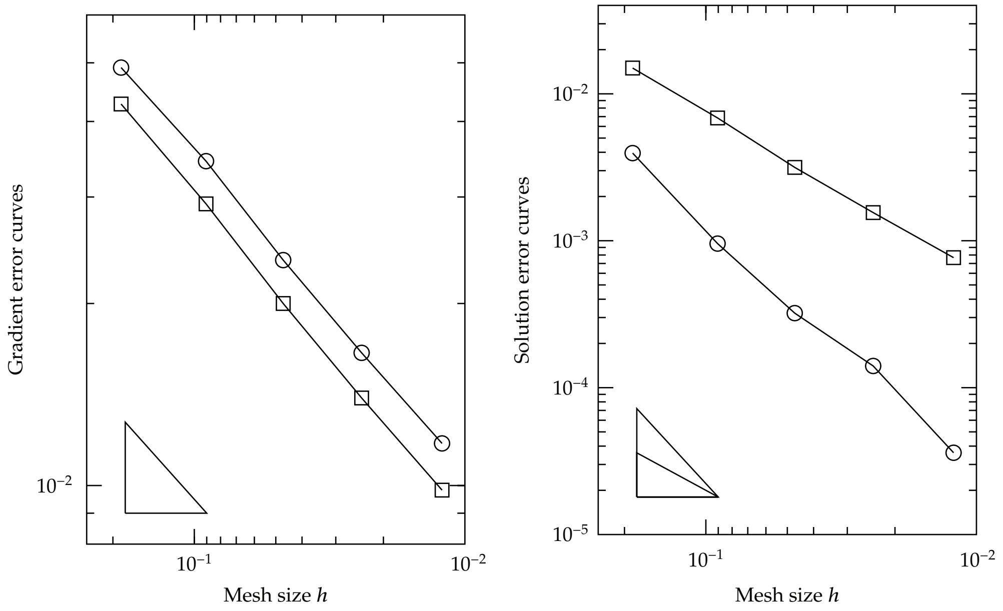
Table 4: Data of mesh family $\mathcal{M}_4$; l is the mesh label, $\mathcal{N}_{\mathrm{K}}$ is the number of cells, $\mathcal{N}_\sigma$ is the number of mesh edges, $\mathcal{N}_{\mathrm{V}}$ is the number of mesh nodes, and h is the mesh size parameter.

l	$\mathcal{N}_{\mathrm{K}}$	$\mathcal{N}_\sigma$	$\mathcal{N}_{\mathrm{V}}$	h
1	100	220	121	$1.796\ 10^{-1}$
2	400	840	441	$9.150\ 10^{-2}$
3	1600	3280	1681	$4.788\ 10^{-2}$
4	6400	12960	6561	$2.423\ 10^{-2}$
5	25600	51520	25921	$1.216\ 10^{-2}$



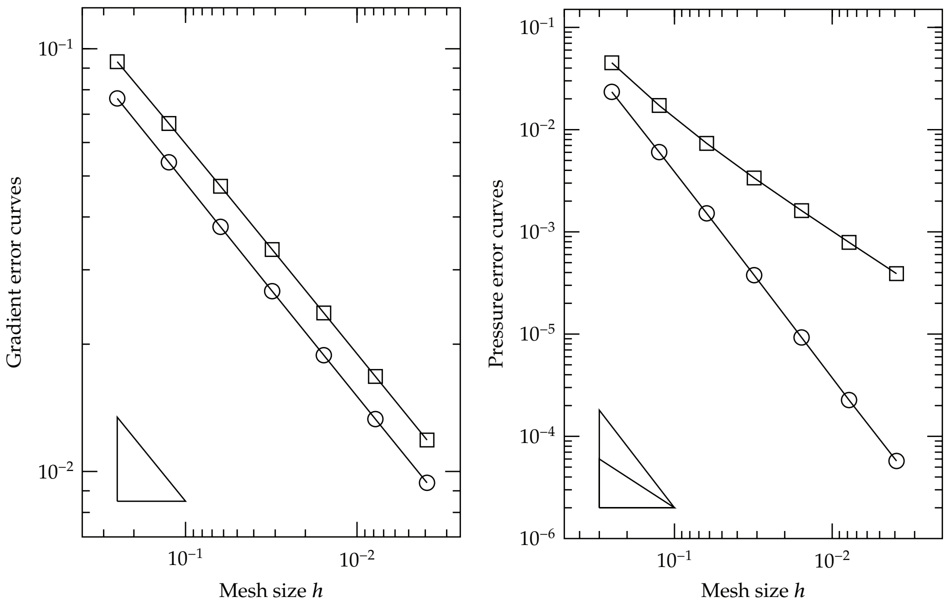
(a) (b)

Figure 7: Test case 1: error curves for the gradient approximation (left plot) and the solution approximation (right plot). We use the anisotropic diffusion tensor Λ_{ani} of (4.4). The errors obtained by average (3.18)-(3.19) are marked by circles, the errors obtained by using the weighted arithmetic average are marked by squares. Note that the two error curves are superimposed. In the bottom-left corners of both plots, we show the exact slopes proportional to h and h^2.



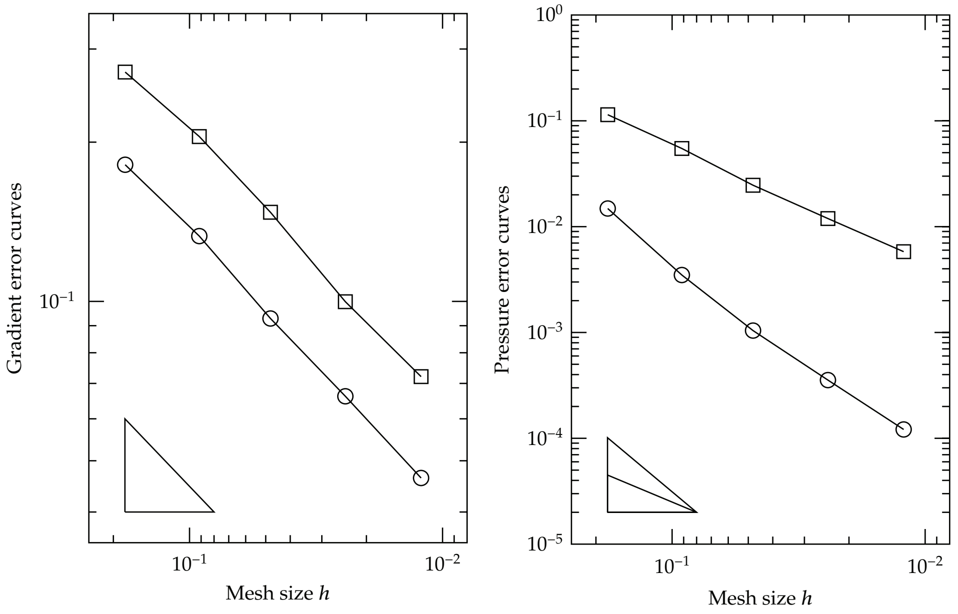
(a) (b)

Figure 8: Test case 2: error curves for the gradient approximation (left plot) and the solution approximation (right plot). Problem (2.1)-(2.2) is solved using mesh family $\mathcal{M}_2$ and the discontinuous diffusion tensor (4.7).The errors obtained by average (3.18)-(3.19) are marked by circles, the errors obtained by using the weighted arithmetic average are marked by squares. In the bottom-left corner of both plots, we show the exact slopes proportional to $h^{1/2}$ (gradient errors) and to h and h^2 (solution errors).



(a) (b)

Figure 9: Test case 2: error curves for the gradient approximation (left plot) and the solution approximation (right plot). Problem (2.1)–(2.2) is solved using mesh family $\mathscr{M}_3$ and the discontinuous diffusion tensor (4.7). The errors obtained by average (3.18)-(3.19) are marked by circles, the errors obtained by using the weighted arithmetic average are marked by squares. In the bottom-left corner of both plots we show the exact slopes proportional to $h^{1/2}$ (gradient errors) and to h and h^2 (solution errors).



(a) (b)

Figure 10: Test case 3: error curves for the gradient approximation (left plot) and the solution approximation (right plot). Problem (2.1)-(2.2) is solved using mesh family $\mathscr{M}_4$ and the discontinuous diffusion tensor (4.11). The errors obtained by average (3.18)-(3.19) are marked by circles, the errors obtained by using the weighted arithmetic average are marked by squares. In the bottom-left corner of both plots we show the exact slopes proportional to $h^{1/2}$ (gradient errors) and to h and h^2 (solution errors).

The diffusion tensor is discontinuous across the horizontal line $y = 1/2$ and the vertical line $x = 1/2$ and is given by

$$\Lambda_i = \frac{1}{\alpha_i}\begin{pmatrix} 1+\left(x-\frac{1}{2}\right)\left(y-\frac{1}{2}\right) & -(x-1/2)\left(y-\frac{1}{2}\right) \\ -\left(x-\frac{1}{2}\right)\left(y-\frac{1}{2}\right) & 1+\left(x-\frac{1}{2}\right)\left(y-\frac{1}{2}\right) \text{ for } (x,y)\in\Omega_i,\ i=1,\dots,4, \end{pmatrix}, \tag{4.11}$$

where $\alpha_1 = \alpha_3 = 1$ and $\alpha_2 = \alpha_4 = 100$. The exact solution is continuous across such line and is designed to ensure flux conservation, that is, continuity of the normal component of the flux field. The exact solution is given by

$$u(x,y) = 2\alpha_i(1-x)\left(\frac{1}{2}-x\right)\cos(2\pi y) \quad \text{for } (x,y)\in\Omega_i, \quad i=1,\dots,4. \tag{4.12}$$

This test case is solved on the mesh family $\mathcal{M}_4$ 5 that is built by displacing randomly all the nodes except those at the subdomain interfaces $x = 1/2$ and $y = 1/2$ of a $n \times n$ regular partition of Ω. We consider five meshes with $n = 10, 20, 40, 80, 160$; mesh data are shown in Table 4, and the first mesh and the first refined mesh of this mesh suite are shown in Figure 5.

The numerical results for the gradient and the solution approximation on mesh family $\mathcal{M}_4$ are shown in the two plots of Figure 10. In the bottom-left corner of these plots, we show the exact slopes proportional to $h^{1/2}$ (gradient errors) and to h and h^2 (solution errors). As in test case 2, the normal component of the solution gradient is discontinuous across the internal lines $x = 1/2$ and $y = 1/2$. Therefore, the convergence rate that is numerically measured in this test case is proportional to $h^{1/2}$, cf. the left plot of Figure 10. This fact is independent of the average technique that is used for the diffusion coefficients. Nonetheless, the approximation errors given by the new average are smaller than those obtained when using the standard weighted arithmetic average. Concerning the solution approximation, such results confirms the behavior seen in test case 2. Consequently, the finite volume approximation of the scalar solution recovers the optimal second-order convergence rate when we apply the new average technique.

5. Conclusions

The numerical treatment of diffusion coefficients is an open problem and major problem, as, for example, the conductivity of different layers of soil is abruptly spatially variable. Moreover, it is particularly important in view of the problems of different scales involved and of random fields of the coefficients themselves. To this purpose, we proposed and tested a new technique for the numerical treatment of the conductivity tensor that interpolates the information coming from the different sides of a mesh interface shared by two adjacent cells. We point it out that this design is particularly suitable to the case of discontinuous conductivity tensors since the interpolation automatically adjusts its value by introducing an appropriate correction to the standard weighted average. Numerical experiments confirm this behavior. Furthermore, it may result in a particularly efficient strategy for the numerical resolution of problems where both the diffusive/dispersive and convective phenomena are simultaneously significant.

Acknowledgments

This work was funded by the PRIN 2007 project “Experimental measurement of the soil-vegetation-atmosphere interaction processes and numerical modelling of their response to climate change”. The authors also thank Dr. G. Manzini for the meshes used in test case 1 and test case 3.

References

[1] R. Eymard, T. Gallouët, and R. Herbin, “Finite volume methods,” in *Handbook of Numerical Analysis*, vol. 7, pp. 713–1020, North-Holland, Amsterdam, The Netherlands, 2000.

[2] Y. Coudière, *Analyse de schémas volumes finis sur maillages non structurés pour des problèmes linéaires hyperboliques et elliptiques*, Ph.D. thesis, Universitè “P. Sabatier” de Toulouse, Toulouse, France, 1999.

[3] Y. Coudière, J.-P. Vila, and P. Villedieu, “Convergence rate of a finite volume scheme for a two-dimensional convection-diffusion problem,” *Mathematical Modelling and Numerical Analysis*, vol. 33, no. 3, pp. 493–516, 1999.

[4] Y. Coudière and P. Villedieu, “Convergence rate of a finite volume scheme for the linear convection-diffusion equation on locally refined meshes,” *M2AN. Mathematical Modelling and Numerical Analysis*, vol. 34, no. 6, pp. 1123–1149, 2000.

[5] E. Bertolazzi and G. Manzini, “A cell-centered second-order accurate finite volume method for convection-diffusion problems on unstructured meshes,” *Mathematical Models & Methods in Applied Sciences*, vol. 14, no. 8, pp. 1235–1260, 2004.

[6] G. Manzini and A. Russo, “A finite volume method for advection-diffusion problems in convection-dominated regimes,” *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 13–16, pp. 1242–1261, 2008.

[7] G. Manzini and S. Ferraris, “Mass-conservative finite volume methods on 2-D unstructured grids for the Richards’ equation,” *Advances in Water Resources*, vol. 27, no. 12, pp. 1199–1215, 2004.

[8] R. Haverkamp and M. Vauclin, “A note on estimating finite difference interblock hydraulic conductivity values for transient unsaturated flow problems,” *Water Resources Research*, vol. 15, no. 1, pp. 181–188, 1979.

[9] N. Varado, I. Braud, P. J. Ross, and R. Haverkamp, “Assessment of an efficient numerical solution of the 1D Richards’ equation on bare soil,” *Journal of Hydrology*, vol. 323, no. 1–4, pp. 244–257, 2006.

[10] N. Romano, B. Brunone, and A. Santini, “Numerical analysis of one-dimensional unsaturated flow in layered soils,” *Advances in Water Resources*, vol. 21, no. 4, pp. 315–324, 1998.

[11] J. M. Gastó, J. Grifoll, and Y. Cohen, “Estimation of internodal permeabilities for numerical simulation of unsaturated flows,” *Water Resources Research*, vol. 38, no. 12, pp. 621–6210, 2002.

[12] A. Szymkiewicz and R. Helmig, “Comparison of conductivity averaging methods for one-dimensional unsaturated flow in layered soils,” *Advances in Water Resources*, vol. 34, no. 8, pp. 1012–1025, 2011.

[13] P. Berg, “Long-term simulation of water movement in soils using mass-conserving procedures,” *Advances in Water Resources*, vol. 22, no. 5, pp. 419–430, 1999.

[14] F. Chen and L. Ren, “Application of the finite difference heterogeneous multiscale method to the Richards’ equation,” *Water Resources Research*, vol. 44, no. 7, article W07413, 2008.

[15] L. Li, H. Zhou, and J. J. Gómez-Hernández J. Jaime, “A comparative study of three-dimensional hydraulic conductivity upscaling at the macro-dispersion experiment (MADE) site, Columbus Air Force Base, Mississippi (USA),” *Journal of Hydrology*, vol. 404, no. 3-4, pp. 278–293, 2011.

[16] W. E and B. Engquist, “The heterogeneous multiscale methods,” *Communications in Mathematical Sciences*, vol. 1, no. 1, pp. 87–132, 2003.

[17] Y. Efendiev and T. Y. Hou, *Multiscale Finite Element Methods. Theory and Applications*, vol. 4, Springer, New York, NY, USA, 2009.

[18] I. Babuška, "Solution of interface problems by homogenization—I, II," *SIAM Journal on Mathematical Analysis*, vol. 7, no. 5, pp. 603–634, 1976.

[19] I. Babuška, "Homogenization and its application. Mathematical and computational problems," in *Numerical Solution of Partial Differential Equations III*, B. Hubbard, Ed., pp. 89–116, Academic Press, New York, NY, USA, 1976.

[20] I. Babuška, "Solution of interface problems by homogenization—III," *SIAM Journal on Mathematical Analysis*, vol. 8, no. 6, pp. 923–937, 1977.

[21] L. J. Durlofsky, Y. Efendiev, and V. Ginting, "An adaptive localglobal multiscale finite volume element method for two-phase flow simulations," *Advances in Water Resources*, vol. 30, pp. 576–588, 2007.

[22] P. Dostert, Y. Efendiev, and B. Mohanty, "Efficient uncertainty quantification techniques for Richards' equation," *Advances in Water Resources*, vol. 32, pp. 329–339, 2009.

[23] L. Bergamaschi, S. Mantica, and G. Manzini, "A mixed finite element—finite volume formulation of the black-oil model," *SIAM Journal on Scientific Computing*, vol. 20, no. 3, pp. 970–997, 1999.

[24] C. Gallo and G. Manzini, "2-D numerical modeling of bioremediation in heterogeneous saturated soils," *Transport in Porous Media*, vol. 31, no. 1, pp. 67–88, 1998.

[25] C. Gallo and G. Manzini, "A mixed finite element finite volume approach for solving biodegradation transport in groundwater," *International Journal for Numerical Methods in Fluids*, vol. 26, no. 5, pp. 533–556, 1998.

[26] C. Gallo and G. Manzini, "A fully coupled numerical model for two-phase flow with contaminant transport biodegradation kinetics," *Communications in Numerical Methods in Engineering*, vol. 17, no. 5, pp. 325–336, 2001.

[27] L. Beirão da Veiga, V. Gyrya, K. Lipnikov, and G. Manzini, "Mimetic finite difference method for the Stokes problem on polygonal meshes," *Journal of Computational Physics*, vol. 228, no. 19, pp. 7215–7232, 2009.

[28] A. Bouziani and I. Mounir, "On the solvability of a class of reaction-diffusion systems," *Mathematical Problems in Engineering*, vol. 2006, Article ID 47061, 15 pages, 2006.

[29] A. Cangiani and G. Manzini, "Flux reconstruction and solution post-processing in mimetic finite difference methods," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 9–12, pp. 933–945, 2008.

[30] W. Correa de Oliveira Casaca and M. Boaventura, "A decomposition and noise removal method combining diffusion equation and wave atoms for textured images," *Mathematical Problems in Engineering*, vol. 2010, Article ID 764639, 20 pages, 2010.

[31] M. Dehghan, "On the numerical solution of the diffusion equation with a nonlocal boundary condition," *Mathematical Problems in Engineering*, vol. 2003, no. 2, pp. 81–92, 2003.

[32] S. Hu, Z. Liao, D. Sun, and W. Chen, "A numerical method for preserving curve edges in nonlinear anisotropic smoothing," *Mathematical Problems in Engineering*, vol. 2011, Article ID 186507, 14 pages, 2011.

[33] Z. Liao, S. Hu, D. Sun, and W. Chen, "Enclosed Laplacian operator of nonlinear anisotropic diffusion to preserve singularities and delete isolated points in image smoothing," *Mathematical Problems in Engineering*, vol. 2011, Article ID 749456, 15 pages, 2011.

[34] R. J. Moitsheki, "Transient heat diffusion with temperature-dependent conductivity and time-dependent heat transfer coefficient," *Mathematical Problems in Engineering*, vol. 2008, Article ID 347568, 9 pages, 2008.

[35] N. P. Moshkin and D. Yambangwai, "On numerical solution of the incompressible Navier-Stokes equations with static or total pressure specified on boundaries," *Mathematical Problems in Engineering*, vol. 2009, Article ID 372703, 26 pages, 2009.

[36] W. Wenquan, Z. Lixiang, Y. Yan, and G. Yakun, "Finite element analysis of turbulent flows using les and dynamic subgridscale models in complex geometries," *Mathematical Problems in Engineering*, vol. 2011, Article ID 712372, 20 pages, 2011.

[37] J. Yu, Y. Yang, and A. Campo, "Approximate solution of the nonlinear heat conduction equation in a semi-infinite domain," *Mathematical Problems in Engineering*, vol. 2010, Article ID 421657, 24 pages, 2010.

[38] Y. Zheng and Z. Zhao, "A fully discrete Galerkin method for a nonlinear space-fractional diffusion equation," *Mathematical Problems in Engineering*, vol. 2011, Article ID 171620, 20 pages, 2011.

[39] P. Grisvard, *Elliptic Problems in Nonsmooth Domains*, vol. 24, Pitman, Boston, Mass, USA, 1985.

[40] E. Bertolazzi and G. Manzini, "A second-order maximum principle preserving finite volume method for steady convection-diffusion problems," *SIAM Journal on Numerical Analysis*, vol. 43, no. 5, pp. 2172–2199, 2005.

[41] E. Bertolazzi and G. Manzini, "On vertex reconstructions for cell-centered finite volume approximations of 2D anisotropic diffusion problems," *Mathematical Models & Methods in Applied Sciences*, vol. 17, no. 1, pp. 1–32, 2007.

[42] G. Manzini and M. Putti, "Mesh locking effects in the finite volume solution of 2-D anisotropic diffusion equations," *Journal of Computational Physics*, vol. 220, no. 2, pp. 751–771, 2007.

[43] G. P. Leonardi, F. Paronetto, and M. Putti, "Effective anisotropy tensor for the numerical solution of flow problems in heterogeneous porous media," in *Proceedings of the CMWR 16th International Conference*, Copenhagen, Denmark, 2006.

[44] M. Quintard and S. Whitaker, "Single-phase flow in porous media: effect of local heterogeneities," *Journal de Mecanique Theorique et Appliquee*, vol. 6, no. 5, pp. 691–726, 1987.

[45] L. Tartar, "Estimations fines des coefficients homogénéisés," in *Ennio De Giorgi Colloquium*, vol. 125, pp. 168–187, Pitman, Boston, Mass, USA, 1985.

[46] R. Eymard, G. Henri, R. Herbin, R. Klofkorn, and G. Manzini, "3D Benchmark on discretizations schemes for anisotropic diffusion problems on general grids," in *Finite Volumes for Complex Applications VI, Problems and Perspectives*, J. Fort, J. Furst, J. Halama, R. Herbin, and F. Hubert, Eds., vol. 2, pp. 95–130, Springer, 2011.

[47] L. Beirão da Veiga and G. Manzini, "A higher-order formulation of the mimetic finite difference method," *SIAM Journal on Scientific Computing*, vol. 31, no. 1, pp. 732–760, 2008.