Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 220 (2007) 751–771

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Mesh locking effects in the finite volume solution of 2-D anisotropic diffusion equations

Gianmarco Manzini [a], Mario Putti [b,*]

[a] *Istituto di Matematica Applicata e Tecnologie Informatiche, C.N.R., via Ferrata 1, I – 27100 Pavia, Italy*
[b] *Dipartimento di Metodi e Modelli Matematici per le Scienze Applicate, Università degli Studi di Padova, via Belzoni 7, I – 35131 Padova, Italy*

Received 9 January 2006; received in revised form 9 May 2006; accepted 23 May 2006
Available online 14 July 2006

Abstract

Strongly anisotropic diffusion equations require special techniques to overcome or reduce the mesh locking phenomenon. We present a finite volume scheme that tries to approximate with the best possible accuracy the quantities that are of importance in discretizing anisotropic fluxes. In particular, we discuss the crucial role of accurate evaluations of the tangential components of the gradient acting tangentially to the control volume boundaries, that are called into play by anisotropic diffusion tensors. To obtain the sought characteristics from the proposed finite volume method, we employ a second-order accurate reconstruction scheme which is used to evaluate both normal and tangential cell-interface gradients. The experimental results on a number of different meshes show that the scheme maintains optimal convergence rates in both L^2 and H^1 norms except for the benchmark test considering full Neumann boundary conditions on non-uniform grids. In such a case, a severe locking effect is experienced and documented. However, within the range of practical values of the anisotropy ratio, the scheme is robust and efficient. We postulate and verify experimentally the existence of a quadratic relationship between the anisotropy ratio and the mesh size parameter that guarantees optimal and sub-optimal convergence rates.
© 2006 Elsevier Inc. All rights reserved.

Keywords: Mesh locking; Finite volume; Poisson problem; Unstructured grid; Partial differential equation; Numerical solver; Anisotropy

1. Introduction

Diffusion equations with anisotropic (i.e. direction dependent) coefficients arise in many practical applications such as, for example, heat transfer, groundwater flow and contaminant transport, petroleum reservoir simulations, Navier–Stokes equations, and so on. These problems are characterized by a diffusion coefficient represented by a space dependent full rank tensor, which becomes diagonal if the reference system is aligned with the principal directions of anisotropy [7]. This class of problems is also known as parameter dependent

[*] Corresponding author. Tel.: +39 0498275919; fax: +39 0498275995.
E-mail address: putti@dmsa.unipd.it (M. Putti).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.05.026

problems, the parameter in this case being the anisotropy ratio, i.e. the ratio between the smallest and largest eigenvalues of the diffusion tensor [5]. Efficient numerical discretization of strongly anisotropic problems is generally obtained by means of ad hoc, mesh dependent scheme modifications developed to overcome the problem known as parametric locking. Locking is experimentally observed when the discretization error does not decrease at the expected rate for limiting values of the parameter. This loss of convergence disappears for sufficiently fine discretizations, but may involve costly or even unfeasibly large calculations [5]. The numerical solution of this type of problems requires careful consideration of the errors that may be introduced by the discretization scheme.

Several authors have devised various techniques with the aim of alleviating the locking problem. A number of these studies concentrate on the development of schemes capable of handling anisotropic diffusion coefficients, without however specific investigation of the locking phenomenon [18,25–28]. Other authors are mainly concerned with studies of scheme performance in the presence of locking. Within this category, we can mention the recent work of Refs. [20,21], where a modification to the bilinear Galerkin scheme is introduced to resolve the locking problem for small values of the anisotropy ratio. These types of modification (“variational crimes”) may alter the consistency of the original scheme for the isotropic case. For this reason, factors tending to zero proportionally to the expected convergence rate of the scheme must be introduced to recover consistency. This procedure, however, introduces errors that may not be negligible when working on a single mesh. A review of the different schemes addressing the problem of anisotropy is found in Ref. [28]. A thorough presentation of the locking phenomenon with formal definition of locking and locking free or robust schemes is found in Ref. [5]. A completely different point of view is to try to adapt the domain or mesh or both to the anisotropy of the problem [1,2]. This approach can always be used if the anisotropy ratio is constant. It is not, however, the direction of our study, where we try to include as much as possible the anisotropy of the differential problem into the discretization operator. Other symptoms of locking can be identified by a consistent growth of the asymptotic error constant as the anisotropy ratio tends to zero, even though the optimal order of convergence may be maintained. Recent experimental evidence in the field of groundwater contamination has shown that anisotropy ratios of the order of 10^{-3} can be easily obtained [32]. In these problems, the situation is complicated by the fact that non-uniform grids need to be employed and that the anisotropy ratio and the principal directions of anisotropy are space dependent. We will thus concentrate on a practical range of anisotropy ratios of 10^{-3}–1 using the point of view of [1,2,18,28], but we will also conduct numerical experiments for ratios up to 10^{-6} to investigate the phenomenon of locking.

We will focus on the numerical solution of the dimensionless anisotropic steady-state diffusion equation

$$-\operatorname{div}(K\nabla u) = f \quad \text{in } \Omega, \tag{1a}$$

$$u = g^{\mathrm{D}} \quad \text{on } \Gamma^{\mathrm{D}}, \tag{1b}$$

$$-\boldsymbol{n}\cdot K\nabla u = g^{\mathrm{N}} \quad \text{on } \Gamma^{\mathrm{N}}, \tag{1c}$$

where $K(\boldsymbol{x})$ is a 2×2 tensor field. The solution $u(\boldsymbol{x})$ is defined for $\boldsymbol{x}$ in the polygonal domain $\Omega \subset \mathbb{R}^2$, characterized by a piecewise continuously differentiable boundary Γ that is split into two non-overlapping parts, Γ^{D} and Γ^{N}, where conditions of Dirichlet and Neumann types are imposed. We assume that either Γ^{D} or Γ^{N} may be empty and that $\Gamma = \Gamma^{\mathrm{D}} \cup \Gamma^{\mathrm{N}}$. The diffusion coefficient $K(\boldsymbol{x})$ can be represented by a 2×2 symmetric and strictly positive definite real matrix with elements that are almost everywhere defined in Ω and belong to the Sobolev space of bounded functions with bounded first derivatives $W^{1,\infty}(\Omega)$ [3]. The other symbols in (1a)–(1c) have the following meaning:

- f is the forcing term;
- $\boldsymbol{n}$ is the unit vector almost everywhere orthogonal to Γ and outward oriented from Ω;
- g^{D} is the smooth scalar function, almost everywhere defined on Γ^{D} that takes into account the Dirichlet boundary conditions;
- g^{N} is the smooth scalar function, almost everywhere defined on Γ^{N}, that takes into account the Neumann boundary condition. This function is taken explicitly dependent on the boundary edge normals $\boldsymbol{n}$, i.e. $g^{\mathrm{N}}(\boldsymbol{x},\boldsymbol{n})$, to properly handle the case of domain corners.

Under suitable assumptions on the regularity of f and the boundary functions g^{D} and g^{N} this model problem can be re-formulated in weak form, and the existence and uniqueness of the analytical solution can be proved [34].

In this work, we consider the family of second-order accurate finite volume discretizations that is generally referred as *diamond scheme* (see Ref. [17] for a recent literature review). These schemes are designed by locally approximating the diffusive flux balance for any mesh control volume [19]. In our variant, the flux is defined by a careful combination of cell averages and vertex values of the solution. The cell averages are the primary unknowns, while vertex values are obtained from cell averages by means of a Least Squares reconstruction mechanism [15]. The reconstruction is of fundamental importance to achieve optimal accuracy and, in advection–diffusion equations, needs to be employed in the approximation of both diffusive and advective fluxes [9,10]. Despite their use in more complex applications [29], this type of schemes may not appear to be competitive for the approximation of purely diffusive problems with respect to other more standard approaches. In fact, the cost of the reconstruction algorithm makes the finite volume scheme computationally less efficient with respect to, for example, finite elements and finite differences. On the other hand, the role of the reconstruction is crucial to obtain a reliable approximation of the normal fluxes [6,31,30]. In the presence of anisotropic diffusion tensors, approximation of the normal fluxes requires accurate evaluation of both normal and tangential components of the cell-interface gradients. Other recent developments also try to approximate the entire gradient as the adjoint of the divergence operator [23,24] by simultaneously using primal and dual meshes [17,22]. Instead, we propose a modification of the diamond cell approach that makes optimal use of the reconstruction algorithm to yield an accurate discretization of tangential gradients, a key ingredient for achieving robustness of the numerical scheme for limiting values of the anisotropy ratio.

The outline of the paper is as follows. In Section 2, we introduce the general notation and mesh setting framework. In Section 3, we will discuss the derivation of the proposed scheme and point out the importance of the tangential terms in the gradient approximations. Section 4 will contain our numerical experiments, which are based on standard test cases developed to test the locking phenomenon [5,20,21]. In particular, we will show that in the absence of tangential terms in the finite volume formulation the numerical solution can be obtained on a stretched domain, but with high computational efforts. We will report numerical results obtained on a variety of grids, showing that meshes aligned with the principal directions of anisotropy greatly enhance the robustness of the scheme with respect to locking. Nonetheless, we will see that this scheme is robust in the sense of Ref. [5], on a wide range of anisotropy ratios also for completely non-uniform meshes. Finally, we will show numerically that the region of convergence of the method is defined by a quadratic relationships between the anisotropy ratio and the mesh size parameter. Lastly, in Section 5, we discuss some final remarks and draw the conclusions.

2. General setting, notations, and mesh regularity assumption

The mesh $\mathscr{T}_h$ is a finite collection of non-overlapping and non-empty two-dimensional control volumes formed by simplices (triangles, in our case) which are denoted by the letter "T" and indexed by a Latin subscript, e.g. i, j, k. For example, T_i is the ith control volume (cell) of the mesh $\mathscr{T}_h = \{\mathsf{T}_i\}$. According to the definition given in Ref. [14], the parameter h that labels the mesh $\mathscr{T}_h$ is called *the mesh size*, and is formally given by the supremum of the mesh control volume diameters, $h = \max_{\mathsf{T}_i \in \mathscr{T}_h} \mathrm{diam}\, \mathsf{T}_i$. We assume that, for every possible choice of h, $\mathscr{T}_h$ covers the polygonal domain $\Omega \subset \mathbb{R}^2$ in the sense that $\overline{\Omega} = \cup_{\mathsf{T}_i \in \mathscr{T}_h} \mathsf{T}_i$. The edges are denoted by the letter "e" and labeled by a couple of Latin indices, e.g. e_{ij}. When e_{ij} is an internal edge, two control volumes T_i and T_j, such that $\mathsf{e}_{ij} = \partial\mathsf{T}_i \cap \partial\mathsf{T}_j$, must exist. When e_{ij} is a boundary edge, the first index, e.g. i, always refers to the unique control volume T_i such that $\mathsf{e}_{ij} \in \partial\mathsf{T}_i \cap \Gamma$. The second index is defined by a suitable boundary numbering system (like a sort of *ghost cell*). When dealing with an internal edge, the symbols e_{ij} and e_{ji} denote the same edge and are considered only once (for example, by taking the representative with $i < j$). The mesh vertices are denoted by the symbol "v" and indexed by Greek letters (α, β, γ).

The development of this finite volume method requires the use of the following mesh entities:

- $\mathscr{T}_h$ is the set of the control volumes (i.e., *the mesh*);
- $\mathscr{E}_h (= \mathscr{E}_h^{\rm I} \cup \mathscr{E}_h^{\rm B})$ is the set of all mesh edges, formed by the union of *internal* and *boundary edges*. The set $\mathscr{E}_h^{\rm B}$ is also subdivided into
  - $\mathscr{E}_h^{\rm D}$ is the set of boundary edges where a *Dirichlet* condition is imposed;
  - $\mathscr{E}_h^{\rm N}$ is the set of boundary edges where a *Neumann* condition is imposed.

  Clearly, $\mathscr{E}_h^{\rm I} \cap \mathscr{E}_h^{\rm B} = \emptyset$, $\mathscr{E}_h^{\rm B} = \mathscr{E}_h^{\rm D} \cup \mathscr{E}_h^{\rm N}$, and $\mathscr{E}_h^{\rm D} \cap \mathscr{E}_h^{\rm N} = \emptyset$;
- $\mathscr{V}_h (= \mathscr{V}_h^{\rm I} \cup \mathscr{V}_h^{\rm B})$ is the set of all mesh vertices, formed by the union of *internal* and *boundary vertices*. The set $\mathscr{V}_h^{\rm B}$ is also subdivided into
  - $\mathscr{V}_h^{\rm D}$ is the set of boundary vertices where a *Dirichlet* condition is imposed;
  - $\mathscr{V}_h^{\rm N}$ is the set of boundary vertices where a *Neumann* condition is imposed.

  Clearly, $\mathscr{V}_h^{\rm I} \cap \mathscr{V}_h^{\rm B} = \emptyset$, $\mathscr{V}_h^{\rm B} = \mathscr{V}_h^{\rm D} \cup \mathscr{V}_h^{\rm N}$, and $\mathscr{V}_h^{\rm D} \cap \mathscr{V}_h^{\rm N} = \emptyset$.

2.1. Basic geometric quantities

All the quantities related to T_i are consistently labeled by the same index i. For example, $|\mathsf{T}_i|$ is the area of T_i, and $\boldsymbol{x}_i$ is the position vector of the gravity center of T_i. Analogously, the quantities related to the mesh vertex v_α are consistently labeled by the same index α. For example, $\boldsymbol{x}_\alpha$ is the position vector of v_α. We assume that Γ has a counterclockwise orientation and label the boundary edge that precedes the vertex v_α by the minus sign "$-$" and the boundary edge that follows v_α by the plus sign "$+$". The unit normal and tangential vectors on these edges are $\boldsymbol{n}_\alpha^\pm$ and $\boldsymbol{t}_\alpha^\pm$. The data associated to the boundary vertex and its two incident boundary edges $\mathsf{e}_\alpha^\pm$ are denoted by g_α^- and g_α^+. For convenience, we set $g_\alpha^\pm = 0$ if $\mathsf{v}_\alpha \in \mathscr{V}_h^{\rm I} \cup \mathscr{V}_h{}^{\rm D}$ and unify the formulation for internal and boundary vertices. Thus,

$$g_\alpha^\pm = \begin{cases} 0 & \text{if } \mathsf{v}_\alpha \in \mathscr{V}_h^{\rm I} \cup \mathscr{V}_h^{\rm D}, \\ g^{\rm N}(\boldsymbol{x}_\alpha, \boldsymbol{n}_\alpha^\pm) & \text{if } \mathsf{v}_\alpha \in \mathscr{V}_h^{\rm N}. \end{cases}$$

The quantities related to e_{ij} are consistently labeled by ij and are:

- $|\mathsf{e}_{ij}|$ is the length of $\mathsf{e}_{ij} \in \mathscr{E}_h$;
- $\boldsymbol{x}_{ij}$ is the position vector of the midpoint center of e_{ij};
- $\boldsymbol{n}_{ij}$ is the unit vector orthogonal to e_{ij} and oriented from T_i to T_j if $\mathsf{e}_{ij} \in \mathscr{E}_h^{\rm I}$, outward of Ω if $\mathsf{e}_{ij} \in \mathscr{E}_h^{\rm B}$;
- $\boldsymbol{t}_{ij}$ is the unit vector parallel to e_{ij} and such that $(\boldsymbol{t}_{ij}, \boldsymbol{n}_{ij})$ forms a counterclockwise orthogonal reference system for $\mathbb{R}^2$;
- $\tilde{\boldsymbol{x}}_{ij}$ is the position vector of the orthogonal projection of $\boldsymbol{x}_i$ on the line containing the edge $\mathsf{e}_{ij} \in \mathscr{E}_h$;
- $\tilde{\lambda}_\alpha^{ij}$ is the coordinate of $\tilde{\boldsymbol{x}}_{ij}$ with respect to $\mathsf{v}_\alpha \in \mathsf{e}_{ij}$, and given by $\tilde{\lambda}_\alpha^{ij} = 1 - |\tilde{\boldsymbol{x}}_{ij} - \boldsymbol{x}_\alpha| / |\mathsf{e}_{ij}|$; the baricentric coordinates of $\tilde{\boldsymbol{x}}_{ij}$ are such that $0 \leqslant \tilde{\lambda}_\alpha^{ij} \leqslant 1$ when $\tilde{\boldsymbol{x}}_{ij} \in \mathsf{e}_{ij}$.
- $h_{ij} = (\tilde{\boldsymbol{x}}_{ij} - \boldsymbol{x}_i) \cdot \boldsymbol{n}_{ij}$ is the distance between the gravity center of T_i and the edge e_{ij};
- $H_{ij} = (\boldsymbol{x}_j - \boldsymbol{x}_i) \cdot \boldsymbol{n}_{ij} = h_{ij} + h_{ji}$ is the *effective distance* between the gravity centers of T_i and T_j when $\mathsf{e}_{ij} \in \mathscr{E}_h^{\rm I}$ (Fig. 1).

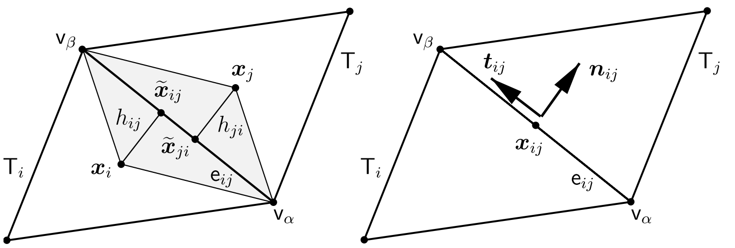

Fig. 1. The diamond cell geometry for edge flux calculation.

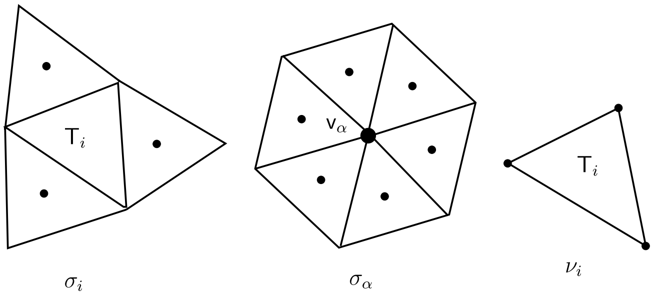

Fig. 2. The finite volume stencils.

The scheme formulation is expressed by using summations over stencils of cells, edges, and vertices (Fig. 2). The set σ_i indicates the cells sharing an edge with T_i; σ'_i indicates the "ghost" cells sharing an edge with T_i; and σ_α indicates the cells surrounding vertex v_α. Thus, e_{ij} is a boundary edge of T_i if $j \in \sigma'_i$ while it is the internal edge shared by T_i and T_j if $j \in \sigma_i$. The index j running on the set $\sigma_i \cup \sigma'_i$ labels all the edges e_{ij} forming the boundary of T_i. Summations may also be taken over index sets of mesh vertices. We denote by v_i the set of vertices of the cell T_i, and by $v_{ij} = \{\alpha,\beta\}$ the set of vertices of the edge e_{ij}. If $\alpha,\beta \in v_{ij}$ we can write $|\mathsf{e}_{ij}| = |\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha|$ because v_α and v_β are the two vertices of e_{ij}.

2.2. The cell average operator

The *cell average operator* is the vector-valued functional $\mathscr{A}_h : L^1(\Omega)\mapsto\mathbb{R}^{N_\mathsf{T}}$ such that

$$\mathscr{A}_h(u)|_i = \frac{1}{|\mathsf{T}_i|}\int_{\mathsf{T}_i} u(\boldsymbol{x})\,\mathrm{d}V \quad \text{for } u \in L^1(\Omega) \text{ and } 0 \leqslant i \leqslant N_\mathsf{T}.$$

Throughout the paper, we will also use the shortcut $\mathscr{A}_i(u) = \mathscr{A}_h(u)|_i$.

Remark 1. The cell average operator $\mathscr{A}_h$ could be alternatively defined as follows. Let $P^0(\mathscr{T}_h)$ be the subspace of the Hilbert space $L^2(\Omega)$ consisting of the functions having constant restriction on the control volumes of the triangulation $\mathscr{T}_h$. The functional operator $\mathscr{A}_h$ is then the orthogonal projector from $L^2(\Omega)$ onto $P^0(\mathscr{T}_h)$. The dimension of $P^0(\mathscr{T}_h)$ is equal to N_T and the identification between $P^0(\mathscr{T}_h)$ and $\mathbb{R}^{N_\mathsf{T}}$ can be readily established.

2.3. Mesh regularity assumption

The finite volume method presented in this paper approximates the solution of (1a)–(1c) by using the family of grids $\{\mathscr{T}_h\}$. Each element of $\{\mathscr{T}_h\}$, distinguished by the label h, is a *regular partition* of the domain Ω; we have the following:

Assumption 2 (*Mesh regularity*)

(i) All the triangulations $\mathscr{T}_h$ for $h \leqslant h_0$ are conformal in accordance with the definition of Ref. [14].
(ii) All the triangulations $\mathscr{T}_h$ for $h \leqslant h_0$ are weakly acute; i.e., all the mesh angles are less than or equal to $\pi/2$.

The first item of the regularity assumption is usually met in finite element analysis and is quite reasonable in finite volume formulations. Its aim is to prevent the use of degenerate triangles, either too obtuse or too acute, in the approximation process for $h \rightarrow 0$. The second item of the regularity assumption ensures that the orthogonal projection of the gravity center of any control volume T_i onto each one of the edges $\mathsf{e}_{ij} \in \partial\mathsf{T}_i$ is an internal point of e_{ij}. Weakly acute triangulations always satisfy this property, but more general partitions may also be considered even if obtuse triangles are present. For simplicity, we assume that any member of the family $\{\mathscr{T}_h \text{ for } h \leqslant h_0\}$ is weakly acute, thus accepting at most the case where $\tilde{\boldsymbol{x}}_{ij}$ coincides with one of the vertices of edge e_{ij}. Much stronger assumptions are needed to be able to demonstrate the coercivity (and thus the non-singularity) of the finite volume operator [15,16]. However, in practical calculations the above mesh regularity assumptions are generally enough. On the other hand, the proposed scheme will work also on non-conforming grids, as discussed in Ref. [16].

3. Finite volume formulation

The first step of the derivation of the proposed cell-centered finite volume method for the solution of (1a)–(1c) is to reformulate the governing equation (1a) on the generic cell T_i in integral (also said "conservative") form. Applying the Gauss divergence theorem, we obtain the set of balance equations

$$-\int_{\partial\mathsf{T}_i} \boldsymbol{n}\cdot K\nabla u\,\mathrm{d}S = \int_{\mathsf{T}_i} f\,\mathrm{d}V \quad \text{for } \mathsf{T}_i\in\mathscr{T}_h. \tag{2}$$

Let $u_h\in\mathbb{R}^{N_\mathsf{T}}$ be the approximation of the N_T-sized vector of cell averages $\mathscr{A}_h(u)$. Thus, u_h can also be interpreted as the piece-wise constant function in $L^2(\Omega)$ whose restriction on T_i approximates $\mathscr{A}_i(u)$ (see the remark at the end of the Section 2.2). The restriction of u_h on T_i is denoted by u_i. The cell-centered finite volume discretization mimics Eqs. (2) by introducing a set of *discrete* balance equations that correlate each u_i to the balance of the numerical fluxes across control volume boundary $\partial\mathsf{T}_i$:

$$\frac{1}{|\mathsf{T}_i|}\sum_{j\in\sigma_i\cup\sigma_i'} |\mathsf{e}_{ij}|\mathsf{G}_{ij}(u_h) = \mathsf{s}_i \quad \text{for } \mathsf{T}_i\in\mathscr{T}_h, \tag{3}$$

where $\mathsf{G}_{ij}(u_h)$ denotes the numerical integral of the diffusive flux on the control volume edge $\mathsf{e}_{ij}\in\partial\mathsf{T}_i$ and s_i is the cell-average of the source term f in (1a).

The functional dependence of $\mathsf{G}_{ij}(u_h)$ on the cell average approximation vector u_h must be carefully devised in the cell-centered finite volume framework in order to provide a proper formulation of the numerical discretization method. To this aim, let $\mathscr{G}^{\lozenge}_{ij}(u_h)$ denote the discrete constant gradient that we assume to be uniquely defined on edge $\mathsf{e}_{ij}\in\mathscr{E}_h$. The numerical diffusive flux is then

$$\mathsf{G}_{ij}(u_h) = \begin{cases} -\boldsymbol{n}_{ij}\cdot K_{ij}\mathscr{G}^{\lozenge}_{ij}(u_h) & \text{if } \mathsf{e}_{ij}\in\mathscr{E}^{\mathrm{I}}_h\cup\mathscr{E}^{\mathrm{D}}_h, \\ g^{\mathrm{N}}_{ij} & \text{if } \mathsf{e}_{ij}\in\mathscr{E}^{\mathrm{N}}_h. \end{cases} \tag{4}$$

The first part of the numerical flux formula is derived by approximating the co-normal derivative of u that appears in (2), $\boldsymbol{n}\cdot K\nabla u$, by its discrete version $\boldsymbol{n}_{ij}\cdot K_{ij}\mathscr{G}^{\lozenge}_{ij}(u_h)$. To this purpose, we apply the second-order mid-point quadrature rule to the integral (2) decomposed on every edge $\mathsf{e}_{ij}\in\partial\mathsf{T}_i$ and introduce a suitably defined mean diffusion tensor K_{ij} on e_{ij}. Convenient choices for the mean edge-based diffusion tensor are $K_{ij}=K(\boldsymbol{x}_{ij})$ or the line-integral average

$$K_{ij} = \frac{1}{|\mathsf{e}_{ij}|}\int_{\mathsf{e}_{ij}} K(\boldsymbol{x})\,\mathrm{d}S,$$

that has to be interpreted component-wise. The second part of the numerical flux formula accounts for Neumann boundary conditions by considering the function $g^{\mathrm{N}}(\boldsymbol{x},\boldsymbol{n}(\boldsymbol{x}))$ of the right-hand side of (1c) on the boundary edge $\mathsf{e}_{ij}\in\partial\mathsf{T}_i\cap\mathscr{E}^{\mathrm{N}}_h$. In analogy with the definition of the components of the conductivity tensor, the discrete flux term g^{N}_{ij} can be given by $g^{\mathrm{N}}_{ij}=g^{\mathrm{N}}(\boldsymbol{x}_{ij},\boldsymbol{n}_{ij})$ or

$$g^{\mathrm{N}}_{ij} = \frac{1}{|\mathsf{e}_{ij}|}\int_{\mathsf{e}_{ij}} g^{\mathrm{N}}(\boldsymbol{x},\boldsymbol{n}(\boldsymbol{x}))\,\mathrm{d}S.$$

3.1. Approximation of the solution gradient at mesh edges

The first step for the formulation of the discrete gradient $\mathscr{G}^{\lozenge}_{ij}(u_h)$ for internal and Dirichlet-boundary edges consists in defining the *one-sided* gradient $\mathscr{G}_{ij}(u_h)$ associated to the edge e_{ij} from the interior of triangle T_i. At every internal edge $\mathsf{e}_{ij}\in\mathscr{E}^{\mathrm{I}}_h$, the discrete gradient $\mathscr{G}^{\lozenge}_{ij}(u_h)$ is uniquely calculated by averaging the two one-sided contributions that are built within cells T_i and T_j sharing e_{ij}. At every boundary edge of Dirichlet type $\mathsf{e}_{ij} \in \mathscr{E}_h^{\mathrm{D}}$, the discrete gradient $\boldsymbol{\mathscr{G}}_{ij}^{\lozenge}(u_h)$ coincides with the gradient defined within the (unique) cell T_i to which edge e_{ij} belongs. Second order accuracy of the finite volume scheme is achieved by ensuring that the one-sided gradients $\boldsymbol{\mathscr{G}}_{ij}(u_h)$ are exact when applied to a linear function. To this aim, we calculate $G_{ij}(u_h)$ using a suitable interpolation of the cell-average u_i and of the solution approximations u_α and u_β at the vertices v_α and v_β of edge e_{ij}. Note that u_i is a second-order accurate approximation of the solution value at the center of gravity $\boldsymbol{x}_i$ of cell T_i. Instead, the vertex approximations u_α and u_β are provided by the formally second-order accurate reconstruction algorithm that will be described in Section 3.2. The formula for the one-sided gradient is then developed by approximating the cell-average of the solution gradient ∇u on triangle $\mathsf{T}_{i\alpha\beta}$ defined by the gravity center of T_i and the vertices of edge e_{ij} (see Fig. 3):

$$\boldsymbol{\mathscr{G}}_{ij}(u_h) \approx \frac{1}{|\mathsf{T}_{i\alpha\beta}|} \int_{\mathsf{T}_{i\alpha\beta}} \nabla u \,\mathrm{d}V.$$

Application of the Gauss–Green lemma yields:

$$\frac{1}{|\mathsf{T}_{i\alpha\beta}|} \int_{\mathsf{T}_{i\alpha\beta}} \nabla u \,\mathrm{d}V = \frac{1}{|\mathsf{T}_{i\alpha\beta}|} \int_{\partial\mathsf{T}_{i\alpha\beta}} \boldsymbol{n} u \,\mathrm{d}S,$$

where $\partial\mathsf{T}_{i\alpha\beta} = \mathsf{e}_{i\alpha} \cup \mathsf{e}_{i\beta} \cup \mathsf{e}_{ij}$. The three line integrals on $\mathsf{e}_{i\alpha}$, $\mathsf{e}_{i\beta}$, and e_{ij} are then approximated by means of the trapezoidal rule, and, taking into account that $|\mathsf{e}_{i\alpha}|\boldsymbol{n}_{i\alpha} + |\mathsf{e}_{i\beta}|\boldsymbol{n}_{i\beta} + |\mathsf{e}_{ij}|\boldsymbol{n}_{ij} = 0$ and $|\mathsf{T}_{i\alpha\beta}| = |\mathsf{e}_{ij}|h_{ij}/2$, we obtain:

$$\boldsymbol{\mathscr{G}}_{ij}(u_h) = \frac{1}{h_{ij}} \left[u_\alpha \left(\boldsymbol{n}_{ij} + \boldsymbol{n}_{i\alpha} \frac{|\mathsf{e}_{i\alpha}|}{|\mathsf{e}_{ij}|} \right) + u_\beta \left(\boldsymbol{n}_{ij} + \boldsymbol{n}_{i\beta} \frac{|\mathsf{e}_{i\beta}|}{|\mathsf{e}_{ij}|} \right) - u_i \boldsymbol{n}_{ij} \right], \tag{5}$$

where the three unit normal vectors $\boldsymbol{n}_{i\alpha}$, $\boldsymbol{n}_{i\beta}$, and $\boldsymbol{n}_{ij}$ point out of the integration domain as shown by Fig. 3. The normal and tangential components of the one-sided gradient, denoted by $\mathscr{G}_{ij}^{(n)}(u_h)$ and $\mathscr{G}_{ij}^{(t)}(u_h)$, are the projection of the right-hand side of (5) onto the mutually orthogonal directions of the unit vectors $\boldsymbol{n}_{ij}$ and $\boldsymbol{t}_{ij}$. Thus, we obtain:

$$\boldsymbol{\mathscr{G}}_{ij}(u_h) = \mathscr{G}_{ij}^{(n)}(u_h)\boldsymbol{n}_{ij} + \mathscr{G}_{ij}^{(t)}(u_h)\boldsymbol{t}_{ij}, \tag{6}$$

where

$$\mathscr{G}_{ij}^{(n)}(u_h) = \boldsymbol{n}_{ij} \cdot \boldsymbol{\mathscr{G}}_{ij}(u_h) = \frac{1}{h_{ij}} \left[\tilde{\lambda}_\alpha^{ij} u_\alpha + \tilde{\lambda}_\beta^{ij} u_\beta - u_i \right], \tag{7a}$$

$$\mathscr{G}_{ij}^{(t)}(u_h) = \boldsymbol{n}_{ij} \cdot \boldsymbol{\mathscr{G}}_{ij}(u_h) = \frac{1}{h_{ij}} \left[\tilde{\mu}_\alpha^{ij} u_\alpha + \tilde{\mu}_\beta^{ij} u_\beta \right], \tag{7b}$$

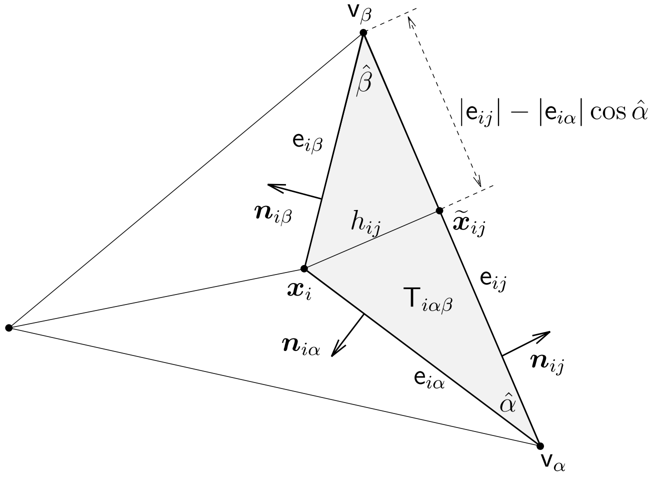

Fig. 3. The edge geometry for the definition of the one-sided gradient $\boldsymbol{\mathscr{G}}_{ij}(u_h)$.

and:

$$\tilde{\lambda}^{ij}_{\alpha} = \boldsymbol{n}_{ij} \cdot \left(\boldsymbol{n}_{ij} + \boldsymbol{n}_{i\alpha} \frac{|\mathsf{e}_{i\alpha}|}{|\mathsf{e}_{ij}|} \right) = \left(1 - \frac{|\mathsf{e}_{i\alpha}|}{|\mathsf{e}_{ij}|} \cos \hat{\alpha} \right), \tag{8a}$$

$$\tilde{\lambda}^{ij}_{\beta} = \boldsymbol{n}_{ij} \cdot \left(\boldsymbol{n}_{ij} + \boldsymbol{n}_{i\beta} \frac{|\mathsf{e}_{i\beta}|}{|\mathsf{e}_{ij}|} \right) = \left(1 - \frac{|\mathsf{e}_{i\beta}|}{|\mathsf{e}_{ij}|} \cos \hat{\beta} \right), \tag{8b}$$

$$\tilde{\mu}^{ij}_{\alpha} = \boldsymbol{t}_{ij} \cdot \boldsymbol{n}_{i\alpha} \frac{|\mathsf{e}_{i\alpha}|}{|\mathsf{e}_{ij}|} = -1, \tag{8c}$$

$$\tilde{\mu}^{ij}_{\beta} = \boldsymbol{t}_{ij} \cdot \boldsymbol{n}_{i\beta} \frac{|\mathsf{e}_{i\beta}|}{|\mathsf{e}_{ij}|} = +1. \tag{8d}$$

As shown in Fig. 3, the angles $\hat{\alpha}$ and $\hat{\beta}$ in (8a) and (8b) satisfy the conditions $\boldsymbol{n}_{ij} \cdot \boldsymbol{n}_{i\alpha} = \cos(\pi - \hat{\alpha})$ and $\boldsymbol{n}_{ij} \cdot \boldsymbol{n}_{i\beta} = \cos(\pi - \hat{\beta})$. Note that $\tilde{\lambda}^{ij}_{\alpha}$ and $\tilde{\lambda}^{ij}_{\beta}$ are the baricentric coordinates of $\tilde{\boldsymbol{x}}_{ij} \in \mathsf{e}_{ij}$ (the orthogonal projection of the cell center $\boldsymbol{x}_i$ onto edge e_{ij}) with respect to the vertices v_{α} and v_{β}. As a result, it is easy to recognize that Eq. (7a) actually contains the linear interpolation of the vertex values u_{α} and u_{β} at $\tilde{\boldsymbol{x}}_{ij}$, as given by:

$$u_{ij} = \tilde{\lambda}^{ij}_{\alpha} u_{\alpha} + \tilde{\lambda}^{ij}_{\beta} u_{\beta}. \tag{9}$$

In view of the second mesh regularity assumption, the baricentric coordinates of $\tilde{\boldsymbol{x}}_{ij}$ are non-negative numbers bounded from above by 1 and, consequently, u_{ij} is a *convex* interpolation of u_{α} and u_{β}. Finally, by substituting (7a), (7b) and (8a)–(8d) in (6), and using the definition given in (9), we obtain the final formulation of the one-sided gradient at e_{ij} within T_i:

$$\mathscr{G}_{ij}(u_h) = \frac{u_{ij} - u_i}{h_{ij}} \boldsymbol{n}_{ij} + \frac{u_{\beta} - u_{\alpha}}{|\mathsf{e}_{ij}|} \boldsymbol{t}_{ij}. \tag{10}$$

3.1.1. Internal edges

A unique definition of the numerical edge flux is required to obtain a *conservative* formulation of the numerical diffusive flux $\mathsf{G}_{ij}(u_h)$. To this aim, given an internal edge e_{ij}, the two contributions arising from the one-sided numerical gradients, $\mathscr{G}_{ij}(u_h)$ built in T_i and $\mathscr{G}_{ji}(u_h)$ built in T_j, are averaged by:

$$\mathscr{G}^{\Diamond}_{ij}(u_h) = W_{ij} \mathscr{G}_{ij}(u_h) + W_{ji} \mathscr{G}_{ji}(u_h), \tag{11}$$

where the non-negative weights are:

$$W_{ij} = 1 - W_{ji} = \frac{h_{ij}}{H_{ij}}.$$

This results in the following definition of the numerical flux:

$$\mathsf{G}_{ij}(u_h) = \left(\frac{u_j - u_i}{H_{ij}} \right) \kappa^{(n)}_{ij} + \left(\frac{u_{\beta} - u_{\alpha}}{|\mathsf{e}_{ij}|} \right) \kappa^{(t)}_{ij},$$

where the coefficients $\kappa^{(n)}_{ij}$ and $\kappa^{(t)}_{ij}$ contain the normal and tangential projections of the co-normal vector $\boldsymbol{K n}_{ij}$, i.e.:

$$\kappa^{(n)}_{ij} = \boldsymbol{n}_{ij} \cdot K_{ij} \boldsymbol{n}_{ij},$$
$$\kappa^{(t)}_{ij} = \left(\tilde{\lambda}^{ij}_{\alpha} + \tilde{\lambda}^{ji}_{\beta} - 1 \right) \frac{|\mathsf{e}_{ij}|}{H_{ij}} \boldsymbol{n}_{ij} \cdot K_{ij} \boldsymbol{n}_{ij} + \boldsymbol{n}_{ij} \cdot K_{ij} \boldsymbol{t}_{ij}.$$

Remark 3. The tangential term disappears if $K = \alpha I$, i.e. K is isotropic. It is the tangential term that bears the responsibility for the approximation of the anisotropic flux term. Accuracy of the vertex reconstruction algorithm is thus of fundamental importance to achieve full accuracy of the tangential component of the numerical flux and thus good performance for strongly anisotropic diffusion tensors.

3.1.2. Dirichlet boundary edges

When $\mathsf{e}_{ij} \in \mathscr{E}_h^{\mathrm{D}}$, the edge gradient $\mathscr{G}_{ij}^{\Diamond}(u_h)$ is set equal to the unique one-sided edge gradient $\mathscr{G}_{ij}(u_h)$. Thus, we have

$$\mathscr{G}_{ij}^{\Diamond}(u_h) = \frac{u_{ij} - u_i}{h_{ij}} \boldsymbol{n}_{ij} + \frac{u_\beta - u_\alpha}{|\mathsf{e}_{ij}|} \boldsymbol{t}_{ij},$$

so that

$$\mathsf{G}_{ij}(u_h) = -\frac{u_i}{h_{ij}} \kappa_{ij}^{(n)} + \frac{\kappa_{\beta,ij}^{(t)} u_\beta - \kappa_{\alpha,ij}^{(t)} u_\alpha}{|\mathsf{e}_{ij}|}.$$

Tangential contributions involve the boundary coefficients

$$\kappa_{\alpha,ij}^{(t)} = -\frac{\tilde{\lambda}_\beta^{ij} |\mathsf{e}_{ij}|}{h_{ij}} \boldsymbol{n}_{ij} \cdot K_{ij} \boldsymbol{n}_{ij} - \boldsymbol{n}_{ij} \cdot K_{ij} \boldsymbol{t}_{ij},$$
$$\kappa_{\beta,ij}^{(t)} = +\frac{\tilde{\lambda}_\alpha^{ij} |\mathsf{e}_{ij}|}{h_{ij}} \boldsymbol{n}_{ij} \cdot K_{ij} \boldsymbol{n}_{ij} + \boldsymbol{n}_{ij} \cdot K_{ij} \boldsymbol{t}_{ij}.$$

Since at least one of the two boundary vertices v_α and v_β is of Dirichlet type, we may argue that the relation $u_\alpha = g_\alpha^{\mathrm{D}}$ or $u_\beta = g_\beta^{\mathrm{D}}$ (or both, simultaneously) must be true. We remark that the tangential component of the diffusive flux on the boundary edge is proportional to $u_\beta - u_\alpha$. This clearly implies that this flux component is completely determined by the Dirichlet boundary function $g^{\mathrm{D}}(\boldsymbol{x})$ when both $u_\alpha = g_\alpha^{\mathrm{D}}$ and $u_\beta = g_\beta^{\mathrm{D}}$ are used.

3.2. Vertex reconstruction and Neumann boundary conditions

To complete the description of this finite volume method, we must still explain how the vertex value u_α, which is defined for every $\mathsf{v}_\alpha \in \mathscr{V}_h$, depends on the approximate cell averages u_h. If v_α is a Dirichlet boundary vertex, we set $u_\alpha = g^{\mathrm{D}}(\boldsymbol{x}_\alpha)$. For internal and Neumann boundary vertices, we approximate the solution by a weighted mean of the cell averages surrounding the vertex. Careful consideration must be given to the case of Neumann boundary conditions, as they play a crucial role in the investigation of the mesh locking phenomenon. In general, we look for a linear Least Squares approximation of the cell-averaged data set $\{(\boldsymbol{x}_k, u_k), k \in \sigma_\alpha\}$ on the co-volume $\mathscr{V}_\alpha = \bigcup_{k\in\sigma_\alpha} \mathsf{T}_k$. This linear approximation takes on the form

$$\mathscr{R}_\alpha(\boldsymbol{x}; u_h) = a + \mathbf{b} \cdot (\boldsymbol{x} - \boldsymbol{x}_\alpha) \quad \text{for } \boldsymbol{x} \in \mathscr{V}_\alpha.$$

The coefficients $(a, \mathbf{b}^{\mathrm{T}})^{\mathrm{T}}$ are the minimizers of the Least Squares functional

$$\mathscr{J}(a, \mathbf{b}^{\mathrm{T}}) = \sum_{k\in\sigma_\alpha} W_k [a + \mathbf{b} \cdot (\boldsymbol{x}_k - \boldsymbol{x}_\alpha) - u_k]^2, \tag{12}$$

with $W_k = |\mathsf{T}_k| / \sum_{s\in\sigma_\alpha} |\mathsf{T}_s|$. The minimization takes place over $\mathbb{R}^3$ if v_α is an internal vertex. This procedure is thoroughly discussed in Ref. [15], and we refer the reader to this paper for the evaluation of the reconstruction weights in this case.

However, we need to develop a generalization of this strategy for a more accurate reconstruction of the solution value at a vertex where a Neumann boundary condition must be imposed. To this aim, starting from the development of Ref. [11], we implement condition (1c) in our formulation by enforcing the linear constraint

$$\boldsymbol{n}_\alpha^{\pm} \cdot (K_\alpha \mathbf{b}) = g^{\mathrm{N}}(\boldsymbol{x}_\alpha, \boldsymbol{n}_\alpha^{\pm}),$$

on the two boundary edges to the left and to the right of v_α and setting $K_\alpha = K(\boldsymbol{x}_\alpha)$. The reconstruction weights are provided by the minimization of the Least Squares functional (12) with respect to the unknown variables $(a, \mathbf{b}^{\mathrm{T}})^{\mathrm{T}}$ over the constraint region

$$\mathbb{O} = \{(a', \mathbf{b}'^{,\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^3 \text{ such that } \boldsymbol{n}_\alpha^{\pm} \cdot (K_\alpha \mathbf{b}') = g_\alpha^{\pm}\}. \tag{13}$$

The implementation of the Least Squares problem proceeds as follows. Let us denote by N_α the cardinality of the set σ_α. We introduce the N_α-sized vector $\mathbf{u}_\alpha = (u_1, \ldots, u_{N_\alpha})^{\mathrm{T}}$, whose components are the cell averages of the cells sharing vertex v_α. We construct the diagonal matrix $\Lambda = \mathrm{diag}(W_1, \ldots, W_{N_\alpha})$ using the coefficients of (12), and the $N_\alpha \times 3$ rectangular matrix $\mathscr{A} = [\mathbf{1}, \boldsymbol{\delta x}, \boldsymbol{\delta y}]$, this latter being defined by the three N_α-sized column vectors $\mathbf{1} = (1,\ldots,1)^{\mathrm{T}}$, $\delta x = (x_1 - x_\alpha, \ldots, x_{N_\alpha} - x_\alpha)^{\mathrm{T}}$, and $\delta y = (y_1 - y_\alpha, \ldots, y_{N_\alpha} - y_\alpha)^{\mathrm{T}}$, where (x_k, y_y) for $k = 1, \ldots, N_\alpha$ are the coordinates of the gravity center of cell T_k. We also consider the matrix $\mathscr{B}$ to conveniently incorporate the Neumann constraints (13). The definition of $\mathscr{B}$ requires some special care as two different cases must be taken into account: case (i), the two normal vectors $\boldsymbol{n}_\alpha^\pm$ are parallel, i.e. $\boldsymbol{n}_\alpha^+ = \boldsymbol{n}_\alpha^-$, and case (ii), the two normal vectors $\boldsymbol{n}_\alpha^\pm$ are pointing to two different directions. The constraint matrix $\mathscr{B}$ is given by

$$\text{case (i)}: \ \mathscr{B} = \begin{bmatrix} 0 \\ K_\alpha \boldsymbol{n}_\alpha^+ \end{bmatrix}; \quad \text{case (ii)}: \ \mathscr{B} = \begin{bmatrix} 0 & 0 \\ K_\alpha \boldsymbol{n}_\alpha^+ & K_\alpha \boldsymbol{n}_\alpha^- \end{bmatrix}.$$

For polygonal domains, case (i) always occurs when v_α is an interior point of a straight boundary, while case (ii) occurs when v_α is a corner connecting two distinct straight boundary lines. Case (ii) also covers the situation in which v_α is a boundary corner point but belongs to a single triangular cell with two boundary edges both incident to this vertex. For a more general domain shape, the boundary has to be properly approximated by the sequence of triangulations built for $h \to 0$, so that polygonal numerical domains are obtained. It is worth noting that, in this approximation process, case (i) is the limiting situation that takes place when the boundary is locally regular, i.e. at least continuously differentiable, at v_α. Instead, case (ii) occurs when the boundary is only Lipschitz continuous at v_α, i.e. when v_α is a domain corner.

The constrained least square procedure yields the augmented system

$$\begin{bmatrix} \mathscr{A}^{\mathrm{T}} \Lambda \mathscr{A} & \mathscr{B} \\ \mathscr{B}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} (a, \mathbf{b}^{\mathrm{T}})^{\mathrm{T}} \\ \mu \end{bmatrix} = \begin{bmatrix} \mathscr{A}^{\mathrm{T}} \Lambda \mathbf{u}_\alpha \\ \mathbf{g}_\alpha \end{bmatrix}, \tag{14}$$

where μ is the vector of Lagrangian multipliers for the Neumann constraints and $\mathbf{g}_\alpha$ is the vector of edge-based data $g_\alpha^\pm$. Consistently with the double definition of $\mathscr{B}$, we take $\mu = (\mu^+)$ and $\mathbf{g}_\alpha = (g_\alpha^+)$ in case (i) and $\mu = (\mu^+, \mu^-)^{\mathrm{T}}$ and $\mathbf{g}_\alpha = (g_\alpha^+, g_\alpha^-)^{\mathrm{T}}$ in case (ii). It is easy to see that $\mathscr{A}$ is a maximum rank matrix because mesh assumption 2-(ii) implies that the cell centers cannot be all simultaneously collinear with vertex v_α. Thus, matrix $\mathscr{A}^{\mathrm{T}} \Lambda \mathscr{A}$ must have maximum rank and is non-singular. Let us denote by $\mathscr{K}$ the (non-unique) matrix whose columns are a basis for the kernel of $\mathscr{B}^{\mathrm{T}}$. By definition, $\mathscr{K}$ is a maximum rank matrix satisfying $\mathscr{B}^{\mathrm{T}} \mathscr{K} = 0$. From the assumption that the mesh is weakly acute it follows that also the projected matrix $\mathscr{K}^{\mathrm{T}} \mathscr{A}^{\mathrm{T}} \Lambda \mathscr{A} \mathscr{K}$ is non-singular. From this last consideration and since $\mathscr{B}^{\mathrm{T}}$ is by definition a maximum rank matrix for both cases (i) and (ii), the non-singularity of the system matrix in (14) is readily deduced [13].

The solution of the augmented system (14) for the unknown $a = \mathscr{R}_\alpha(\boldsymbol{x}_\alpha; u_h) = u_\alpha$ can be formally expressed by the null space formulation. This approach allows us to write u_α as a weighted average of the cell-centered data $\mathbf{u}_\alpha$ and the edge-based data $\mathbf{g}_\alpha$. The set of cell-centered and boundary edge weights are denoted by $\{W_{\alpha,k}\}$ and $\{W_\alpha^\pm\}$, respectively. Taking $W_\alpha^\pm = 0$ if v_α is an internal vertex and $W_\alpha^+ = W_\alpha^-$ if v_α is a Neumann vertex in case (i) makes it possible to derive a single reconstruction formula, which reads as

$$u_\alpha = \begin{cases} \sum\limits_{k \in \sigma_\alpha} W_{\alpha,k} u_k + W_\alpha^- g_\alpha^- + W_\alpha^+ g_\alpha^+ & \text{for } \mathsf{v}_\alpha \in \mathscr{V}_h^{\mathrm{I}} \cup \mathscr{V}_h^{\mathrm{N}}, \\ g^{\mathrm{D}}(\boldsymbol{x}_\alpha) & \text{for } \mathsf{v}_\alpha \in \mathscr{V}_h^{\mathrm{D}}. \end{cases} \tag{15}$$

We conclude this section by pointing out that the choice of the weights in (15) is not unique and different techniques, still providing second-order accuracy, may be envisaged. For example, in Ref. [10] an alternative method is proposed that does not rely on the Least Squares technique and ensures that the reconstruction weights are uniformly bounded in the open interval $(0, 1)$ for every h.

3.3. Source term contribution

The discretization of the right-hand side source term f in (1a) is taken into account by the N_{T}-sized vector $\boldsymbol{s} = \{\mathsf{s}_i\}$. The components of $\boldsymbol{s}$ are estimated by averaging the right-hand side analytical source function $f(\boldsymbol{x})$ on T_i. We have

$$\mathsf{s}_i = \frac{1}{|\mathsf{T}_i|}\int_{\mathsf{T}_i} f(\boldsymbol{x})\,\mathrm{d}V = \frac{1}{3}\sum_{\alpha\in v_i} f(\boldsymbol{x}_\alpha).$$

3.4. The finite volume linear system

The final system of linear equations is written as:

$$\boldsymbol{G}\mathbf{u}_h = \boldsymbol{s} + \boldsymbol{g}, \tag{16}$$

where $\boldsymbol{G} = \{\mathsf{G}_{ij}\}$ is the $N_\mathsf{T} \times N_\mathsf{T}$ stiffness matrix, $\mathbf{u}_h = \{u_i\} \in \mathbb{R}^{N_\mathsf{T}}$ is the vector of approximate cell averages, $\boldsymbol{s} = \{\mathsf{s}_i\} \in \mathbb{R}^{N_\mathsf{T}}$ is the source vector defined above, and $\mathbf{g} = \{\mathsf{g}_i\} \in \mathbb{R}^{N_\mathsf{T}}$ is the vector of boundary conditions. The matrix $\boldsymbol{G}$ and the vector $\boldsymbol{g}$ are built in the following assembly process. First, we express u_{ij} as a linear combination of vertex values by substituting (15) into (9). Then, we use these expressions in (10) to obtain the one-sided gradient $\mathscr{G}_{ij}(u_h)$ for every $\mathsf{T}_i \in \mathscr{T}_h$ and $j \in \sigma_i \cup \sigma_i'$. Note that the resulting stencil is formed by the gravity centers of all triangles sharing v_α and all triangles sharing v_β. The gradient $\mathscr{G}_{ij}(u_h)$ is, thus, defined in terms of cell averages of the elements of this stencil. In accordance with (11), we define the gradient $\mathscr{G}^\Diamond_{ij}(u_h)$ by averaging the two-one sided gradients $\mathscr{G}_{ij}(u_h)$ and $\mathscr{G}_{ji}(u_h)$. On the boundary, the gradient $\mathscr{G}^\Diamond_{ij}(u_h)$ coincides with the one-sided gradient $\mathscr{G}_{ij}(u_h)$. Finally, we reformulate the discrete flux balance, i.e. Eq. (3), as

$$\frac{1}{|\mathsf{T}_i|}\sum_{j\in\sigma_i\cup\sigma_i'} |\mathsf{e}_{ij}|\mathsf{G}_{ij}(u_h) = (\boldsymbol{G}\mathbf{u}_h - \boldsymbol{g})|_i = \mathsf{s}_i \quad \text{for } \mathsf{T}_i \in \mathscr{T}_h,$$

by employing the expressions of the internal and boundary edge gradients in the definition of the numerical flux (4). Further details of this assembly process as well as the formulas for $\boldsymbol{G} = \{\mathsf{G}_{ij}\}$ and $\boldsymbol{g} = \{\mathsf{g}_i\}$ are reported in Ref. [9]. The resulting non-symmetric linear system arising from this discretization is solved by using the routine *MA41* [4] from the *HSL* collection of FORTRAN linear algebraic solvers for sparse matrices.

4. Numerical experiments

The performance of the proposed finite volume method is evaluated by focusing on the numerical solution of the parametric model problem [5,20,21]:

$$-\frac{\partial^2 u}{\partial x^2} - \epsilon\frac{\partial^2 u}{\partial y^2} = f(x,y) \quad \text{for } (x,y) \in (0,1)\times(0,1), \tag{16a}$$

$$u(x,y) = g^\mathrm{D}(x,y) \quad \text{for } (x,y) \in \Gamma^\mathrm{D}, \tag{16b}$$

$$-n_x\frac{\partial u}{\partial u} - \epsilon n_y\frac{\partial u}{\partial y} = g^\mathrm{N}(x,y) \quad \text{for } (x,y) \in \Gamma^{\ \mathrm{N}}, \tag{16c}$$

where the *diffusion parameter* ϵ is a positive real number in the range $[10^{-6}, 1]$, and n_x ad n_y are the components of $\boldsymbol{n}$. The definition of the boundaries Γ^D and Γ^N depends on the particular test case that we will consider. Note that problem (16a)–(16c) is equivalent to (1a)–(1c) with an anisotropic conductivity tensor

$$K = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix}. \tag{17}$$

For a non-diagonal tensor, model problem (16a)–(16c) represents (1a)–(1c) on a rotated reference system aligned with the principal directions of anisotropy (the eigenvectors of K). The performance of the proposed method in solving (16a)–(16c) will be investigated by evaluating the rate of convergence on a sequence of successively refined meshes. We will consider three different grid families that will include uniform, quasi-uniform, and non-uniform triangulations, respectively. The mesh data structures are managed by P2MESH [8], a C^{++} public domain library designed for fast and efficient implementation of partial differential equation solvers. The base grids for the different families are shown in Fig. 4. The uniform grid family starts by a 4 × 4 partition of Ω into equally-shaped squares. Each square is then divided by its diagonals into four sub-triangles, as shown by Fig. 4 (left). The computational meshes at the next refinement levels are generated by halving the

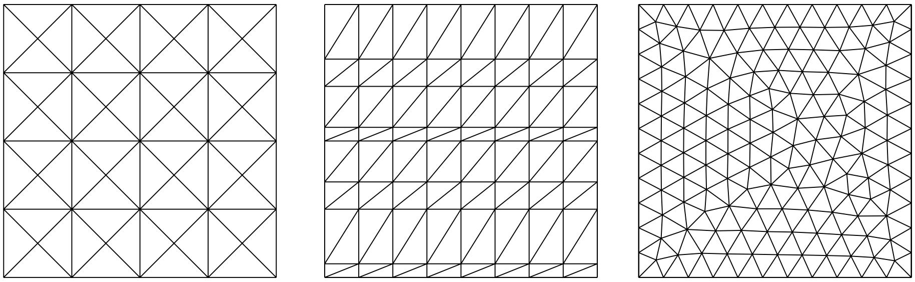

Fig. 4. Base computational grids for numerical experiments: $4\times\ 4\times 4$-sized uniform regular mesh (left), $8\times 8\times 2$ quasi-uniform mesh (middle), and strictly acute unstructured mesh (right).

partitioning in both coordinate directions and then subdividing the grid squares into four equally-shaped subtriangles. The quasi-uniform base grid is obtained as follows. The domain is first uniformly subdivided into 8×8 rectangles. The y-coordinate of the nodes along each horizontal line are changed to form non-uniform intervals along y having length proportional to the numbers $\{1,4,2,3,1,3,2,4\}$, respectively. The resulting triangulation (shown in Fig. 4, middle) is obtained by choosing the south-west north-east diagonal of each rectangle. Alternatively, we may consider the quasi-uniform base grid generated by changing the x-coordinates of the nodes accordingly to the previously mentioned rule. The non-uniform base grid is shown in Fig. 4 (right), and consists of 272 triangular cells, 430 edges, 159 vertices, and 44 boundary edges and vertices. All the angles are acute, thus, satisfying the mesh regularity assumptions. The refinement process for both the quasi-uniform and the non-uniform families progresses by connecting together the midpoints of the edges of each triangle. This ensures that the mesh size parameter is halved at every level and that the newly generated triangles maintain the same aspect ratio of the original ones.

We will also find it convenient to work with the equivalent stretched problem given by the y-scaled version of (16a)–(16c). This scaled version relies on the change of variables

$$\hat{x}=x \quad \text{and} \quad \hat{y}=y/\sqrt{\epsilon}. \tag{18}$$

Thus, we look for the the solution $\hat{u}(\hat{x},\hat{y})$ of the Poisson equation

$$-\frac{\partial^2\hat{u}}{\partial\hat{x}^2}-\frac{\partial^2\hat{u}}{\partial\hat{y}^2}=\hat{f}(\hat{x},\hat{y}) \quad \text{for } (\hat{x},\hat{y})\in\hat{\Omega}, \tag{19}$$

in the stretched domain $\hat{\Omega}=(0,1)\times(0,1/\sqrt{\epsilon})$ under suitably scaled Dirichlet and Neumann boundary conditions. The equivalence of the two formulations is guaranteed by taking $\hat{f}(\hat{x},\hat{y})=f(x,y)$ when the coordinate pairs $(\hat{x},\hat{y})$ and (x,y) are related by (18). This equivalence implies that $\hat{u}(\hat{x},\hat{y})=u(x,y)$.

4.1. Description of the test cases

According to the classification suggested in Ref. [21], we consider the following three test cases:

Case A: Dirichlet boundary data with $\Gamma^{\mathrm{N}}=\emptyset$ and

$$u=g^{\mathrm{D}} \quad \text{on } \Gamma^{\mathrm{D}}=\{(x,y)|x=0,1 \text{ or } y=0,1\}.$$

Case B: mixed Dirichlet–Neumann boundary data with

$$u=g^{\mathrm{D}} \quad \text{on } \Gamma^{\mathrm{D}}=\{(x,y)|x=0 \text{ or } y=0\},$$
$$-\boldsymbol{n}\cdot K\nabla u=g^{\mathrm{N}} \quad \text{on } \Gamma^{\mathrm{N}}=\{(x,y)|x=1 \text{ or } y=1\};$$

Case C: (nearly) pure Neumann boundary data with

$$u=g^{\mathrm{D}} \quad \text{on } \Gamma^{\mathrm{D}}=\{(x,y)|1-h\leqslant x\leqslant 1, y=1 \text{ or } x=1, 1-h\leqslant y\leqslant 1\},$$
$$-\boldsymbol{n}\cdot K\nabla u=g^{\mathrm{N}} \quad \text{on } \Gamma^{\mathrm{N}}=\Gamma\setminus\Gamma^{\mathrm{D}}.$$

Test Case C originally considers only the Neumann condition, i.e. $\Gamma^{\mathrm{D}} = \emptyset$, leading to a singular discrete Laplacian operator. The insertion of the modified Dirichlet condition on the two boundary edges indicated above is used to remove this singularity. Note that the length of these two edges tends to zero as $h \rightarrow 0$.

As pointed out in Ref. [21], a parametric error amplification, i.e. the *locking* effect, may arise for small values of ϵ when solving model problem (16a)–(16c). The error amplification may take place due to the asymptotic behavior of the solution in the limit for $\epsilon \rightarrow 0$. In particular, the existence of two possible asymptotic states, which are called *cool* and *hot* and have different main directions of diffusion, is conjectured [20,21]. The cool state is induced by the boundary data of type A and B, so that diffusion takes place along the direction of high conductivity not affected by the parameter ϵ. Conversely, the hot state is induced by the boundary data of type C. In this second case, diffusion occurs mainly along the direction of low conductivity, which is depressed by ϵ. As discussed in Ref. [21], this situation may be difficult to be treated numerically because of the locking effect.

We test the behavior of the present finite volume method by using the analytical solution [5]:

$$u(x,y) = \exp(-2\pi\sqrt{\epsilon}x)\sin(2\pi y). \tag{20}$$

The solution (20) is parametrically dependent on ϵ in the x-direction so that the corresponding forcing term is null, i.e. $f = 0$. The presence of ϵ in (20) is crucial because propagation of information along the low conductivity direction becomes dominant for very small values of the parameter. In this case, severe locking effects have been reported, particularly in the approximation of the solution of Case C [5,20,21].

Locking effects will be mainly investigated for the proposed finite volume method by reporting experimental convergence rates. These rates are measured by means of the relative solution error calculated using L^2-norms:

$$\mathscr{E}_h = \frac{\|u_h - \mathscr{A}_h(u)\|_h}{\|u\|_{L^2(\Omega)}} = \frac{\left(\sum_{\mathsf{T}_i \in \mathscr{T}_h} |\mathsf{T}_i| |u_i - \mathscr{A}_i(u)|^2\right)^{\frac{1}{2}}}{\|u\|_{L^2(\Omega)}}, \tag{21}$$

and the relative gradient error calculated using H^1-seminorms:

$$\mathscr{E}_{\mathsf{G},h} = \frac{\|\mathscr{G}_h(u_h) - \mathscr{A}_h(\nabla u)\|_h}{|u|_{H^1(\Omega)}} = \frac{\left(\sum_{\mathsf{T}_i \in \mathscr{T}_h} |\mathsf{T}_i| |\mathscr{G}_i - \mathscr{A}_i(\nabla u)|^2\right)^{\frac{1}{2}}}{|u|_{H^1(\Omega)}}. \tag{22}$$

The value of $\mathscr{E}_h$ is a measure of the approximation error of the cell averages of the solution, while $\mathscr{E}_{\mathsf{G},h}$ measures the error of the approximation of the solution gradient. The error $\mathscr{E}_{\mathsf{G},h}$ is defined in (22) by comparing the piecewise-constant finite volume gradient $\mathscr{G}_h(u_h) = \{\mathscr{G}_i\} \in P^0(\mathscr{T}_h)$ with the cell average of the solution gradient $\mathscr{A}_h(\nabla u) = (\mathscr{A}_h(\partial u/\partial x), \mathscr{A}_h(\partial u/\partial y)) \in P^0(\mathscr{T}_h)$. The value of $\mathscr{G}_i$ is calculated by taking the gradient of the plane interpolating the numerical solution at the three vertices of T_i. The rate of convergence is measured by comparing the errors at two consecutive mesh levels with mesh parameters h and $h/2$.

All the convergence rates are graphically presented in Figs. 5–8 as log–log plots of the error norms (L^2 on the left, H^1 on the right) versus the characteristic mesh size h. In each plot, the values of ϵ used to obtain the error curves are labeled by the symbols listed just above the figure captions. Since this finite volume method is formally second-order accurate, the convergence rates are expected to be $\mathscr{O}(h^2)$ for the L^2 norm and $\mathscr{O}(h)$ for the H^1 seminorm. The actual order of accuracy is reflected by the slopes of the experimental error curves, and can be approximately evaluated by comparison with the "theoretical" first- and second-order slopes represented in the bottom-left corner of each graph.

Two are the symptoms that characterize mesh locking effects for decreasing ϵ: convergence curves moving upward, denoting an increase of the error constant, and flat or nearly flat error curves, revealing a loss in the expected convergence rates. The behavior of the latter case actually describes a pre-asymptotic convergence regime [5]. Eventually, the expected rates would be observed if sufficiently small values of h were used, leading, however, to possibly unfeasible calculations.

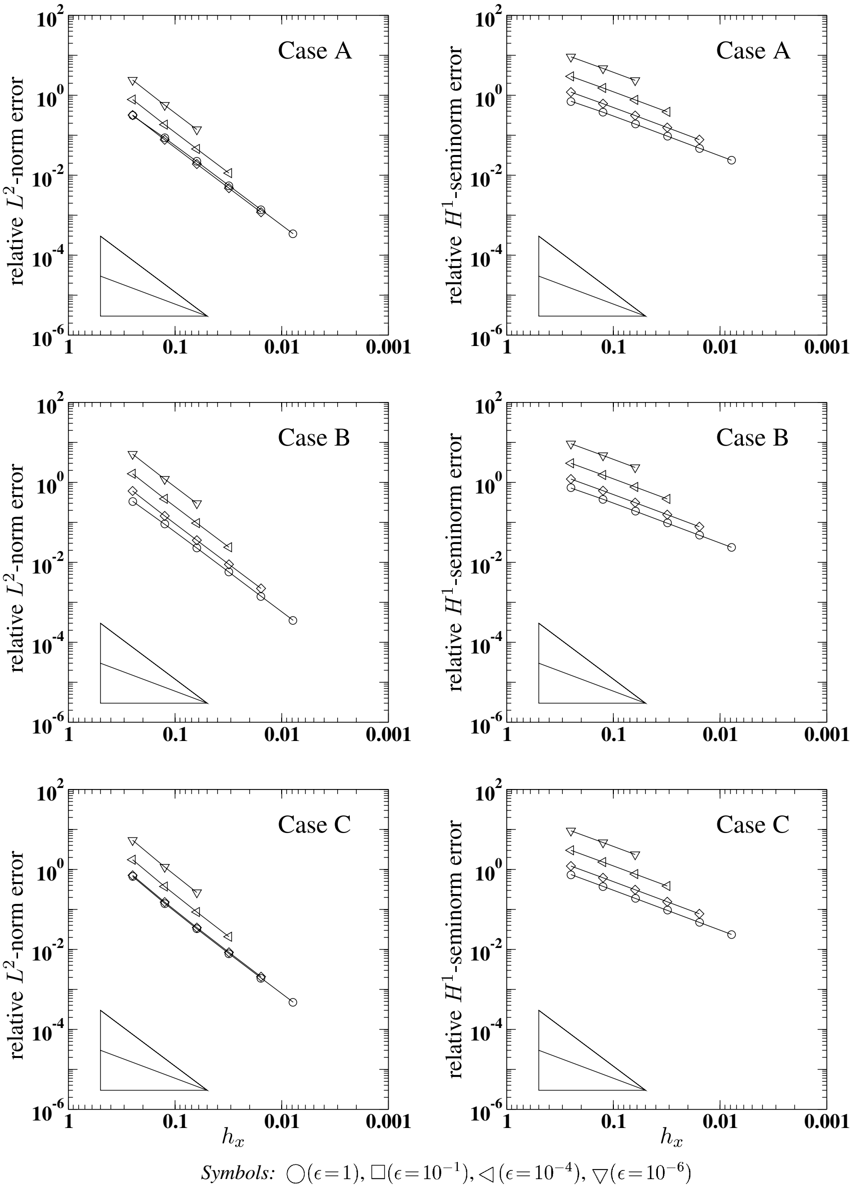

Fig. 5. Stretched domain; Test Cases A–C: error curves for the regular mesh partition of $\hat{\Omega} = (0,1) \times (0, 1/\sqrt{\epsilon})$: the finite volume method does not include the tangential gradient contributions to the numerical edge flux. Note that h_x represents the value of h along the (unstretched) x-direction.

4.2. Discussion of results

4.2.1. Results on stretched domain

In the isotropic case $K = \alpha I$, where I is the 2×2 identity matrix and α a real positive number, only the normal component of the gradient (10) intervenes in the finite volume discretization. Instead, in the anisotropic case, the additional contribution of the tangential term is essential to maintain the consistency of the scheme. The absence of this term would compromise the convergence of the sequence of approximate solutions when the meshes are refined. When K has the form (17), one may solve the equivalent isotropic problem on the stretched domain [33]. This approach simplifies the flux calculations because the tangential terms can be discarded. We study the performance of this strategy by employing transformation (18) and solving Eq. (19) by

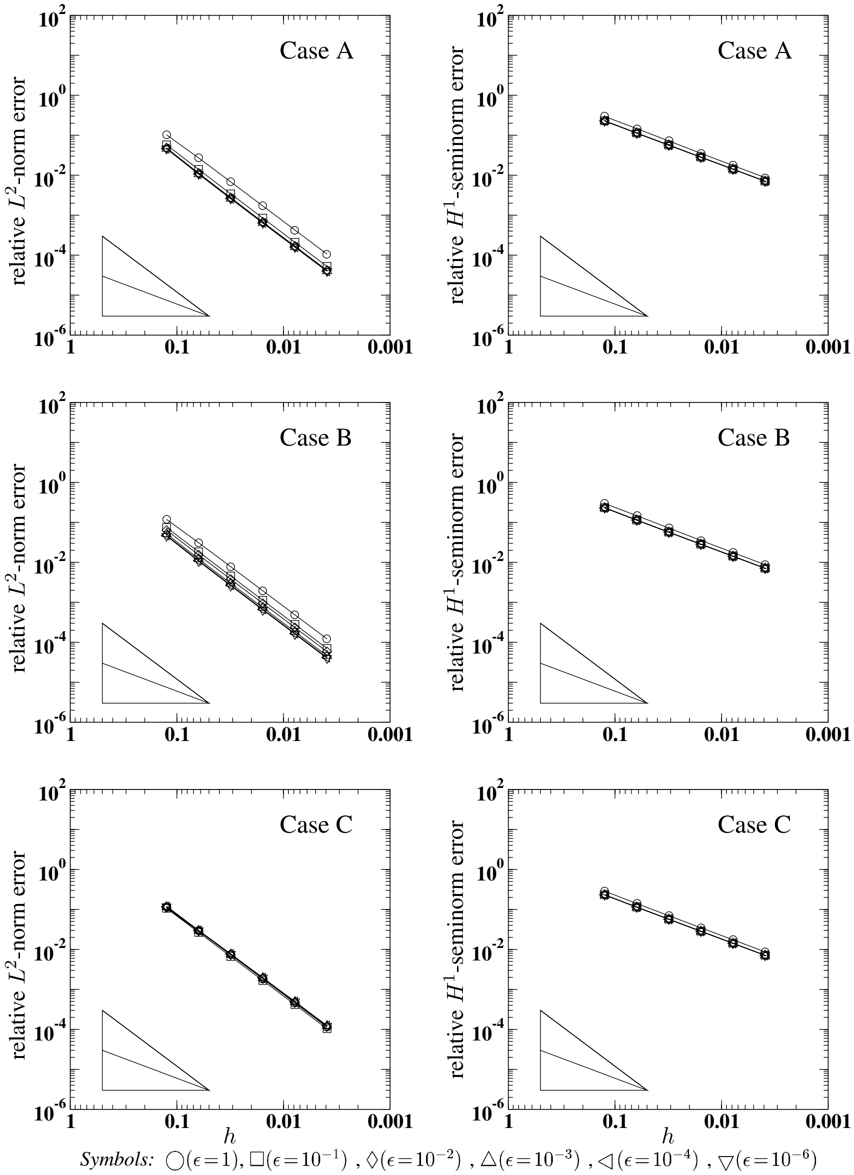

Fig. 6. Uniform mesh; Test Cases A–C: error curves relative to the complete finite volume scheme for the domain $\Omega = (0,1) \times (0,1)$.

uniformly triangulating the deformed domain. The problem with this approach is that the number of mesh control volumes covering the stretched computational domain significantly increases if cell aspect ratio must be maintained. Denote by $\sigma = 1/\sqrt{\epsilon}$ the *stretching factor*. It is easy to see that for the uniform and quasi-uniform mesh families the number of cells of the scaled mesh is exactly equal to σ times the number of cells of the mesh covering the original domain Ω. This relationship is only approximate for the non-uniform triangulation family and our chosen range of variation of ϵ corresponds to values of σ between 1 and 1000. To avoid complications on the resulting mesh, we do not experiment with non-integer stretching factors. For this reason, the plots of Fig. 5 do not show the results for $\epsilon = 10^{-1}$ and 10^{-3}. The most anisotropic cases that we consider in our simulations, i.e. $\epsilon = 10^{-4}$ and $\epsilon = 10^{-6}$, are rescaled into the isotropic form yielding meshes with a dimension from one-hundred to one-thousand times larger than that of the original meshes. Table 1 reports the value of some characteristic topological elements of the uniform base mesh for the experimented diffusion parameters ϵ and stretching factors σ. It is clear from this table that the size of the meshes involved in

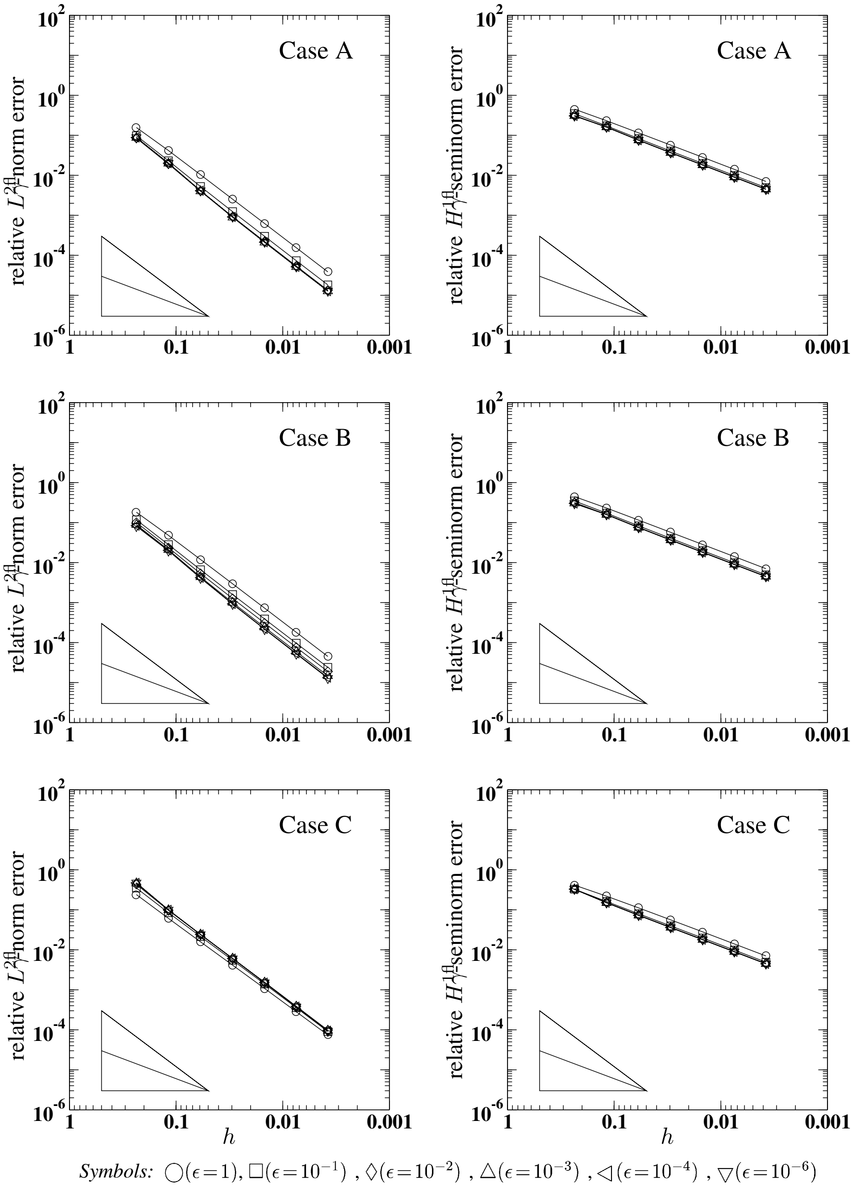

Fig. 7. Quasi-uniform mesh; Test Cases A–C: error curves relative to the domain $\Omega = (0,1) \times (0,1)$. The mesh is obtained by moving the y-coordinates of the nodes of the uniform subdivision of Ω to form non-uniform intervals along y having length proportional to the numbers $\{1,4,2,3,1,3,2,4\}$.

these calculations quickly becomes impractical. In particular, note that in the case $K = 10^{-6}$ the mesh obtained after the second refinement has 1,024,000 triangles, which is very close to the upper limit of the computers available to us. This is reflected by the convergence curves having fewer experimental points as $\epsilon \to 0$.

The results of the simulations are shown in Fig. 5. We maintain consistency of the plot scales throughout all the test problems so that all the graphs are directly comparable one to the other. Thus, we have plotted on the x-axis the value of h along the unstretched x-direction, i.e. h_x. The mesh size parameter of the stretched y-direction is $h_y = \sigma h_x$. The obtained convergence rates are very close to the expected values for both error norms (21) and (22). A small locking effect is perceived by the increasing error constant. This behavior may indicate that the stretching factor of the domain does not exhaust all the numerical difficulties connected with anisotropy. We should point out that this approach cannot be used in the case of spatially varying σ.

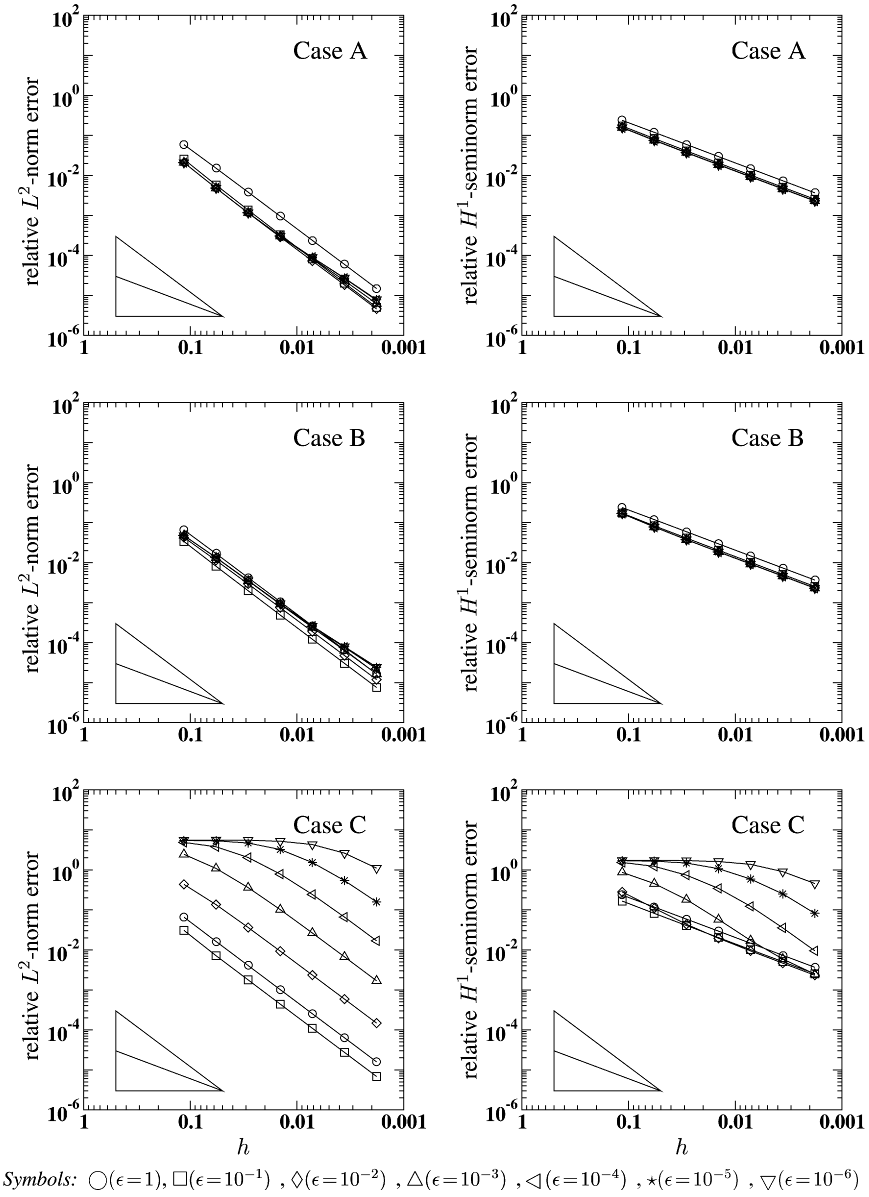

Fig. 8. Non-uniform mesh; Test Cases A–C: error curves for the non-uniform mesh partition of the domain $\Omega = (0,1) \times (0,1)$. Note the dramatic loss of accuracy due to the locking effect for Case C.

Table 1
Stretched domain

ϵ	σ	N_x	N_y	N_T	N_e	N_v	N_e^B
1	1	4	4	64	104	41	16
10^{-2}	10	4	40	640	1004	365	88
10^{-4}	100	4	400	6400	10,004	3605	808
10^{-6}	1000	4	4000	64,000	100,004	36,005	8008

Characteristics of the base meshes for different values of diffusion ratio ϵ and stretching factor σ; N_x and N_y are the number of partitions along the "X"- and "Y"-axis, respectively, N_T is the number of triangles of the mesh, N_e the number of edges, N_v the number of vertices, and N_e^B the number of boundary edges (which is equal to the number of boundary vertices).

Note that these results have been obtained on the uniform mesh. The results on the other grid families are similar and for this reason are not reported here. From these observations, we may conclude that this strategy, although feasible in some cases, is not computationally efficient. We prefer to solve the original unstretched problem and exploit the fact that our scheme naturally yields an accurate discretization of both normal and tangential gradients.

4.2.2. Results on uniform and quasi-uniform meshes

The numerical results on the uniform meshes are reported in Fig. 6. The scheme seems to work optimally in terms of convergence rates for all test cases and both L^2-norm and H^1-seminorm. A small locking effect is visible in Case C, where a slight increase in the error constant occurs. This yields a deterioration of the L^2-norm error by a factor of about 5 going from the isotropic to the most anisotropic case. The reason for the optimal convergence of the scheme may be attributable to the particular geometry of the mesh. As a matter of fact, independently of the refinement level, triangle edges parallel to the coordinate directions of anisotropy are always present. Furthermore, the triangles are all constructed from squares using the two main diagonals. The resulting symmetry of the triangulation is presumably responsible of error cancellations occurring within the discrete operator that could explain the behavior of the method.

The same numerical performance can be observed for the convergence on the quasi-uniform grid family generated by the base mesh of Fig. 4 (middle). The convergence curves are shown in Fig. 7. The only difference from the previous case is the initial error value, which is smaller for the quasi-uniform case by a factor of about 3 for the L^2-norm, consistently with the difference in the size parameters of the two base meshes. Very similar results (not shown) are obtained when the quasi-uniform base mesh is provided by changing the node coordinates of an initial $8 \times 8 \times 2$ regular partitioning along the x-direction instead of the y-direction, as already mentioned at the beginning of the section. In both situations, the mesh family maintains triangle edges that are aligned with the directions of anisotropy, but the symmetry of the diagonals is lost along one of the coordinate directions. For this reason, we believe that the quality of the numerical results is increased by mesh alignment reasons rather than by some effects of error cancellation due to grid symmetry. Finally, it is worth noting that these results are much more satisfactory than the behavior observed in the stretched domain (Fig. 5), where a stronger deterioration of the error constant due to locking is clearly visible. This shows that accuracy in the discretization of the tangential components of the cell-interface gradients is of fundamental importance for maintaining good performance.

4.2.3. Results on non-uniform meshes

The non-uniform mesh does not contain any special direction with respect to anisotropy, nor any peculiar symmetry. The base grid is obtained by a constrained Delaunay triangulation algorithm with the additional constraint that no angles greater than $\pi/2$ are present, so that the mesh regularity assumption is satisfied. Except for the usual difference in the starting error due to the changing mesh size parameter, the behavior of the scheme seems optimal only for Test Cases A and B. A strong locking effect is visible in Test Case C, where not only an error constant deterioration appears, but also a severe loss of convergence rate occurs as $\epsilon \to 10^{-6}$.

A tentative explanation of these results requires an in depth discussion of this test case. The differential problem is in fact a Neumann problem and, thus, the corresponding operator is singular, the solution being defined up to a constant. This is reflected by the discrete operator that is equipped with a non-trivial kernel. To remove the singularity, we impose Dirichlet conditions on the top-right corner of the domain, as previously described. Note that this is one of the commonest approaches for solving pure Neumann problems and, for two-dimensional finite elements, is known to have only little influence on the condition number of the system matrix [12]. Thus, we expect that the ill-conditioning of our linear system is mainly controlled by the condition number of the diffusion tensor, i.e. the ratio $1/\epsilon$. Since we are using a direct linear solver (Gaussian elimination with partial pivoting [4]), ill-conditioning can be observed by monitoring the behavior of the solution residual norm with respect to $1/\epsilon$. For this reason, in Table 2 we report the Euclidean norm of the residuals

$$\mathcal{R}_l = \boldsymbol{s} + \boldsymbol{g} - \boldsymbol{G}\mathbb{u}_{h_l}$$

Table 2
Non-uniform mesh

ϵ	$\mathscr{R}_0$	$\mathscr{R}_2$	$\mathscr{R}_4$	$\mathscr{R}_6$
1	8.43×10^{-13}	9.17×10^{-13}	7.09×10^{-12}	7.95×10^{-12}
10^{-2}	5.79×10^{-11}	7.85×10^{-11}	1.76×10^{-10}	5.78×10^{-10}
10^{-4}	3.83×10^{-9}	5.29×10^{-9}	5.66×10^{-9}	2.82×10^{-8}
10^{-6}	2.54×10^{-7}	3.46×10^{-7}	4.36×10^{-7}	4.06×10^{-7}

Test Case C: final residuals $\mathscr{R}_l$ for the solution of the linear system (16); h_0 is the mesh size parameter of the non-uniform base grid, and the refined meshes at refinement levels $l \geqslant 1$ are characterized by $h_{l+1} = h_l/2$.

calculated with $\mathbb{u}_{h_l}$, the solution of the linear algebraic system (16) approximating the "true" finite volume solution u_{h_l} at the lth level of refinement characterized by a mesh size parameter h_l.

Following the discussion reported in Refs. [5,20,21], we may postulate the existence of a threshold value for the mesh size parameter below which the scheme converges optimally. In this work, we experimentally investigate the relationship between ϵ and the threshold value of h, as hinted in their analysis of the asymptotic behavior of the cool and hot state solutions to the model problem (16a)–(16c) presented in Ref. [21]. This type of relationship suggests that the region of optimal convergence might be bounded by

$$h^2 \leqslant C\epsilon,$$

where C is a constant independent of h and ϵ. Experimental evaluation and "definition" of this constant may be problematic. For a confirmation of this behavior, we introduce a variant of the definition of *region of robustness* given in Ref. [5], that applies to the present scheme. More precisely, let

$$\mathsf{R}_0 = \{(h,\epsilon) | h \leqslant h_0, 10^{-6} \leqslant \epsilon \leqslant 1\}$$

be the space of parameters that contains the working values of ϵ and h considered in our test cases. We will try to identify the sub-region of R_0 where the method performs reasonably well. This last statement is made more precise by operatively defining the region where the convergence rates are greater than 1.9 as the region of optimal behavior, and the region where the convergence rates are between 1.6 and 1.9 as the region of sub-optimal behavior. If the convergence rate is less that 1.6 we consider our scheme to perform poorly, and when the convergence rate is above this value we say that the scheme is robust. In practice, this definition is very conservative, as in many applications convergence rates of about 1.5 are considered acceptable. In Fig. 9, we report the L^2-norm convergence rates that characterize the performance of the method for Test Case C

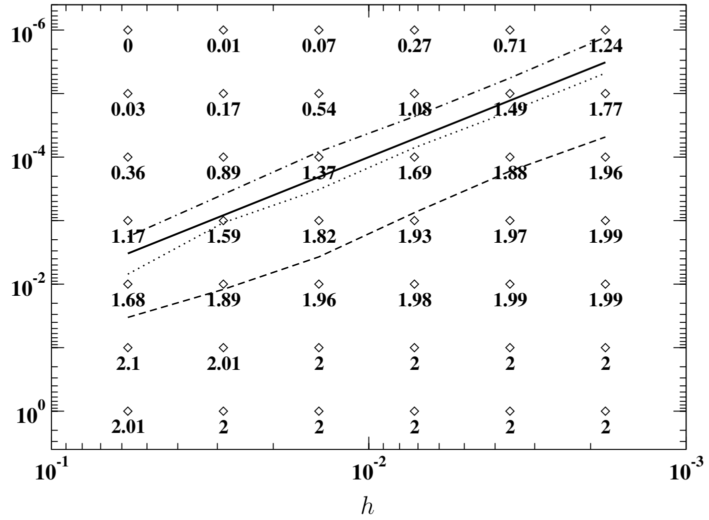

Fig. 9. Non-uniform mesh; Test Case C: experimental convergence rates within the region R_0 for calculations using the non-uniform base mesh; the continuous line represents the threshold curve $\epsilon = h^2$; the other contour lines are interpolating the rate values 1.9 (dashes), 1.6 (dots), and 1.3 (dot-dashes).

on non-uniform meshes. Every point of this figure represents a simulation run that was performed by using the corresponding pair of parameter values $(h,\epsilon) \in \mathsf{R}_0$. In this figure, we also draw the contour lines for rate values of 1.3, 1.6, and 1.9, and the "theoretical threshold" $\epsilon = h^2$. We should point it out that the correct location of the threshold curve on this plot depends upon the knowledge of the value of the constant C, this latter being dependent, in its turn, on the working definition of robustness. As the theoretical convergence analysis of the method at this stage of the research is not yet available, we plot this curve just to give an indication of the behavior that, we believe, is likely to be expected. However, it is an enlightening fact that the slopes of the numerical contours are very close to the slope of the theoretical curve, showing that the hypothesized behavior may be realistic.

5. Conclusions

We presented a second-order accurate cell-centered finite volume method, which is robust and reliable in the solution of strongly anisotropic diffusion. The approximation method is based on an edge-centered piecewise constant definition of the solution gradients that properly takes into account both the normal and the tangential components of the gradients in the formulation of the local flux balance equations. We reported the results of numerical experiments on a set of benchmark problems of varying degree of difficulty, that test the scheme performance under different anisotropy ratios and boundary conditions. The numerical results show that optimal or sub-optimal performances are obtained for all test cases. Only in the problem having exclusively Neumann boundary conditions the scheme revealed mesh locking phenomena for very small anisotropy ratios. These results show the importance of a proper approximation of the tangential component of the numerical gradients at cell interfaces. The presence of fully Neumann boundary conditions enhances the difficulties introduced by anisotropy. For this last test case, we reported an experimental study of the region of robustness of the proposed scheme. A quadratic relationship between anisotropy ratio and mesh size parameter was found. This shows that this finite volume scheme is robust and efficient in particular in the practical range of anisotropy ratio 10^{-3}–1 that is of interest in our applications.

Acknowledgments

The authors thank the two anonymous reviewers for their useful suggestions. The work of the second author has been partially supported by the Italian MIUR-PRIN project "Numerical Models for Multiphase Flow and Deformation in Porous Media".

References

[1] I. Aavatsmark, T. Barkve, O. Boe, T. Mannseth, Discretization on unstructured grids for inhomogeneous, anisotropic media. Part I: Derivation of the methods, SIAM J. Sci. Comput. 19 (5) (1998) 1700–1716, ISSN 1064-8275.

[2] I. Aavatsmark, T. Barkve, O. Boe, T. Mannseth, Discretization on unstructured grids for inhomogeneous, anisotropic media. Part II: Discussion and numerical results, SIAM J. Sci. Comput. 19 (5) (1998) 1717–1736, ISSN 1064-8275 (print), 1095-7197 (electronic).

[3] R.A. Adams, Sobolev Spaces, Academic Press, New York, 1975.

[4] P. Amestoy, C. Puglisi, An unsymmetrized LU multifrontal factorization, SIAM J. Matrix Anal. Appl. 24 (2) (2002) 553–569.

[5] I. Babuška, M. Suri, On locking and robustness in the finite element method, SIAM J. Numer. Anal. 29 (5) (1992) 1261–1293, ISSN 0036-1429.

[6] T.J. Barth, D.C. Jespersen, The design and application of upwind schemes on unstructured meshes, AIAA Paper 89-0366. AIAA, January 1989.

[7] J. Bear, Dynamics of Fluids in Porous Media, Dover, New York, 1972.

[8] E. Bertolazzi, G. Manzini, Algorithm 817 P2MESH: generic object-oriented interface between 2-D unstructured meshes and FEM/FVM-based PDE solvers, ACM Trans. Math. Software 28 (1) (2002) 101–132.

[9] E. Bertolazzi, G. Manzini, A cell-centered second-order accurate finite volume method for convection–diffusion problems on unstructured meshes, Math. Mod. Meth. Appl. Sci. 8 (2004) 1235–1260.

[10] E. Bertolazzi, G. Manzini, A second-order maximum principle preserving finite volume method for steady convection–diffusion problems, SIAM, J. Numer. Anal. 43 (5) (2006) 2172–2199.

[11] E. Bertolazzi, G. Manzini, A unified treatment of boundary conditions in least square-based finite volume methods, Comput. Math. Applications 49 (11–12) (2005) 1755–1765.

[12] P. Bochev, R.B. Lehoucq, On the finite element solution of the pure Neumann problem, SIAM Rev. 47 (1) (2005) 50–66.

[13] F. Brezzi, Stability of saddle-points in finite dimensions, in: J.F. Blowey, A.W. Craig, T. Shardlow (Eds.), Frontiers in Numerical Analysis, Universitext, Springer, Berlin, 2003, pp. 17–62.

[14] P.G. Ciarlet, The Finite Element Method for Elliptic Problems, North-Holland Publishing Company, Amsterdam, Holland, 1980.

[15] Y. Coudière, J.-P. Vila, P. Villedieu, Convergence rate of a finite volume scheme for a two dimensional convection–diffusion problem, Math. Model. Numer. Anal. 33 (3) (1999) 493–516.

[16] Y. Coudière, P. Villedieu, Convergence rate of a finite volume scheme for the linear convection–diffusion equation on locally refined meshes, M2AN Math. Model. Numer. Anal. 34 (6) (1999) 1123–1149, ISSN 0764-583X.

[17] K. Domelevo, P. Omnes, A finite volume method for the Laplace equation on almost arbitrary two-dimensional grids, M2AN 39 (6) (2005) 1203–1249.

[18] E. Eymard, T. Gallouët, R. Herbin, A finite volume scheme for anisotropic diffusion problem, Comptes Rendus Acad. Sci., Mathématiques 339 (4) (2004) 299–302.

[19] R. Eymard, T. Gallouët, R. Herbin, in: Ph. Ciarlet, J.L. Lions (Eds.), Handbook for Numerical Analysis, North-Holland, Amsterdam, 2000, pp. 715–1022 (Chapter: The finite volume method).

[20] V. Havu, An analysis of asymptotic consistency error in a parameter dependent model problem, Calcolo 40 (2) (2003) 121–130, ISSN 0008-0624.

[21] V. Havu, J. Pitkäranta, An analysis of finite element locking in a parameter dependent model problem, Numer. Math. 89 (4) (2001) 691–714, ISSN 0029-599X.

[22] F. Hermeline, A finite volume method for the approximation of diffusion operators on distorted meshes, J. Comput. Phys. 160 (2000) 481–499.

[23] J. Hyman, M. Shashkov, Solving diffusion equations with rough coefficients in rough grids, J. Comput. Phys. 129 (1997) 383–405.

[24] J. Hyman, M. Shashkov, S. Steinberg, The numerical solution of diffusion problems in strongly heterogeneous non-isotropic materials, J. Comput. Phys. 132 (1997) 130–148.

[25] P.A. Jayantha, I.W. Turner, On the use of surface interpolation techniques in generalised finite volume strategies for simulating transport in highly anisotropic porous media, J. Comput. Appl. Math. 152 (1–2) (2003) 199–216, ISSN 0377-0427.

[26] P.A. Jayantha, I.W. Turner, A second order finite volume technique for simulating transport in anisotropic media, Int. J. Numer. Methods Heat Fluid Flow 13 (1) (2003) 31–56.

[27] P.A. Jayantha, I.W. Turner, A second order control-volume finite-element least-squares strategy for simulating diffusion in strongly anisotropic media, J. Comput. Math. 23 (1) (2005) 1–16, ISSN 0254-9409.

[28] M. Křižek, L. Liu, Finite element approximation of a nonlinear heat conduction problem in anisotropic media, Comput. Methods Appl. Mech. Eng. 157 (1998) 387–397.

[29] G. Manzini, S. Ferraris, Mass-conservative finite volume methods on 2-D unstructured grids for the Richards' equation, Adv. Water Resour. 27 (12) (2004) 1199–1215.

[30] C. Ollivier-Gooch, M. Van Altena, A high-order-accurate unstructured mesh finite-volume scheme for the advection–diffusion equation, J. Comput. Phys. 181 (2) (2002) 729–752, ISSN 0021-9991.

[31] C.F. Ollivier-Gooch, Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction, J. Comput. Phys. 133 (1997) 6–17.

[32] A. Paster, Modeling the salt-water body in the northern Yarkon–Taninim aquifer, Israel. M.Sc. Thesis, Tel Aviv University, 2004.

[33] M. Putti, C. Cordes, Finite element approximation of the diffusion operator on tetrahedra, SIAM, J. Sci. Comput. 19 (4) (1998) 1154–1168.

[34] A. Quarteroni, A. Valli, Numerical Approximation of Partial Differential Equations, Springer, Berlin, 1994, ISBN 3-540-57111-6.