Computers and Mathematics with Applications 68 (2014) 1229–1249

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

New cell–vertex reconstruction for finite volume scheme: Application to the convection–diffusion–reaction equation

Ricardo Costa[a], Stéphane Clain[a,b], Gaspar J. Machado[a,*]

[a] Centre of Mathematics, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
[b] Institut de Mathématiques de Toulouse, Université Paul Sabatier, 31062 Toulouse, France

ARTICLE INFO

Article history:
Received 26 March 2014
Received in revised form 9 July 2014
Accepted 17 August 2014
Available online 15 September 2014

Keywords:
Convection–diffusion–reaction
Finite volume method
Cell–vertex interpolation
Second-order accuracy

ABSTRACT

The design of efficient, simple, and easy to code, second-order finite volume methods is an important challenge to solve practical problems in physics and in engineering where complex and very accurate techniques are not required. We propose an extension of the original Frink's approach based on a cell-to-vertex interpolation to compute vertex values with neighbour cell values. We also design a specific scheme which enables to use whatever collocation point we want in the cells to overcome the mass centre point restrictive choice. The method is proposed for two- and three-dimensional geometries and a second-order extension time-discretization is given for time-dependent equation. A large number of numerical simulations are carried out to highlight the performance of the new method.

1. Introduction

The finite volume method for the linear convection–diffusion–reaction equation is an important building-block to solve more complex models such as the Navier–Stokes equations and nonlinear coupling problems. In the two last decades, major efforts have been made to design very high-order schemes up to sixth-order [1–6] to increase the accuracy enabling to compute very good approximations even with coarse meshes. Nevertheless, second-order schemes are still attractive since they are quite simple, easy to code and, many practical problems in physics and engineering do not require such complex and accurate methods.

The convection–diffusion–reaction equation contains the three typical operators which appear in mathematical modelling and each one has its own specificities. The diffusion and convection terms are computed via flux contributions but the convection requires some upwind technique to achieve stability while the diffusive term provides a symmetric contribution. On the contrary, the reactive term is not affected by the divergence operator and cannot be treated through a flux on the interfaces. This term turns to be crucial when dealing with time discretization for evolution problems and motivate the effort to design a specific discretization.

Second-order finite volume schemes are very well developed and there exist several groups/families of techniques to provide that order of convergence. We shall mention for example the mimetic method [7] and the finite volume scheme based on primal and dual meshes (DDFV or DMGR schemes) [8–10]. We refer to the book of Eymard, Gallouët, and Herbin [11] and the references therein for a general overview. A popular class of second-order finite volume schemes is based on vertex reconstructions using pointwise approximations in the cells associated to a specific point location (usually the centroid). Then combining cell and vertex values, gradient approximations are evaluated to compute the diffusive flux and a second-order approximation for the convective flux is considered on the interfaces (the so-called diamond scheme). There exist several

[*] Corresponding author.
E-mail addresses: pg24046@alunos.uminho.pt (R. Costa), clain@math.uminho.pt (S. Clain), gjm@math.uminho.pt (G.J. Machado).

http://dx.doi.org/10.1016/j.camwa.2014.08.015
0898-1221/© 2014 Elsevier Ltd. All rights reserved.

studies about the way to provide the nodal values [12–17] where few of them treat the Neumann boundary condition [18–20]. Evaluation of the vertex values ψ_n from the cell values ϕ_i is based on the simple linear combination $\psi_n = \sum_{i\in\mu(n)} \beta_{ni}\phi_i$, where $\mu(n)$ is the index set of the cells in touch with vertex n. First-order approximation using the inverse-distance between the vertex and the centroids of the cells as the weighting factors has been first introduced by Frink [21,22] in 1991 and 1992. To the authors' knowledge, the first document which deals with second-order reconstruction at the vertex is the paper of Holmes and Connel [23] but it is important to mention that in this paper it is considered vertex to vertex reconstructions. The first paper which considers second-order cell to vertex reconstruction is the study of Rausch, Batina, and Yang [17] in 1991 through a minimization of a functional based on the coefficients β_{ni} with a second-order constraint: the reconstruction is exact for affine functions. The method has been adapted for the three-dimensional context by Frink [14] in 1994 (see also the paper of Jawahar and Kamath [15] for a review in 2000 on cell to vertex reconstructions). At last and very recently in 2013, Chandrashekar and Garg [13] use the Frink approach introducing some weights in the minimizer functional to determine the coefficients β_{ni} under the affine constraints. We also would like to mention the work of Sheng and Yuan [24] where they do not use any minimization technique and the coefficients are deduced from geometrical arguments.

All the previous methods are based on a direct evaluation of the coefficients β_{ni} but Coudière, Vila, and Villedieu proposed in [25] an alternative way to compute the values on the vertices by minimizing a functional based on coefficients a, b, and c of the affine reconstruction $a + bx + cy$ under some restrictions to preserve the second-order. Bertolazzi and Manzini proposed several extensions [18,20], in particular, they introduce weighted functionals to include the boundary conditions. To sum-up, there exist two main techniques to determine the values at the vertices from the values associated to the cells: the Frink–Rausch–Batina–Yang way based on the minimization of β_{ni} and the Coudière–Vila–Villedieu–Bertolazzi–Manzini way based on the minimization of the polynomial coefficients. From our point of view, the latter one presents a major drawback since we cannot control the positivity of the coefficients β_{ni}. Indeed, one checks easily that the maximum principle is achieved if coefficients β_{ni} correspond to a convex combination. This property is mandatory when dealing with physical quantities such as concentration, mass fraction, or density.

The present paper is dedicated to a new method to perform the cell-to-vertex reconstruction.

- We adopt the first approach (Frink-like method) where the non-negativity of the coefficients is guaranteed. To this end, we consider a more general functional to minimize introducing the notion of target combination, *i.e.* a set of coefficients θ_{ni} corresponding to a convex combination but that do not, *a priori*, provide a second-order approximation. The method is detailed for the two-dimensional situation with an extension for three-dimensional geometries.
- The second issue we address is that our formulation considers finite volume schemes where the cell-value is associated to a point which may be not the mass centre still preserving the second-order. The Dirichlet and Neumann conditions discretization is performed in the same way to provide an effective second-order scheme.
- The third question we tackle concerns the reactive term where we propose a new scheme which preserves the second-order even if the cell-value location is not the mass centre. Such a discretization turns to be essential when dealing with parabolic problems with an implicit time discretization we shall detail in the context of the Crank–Nicholson scheme for time dependent cases.
- We present a large set of numerical tests in 2D and 3D geometries for steady and time dependent regimes in order to prove the accuracy and the stability of the technique.

We would like to mention that all the problems we shall consider assume that the solution is smooth and we intentionally avoid the problem of the limiting/stabilization procedure one has to carry out for irregular solutions. A usual way consists to apply the MUSCL strategy [18,12] or to introduce an artificial diffusion to stabilize the scheme [16]. Such a question will be tackled in future studies with the help of the MOOD method [2].

The paper is organized as follows. After the introduction of the generic finite volume scheme for the steady-state convection–diffusion–reaction problem in Section 2, we address the cell to vertex reconstruction issue in Section 3. Section 4 is dedicated to the polynomial reconstructions while Section 5 is devoted to the details of the finite volume discretization. Section 6 is devoted to the extension to time-dependent problems. The numerical tests are given in Section 7 and the study ends with a conclusion and some perspectives.

2. Finite volume for the steady-state convection–diffusion–reaction problem

Let Ω be an open bounded polygonal domain of $\mathbb{R}^2$ with boundary Γ. We seek function $\phi \equiv \phi(x, y)$, solution of the steady-state convection–diffusion–reaction equation

$$\nabla \cdot (V\phi - \kappa\nabla\phi) + r\phi = f \quad \text{in } \Omega, \tag{1}$$

where the diffusive coefficient $\kappa \equiv \kappa(x, y)$, the velocity $V = (u, v) \equiv (u(x, y), v(x, y))$, the reaction coefficient $r \equiv r(x, y)$, and the source term $f = f(x, y)$ are all regular functions on $\overline{\Omega}$ and the diffusive coefficient satisfies $\kappa(x, y) \geqslant \kappa_0 > 0$. Boundary Γ is partitioned into three subsets Γ_{D}, Γ_{T}, and Γ_{P} in order to prescribe different types of boundary conditions, namely:

- Dirichlet: $\phi = \phi_{\mathrm{D}}$, on Γ_{D};
- total Neumann: $V \cdot n\phi - \kappa\nabla\phi \cdot n = g_{\mathrm{T}}$, on Γ_{T};
- partial Neumann: $-\kappa\nabla\phi \cdot n = g_{\mathrm{P}}$, on Γ_{P},

where $\phi_{\mathrm{D}} \equiv \phi_{\mathrm{D}}(x, y)$, $g_{\mathrm{T}} \equiv g_{\mathrm{T}}(x, y)$, and $g_{\mathrm{P}} \equiv g_{\mathrm{P}}(x, y)$ are given regular functions, and n denotes the unit normal to Γ outward to Ω.

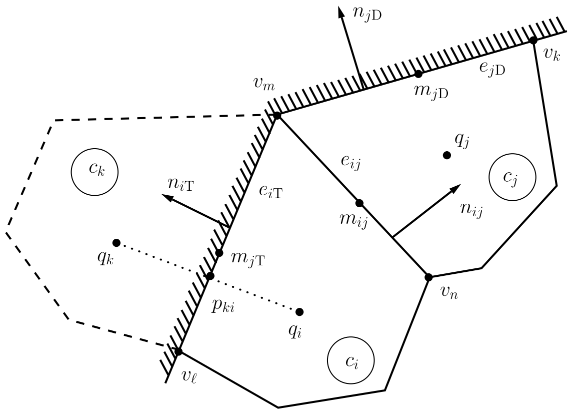

Fig. 1. Mesh notation considering real cells (solid lines) and ghost cells (dashed lines).

2.1. Mesh

To design the numerical scheme, we denote by $\mathcal{T}$ a mesh of Ω consisting of I non-overlapping convex polygonal cells $c_i,\ i = 1, \dots, I$, and N vertices $v_n,\ n = 1, \dots, N$. We adopt the following conventions (see Fig. 1) we detail hereafter:

- for any cell c_i, ∂c_i represents its boundary and $|c_i|$ its area; we denote by q_i, b_i, and m_i a generic point, the centroid, and the mass centre of c_i, respectively;
- two cells c_i and c_j share a common edge e_{ij} whose length is $|e_{ij}|$ and the midpoint is m_{ij}; n_{ij} is the unit normal vector to e_{ij} outward to c_i, *i.e.* $n_{ij} = -n_{ji}$; if an edge of c_i belongs to the boundary Γ, we replace the index j by D, P, or T if e_{ij} belongs to Γ_{D}, Γ_{P}, or Γ_{T}, respectively;
- for any cell c_i we associate the index set $\nu(i) \subset \{1, \dots, I\} \cup \{\mathrm{D}, \mathrm{T}, \mathrm{P}\}$ such that $j \in \nu(i)$ if e_{ij} is a common edge of c_i and c_j or with the boundary Γ_j if $j = \{\mathrm{D}, \mathrm{T}, \mathrm{P}\}$;
- if c_i shares an edge with Γ_{P} or Γ_{T}, c_k is the symmetric ghost cell and q_k is the symmetric point of q_i with respect to the common edge with $k \in \{I + 1, \dots, I + K\}$ (K being the number of ghost cells).

Remark 2.1. If v_n is a vertex at the intersection of Γ_{D} and Γ_{P}, or Γ_{D} and Γ_{T}, we assume that v_n belongs to Γ_{D} and will be treated as a Dirichlet point.

2.2. Generic finite volume scheme

To provide the finite volume scheme, Eq. (1) is integrated over each cell of the mesh

$$\int_{c_i} \nabla \cdot (V\phi - \kappa \nabla \phi)\, \mathrm{d}A + \int_{c_i} r\phi \, \mathrm{d}A = \int_{c_i} f \, \mathrm{d}A,$$

and applying the divergence theorem we get

$$\sum_{j \in \nu(i)} \frac{|e_{ij}|}{|c_i|} \frac{1}{|e_{ij}|} \int_{e_{ij}} \left(V \cdot n_{ij}\phi - \kappa \nabla \phi \cdot n_{ij} \right) \mathrm{d}s + \frac{1}{|c_i|} \int_{c_i} r\phi \, \mathrm{d}A - \frac{1}{|c_i|} \int_{c_i} f \, \mathrm{d}A = 0. \tag{2}$$

Let ϕ_i be an approximation of ϕ at q_i and let us gather all the approximations in vector $\Phi = (\phi_i)_{i=1,\dots,I}$. We then substitute the exact scheme (2) by a second-order accuracy numerical scheme, with respect to the mesh parameter h, and depending on vector Φ,

$$\sum_{j \in \nu(i)} \frac{|e_{ij}|}{|c_i|} \mathcal{F}_{ij}(\Phi) + \mathcal{R}_i(\Phi) - f_i(\Phi) = \mathcal{O}(h^2), \tag{3}$$

where $\mathcal{F}_{ij}$ is an approximation of the convective and diffusive fluxes through the edge e_{ij}, $\mathcal{R}_i$ is an approximation of the mean value of the reactive part over c_i, and f_i is an approximation of the mean value of f over c_i.

3. The interpolation method at the vertices

We want to design a second-order finite volume scheme using both approximations at the reference cell points and at the vertices where the unknowns are only located in the cells. Approximations of the gradient in the cells and at the edges are then obtained with the Green–Gauss theorem (diamond scheme [25,11,15,12] or diamond-path scheme [26]). It results that an accurate evaluation of the vertex values with respect to the cell values must be implemented. Let $\psi_n,\ n = 1, \dots, N$ be an approximation of ϕ at the vertex v_n and let us gather all these approximations in vector $\Psi = (\psi_n)_{n=1,\dots,N}$. The goal of this section is to design a procedure to compute Ψ from Φ.

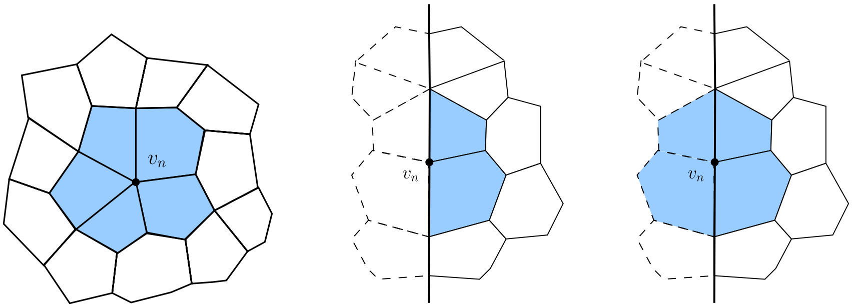

Fig. 2. Stencils (coloured cells) for the vertices in different situations: $\mu(n) = \gamma(n)$ for an inner vertex (left), $\mu(n)$ for a vertex on Γ_{T} or Γ_{P} (centre), and $\gamma(n)$ for a vertex on Γ_{T} or Γ_{P} (right). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.1. The stencils and the data

For each vertex v_n, we associate two stencils: the first one $\mu(n) \subset \{1, \dots, I\}$ consisting of the neighbour cells of the mesh (real cells) and the second one $\gamma(n) \subset \{1, \dots, I + K\}$ composed both of real cells and ghost cells if necessary. In practice, for an inner vertex (a vertex which does not belong to the boundary), we choose $\gamma(n) = \mu(n)$ and $\ell \in \mu(n)$ if v_n is a vertex of cell c_ℓ (Fig. 2-left). For a vertex on Γ_{D}, the two index sets are not necessary since the value is prescribed by the Dirichlet condition. For a vertex on Γ_{T} or Γ_{P} (Neumann conditions), $\mu(n)$ consists of the neighbour real cells c_ℓ (Fig. 2-centre) while $\gamma(n)$ also includes the ghost cells which share the vertex v_n (Fig. 2-right).

Vector $\Phi = (\phi_i)_{i=1,\dots,I}$ corresponds to the values associated to the real cells but approximations at the ghost cells must be evaluated in order to use index set γ. To this end, we seek a combination of the values associated to the real cells to provide an approximation at the reference cell points of the ghost cells. Let c_k, $k = I + 1, \dots, I + K$, be a ghost cell and c_i its symmetric real cell. We denote by p_{ki} the midpoint of the line segment defined by q_i and q_k and by $e_{i\mathrm{T}}$ and $e_{i\mathrm{P}}$ the edges that belong to the total Neumann boundary and the partial Neumann boundary, respectively (see Fig. 1). To determine ϕ_k as an approximation of ϕ at q_k, two situations arise:

- if c_k shares an edge with Γ_{P}, then we use the approximation

$$g_{\mathrm{P}}(p_{ki}) = -\kappa(p_{ki}) \frac{\phi_k - \phi_i}{|q_k q_i|} \tag{4}$$

from which we deduce ϕ_k;

- if c_k shares an edge with Γ_{T}, then we use one of the following approximations:

$$g_{\mathrm{T}}(p_{ki}) = V(p_{ki}) \cdot n_{i\mathrm{T}} \frac{\phi_k + \phi_i}{2} - \kappa(p_{ki}) \frac{\phi_k - \phi_i}{|q_k q_i|}, \tag{5a}$$

$$g_{\mathrm{T}}(p_{ki}) = [V(p_{ki}) \cdot n_{i\mathrm{T}}]^+ \phi_i + [V(p_{ki}) \cdot n_{i\mathrm{T}}]^- \phi_k - \kappa(p_{ki}) \frac{\phi_k - \phi_i}{|q_k q_i|}, \tag{5b}$$

with the notations $[v]^+ = \max(0, v)$ and $[v]^- = \min(0, v)$, $v \in \mathbb{R}$, from which we compute ϕ_k.

Remark 3.1. Point p_{ki} is the midpoint between q_i (the real cell) and q_k (the ghost cell) hence formula (4) is a second-order centred scheme since we evaluate g_{P} and κ at p_{ki}. The same property holds for (5a) and (5b). All the numerical experiences with Neumann condition confirm that all these approximations are second-order.

3.2. The interpolation method

We shall present the interpolation procedure when considering the stencils $\mu(n)$ since the technique for the stencils $\gamma(n)$ is identical. Let us consider the vertex v_n and its associated stencil $\mu(n)$. We then define ψ_n as

$$\psi_n = \sum_{i \in \mu(n)} \beta_{ni} \phi_i, \tag{6}$$

where we gather in vector $B^n = (\beta_{ni})_{i \in \mu(n)}$ the coefficients of the linear combination of the cell data.

3.2.1. $\mathbb{P}_0$ interpolation

Following Frink [21], we choose B^n such that

$$\sum_{i \in \mu(n)} \beta_{ni} = 1, \tag{7}$$

which implies that constant functions satisfy Eq. (6). For instance, the inverse cell–vertex distance is proposed by [22] but other choices such as

$$\beta_{ni} = \frac{|c_i|}{\sum\limits_{j\in\mu(n)} |c_j|} \tag{8}$$

can also be considered. Unfortunately, it is well-known that constraint (7) only provides an effective first-order scheme (see [20, p. 19]).

3.2.2. $\mathbb{P}_1$ *interpolation*

As observed in [25,17], one has to reinforce the restriction such that Eq. (6) is consistent for first-degree polynomials. Let us define the operators

$$f_1(B^n) = \sum_{i\in\mu(n)} \beta_{ni},$$

$$f_2(B^n) = \sum_{i\in\mu(n)} \beta_{ni}x_{ni},$$

$$f_3(B^n) = \sum_{i\in\mu(n)} \beta_{ni}y_{ni},$$

where $(x_{ni}, y_{ni}) = (q_{ix} - v_{nx}, q_{iy} - v_{ny}) = v_n q_i$. We aim to choose vector B^n such that

$$f_1(B^n) = 1, \tag{9a}$$

$$f_2(B^n) = 0, \tag{9b}$$

$$f_3(B^n) = 0. \tag{9c}$$

The linear system (9a)–(9c) has a unique solution in very particular situations ($\#\mu(n) = 3$ for instance, where # denotes the cardinal of the set) hence one has to design a strategy to determine a solution in the general case. We propose here a new method based on the minimization of a functional.

Let $\theta_n = (\theta_{ni})_{i\in\mu(n)}$ be a set of target coefficients such that $\sum_{i\in\mu(n)} \theta_{ni} = 1$. We define the quadratic functional

$$E(B^n) = \frac{1}{2} \sum_{i\in\mu(n)} \omega_{ni}(\beta_{ni} - \theta_{ni})^2 \tag{10}$$

where ω_{ni} are positive weights. We seek the unique vector B^n which minimizes the quadratic functional (10) under constraints (9a)–(9c). Using the classical minimization method with the Lagrange multipliers, the problem turns to find vectors $\Lambda^n = (\lambda_{n1}, \lambda_{n2}, \lambda_{n3})$ and B^n such that

$$\nabla E(B^n) + \lambda_{n1}\nabla f_1(B^n) + \lambda_{n2}\nabla f_2(B^n) + \lambda_{n3}\nabla f_3(B^n) = 0.$$

We deduce that

$$\beta_{ni} = \theta_{ni} - \frac{1}{\omega_{ni}}(\lambda_{n1} + \lambda_{n2}x_{ni} + \lambda_{n3}y_{ni}), \quad i \in \mu(n). \tag{11}$$

Taking into consideration (9a)–(9c) and (11), we obtain the linear system

$$\lambda_{n1} \sum_{i\in\mu(n)} \frac{1}{\omega_{ni}} + \lambda_{n2} \sum_{i\in\mu(n)} \frac{x_{ni}}{\omega_{ni}} + \lambda_{n3} \sum_{i\in\mu(n)} \frac{y_{ni}}{\omega_{ni}} = 0,$$

$$\lambda_{n1} \sum_{i\in\mu(n)} \frac{x_{ni}}{\omega_{ni}} + \lambda_{n2} \sum_{i\in\mu(n)} \frac{x_{ni}^2}{\omega_{ni}} + \lambda_{n3} \sum_{i\in\mu(n)} \frac{x_{ni}y_{ni}}{\omega_{ni}} = \sum_{i\in\mu(n)} \theta_{ni}x_{ni},$$

$$\lambda_{n1} \sum_{i\in\mu(n)} \frac{y_{ni}}{\omega_{ni}} + \lambda_{n2} \sum_{i\in\mu(n)} \frac{x_{ni}y_{ni}}{\omega_{ni}} + \lambda_{n3} \sum_{i\in\mu(n)} \frac{y_{ni}^2}{\omega_{ni}} = \sum_{i\in\mu(n)} \theta_{ni}y_{ni}.$$

This linear system has a unique solution Λ^n from which we determine the coefficients of vector B^n with Eq. (11). Several sets of target coefficients θ_{ni} and weights ω_{ni} will be proposed. For instance, a simple example may be

$$\theta_{ni} = \frac{|c_i|}{\sum\limits_{j\in\mu(n)} |c_j|}, \qquad \omega_{ni} = 1, \quad i \in \mu(n). \tag{12}$$

Notice that the Rausch, Batina, and Yang [17] method corresponds to the case $\theta_{ni} = \frac{1}{\#\mu(n)}$ and $\omega_{ni} = 1$.

Remark 3.2. In [20, pp. 19–20], the authors also propose a method to determine the coefficients such that (9a)–(9c) hold, based on an initial set of coefficients (target coefficients) which only satisfy constraint (7), but the method does not guarantee the non-negativity of the new coefficients.

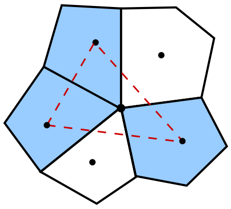

Fig. 3. A combination of three cells of $\mu(n)$ are gathered in $\mu_k(n)$ and corresponds to a set of positive barycentric coordinates.

3.3. *Positivity principle preserving*

Positivity preserving is a mandatory property when dealing with physical quantities such as concentration or density. We aim that the reconstructions guarantee such property and one easily deduces that a necessary and sufficient condition is the non-negativity of coefficients $\beta_{ni} \geqslant 0,\ i \in \mu(n)$, for vertex v_n. The $\mathbb{P}_0$ interpolation preserves this property when, for instance, Eq. (8) is considered but the $\mathbb{P}_1$ interpolation does not guarantee the positivity principle preservation. To overcome this problem, for a given vertex v_n we consider all the subsets of three elements $\mu_k(n) \subset \mu(n),\ k = 1, \dots, \binom{\#\mu(n)}{3}$, and the associated vector $\alpha_{nk} \in \mathbb{R}^3$ formed by the barycentric coordinates of vertex v_n with respect to points $q_i,\ i \in \mu_k(n)$. For each set of barycentric coordinates α_{nk}, we denote by m_{nk} the smallest coordinate. There exists at least one configuration $\mu_K(n)$ which maximizes m_{nk}, *i.e.* $m_{nK} \geqslant m_{nk},\ k = 1, \dots, \binom{\#\mu(n)}{3}$. We then set β_{ni} equal to the corresponding barycentric coordinate if $i \in \mu_K(n)$ and zero elsewhere. One can easily check that the positivity preserving principle is achieved if v_n is a point strictly inside the triangle with vertices q_i, $i \in \mu_K(n)$ as shown in Fig. 3. Notice that this technique does not require any minimization functional like (10).

Remark 3.3. The extension for the three-dimensional case is straightforward. Based on relation (6), we adapt the restrictions (9a)–(9c) by adding one more condition with respect to the z component. Applying the same minimization with constraints procedure, we then get a very similar relation as (11) involving four Lagrange multipliers. The target coefficients can be given by relation (12) where, this time, $|c_i|$ is the volume of the cell.

Remark 3.4. The non-negativity of the coefficients avoids the clipping technique where negative coefficients are set to zero (see [27, p. 12]) which may dramatically reduce the scheme order.

4. Polynomial reconstructions

We now build the local polynomial approximations of the underlying solution involving the two vectors Φ and Ψ. As a first stage, we associate a stencil of vertices for each cell and edge: S_i stands for the index set of the vertices of cell c_i while S_{ij} is the index set of vertices of edge $e_{ij},\ j \in \{1, \dots, I, \mathrm{D}\}$. The second stage, detailed in the following subsections, consists in defining the polynomial reconstructions based on the entries of vectors Φ and Ψ associated to the appropriated stencils.

4.1. *Polynomial reconstructions in the cells*

For each cell c_i we define the conservative polynomial (see [1]) of degree 1 by

$$\boldsymbol{\phi}_i(X) \equiv \boldsymbol{\phi}_i(x, y) = \phi_i + \mathcal{C}_{i,x}\,(x - q_{ix}) + \mathcal{C}_{i,y}\left(y - q_{iy}\right),$$

where $\mathcal{C}_{i,x}$ and $\mathcal{C}_{i,y}$ are the coefficients to be determined. For a given stencil S_i we consider the quadratic functional

$$\widetilde{E}_i(\mathcal{C}_{i,x}, \mathcal{C}_{i,y}) = \sum_{n \in S_i} \left(\boldsymbol{\phi}_i(v_n) - \psi_n\right)^2 \tag{13}$$

and we denote by $\widetilde{\mathcal{C}}_{i,x}$ and $\widetilde{\mathcal{C}}_{i,y}$ the unique coefficients that minimize the quadratic functional (13) with $\widetilde{\boldsymbol{\phi}}_i \equiv \widetilde{\boldsymbol{\phi}}_i(x, y)$ the associated polynomial function (see Fig. 4, left) which corresponds to the best approximation in the least squares sense of the data of the stencil.

4.2. *Polynomial reconstructions at inner edges and Dirichlet boundary*

For a given inner edge e_{ij}, we define the polynomial of degree 1 by

$$\boldsymbol{\phi}_{ij}(x, y) = \phi_i + \mathcal{C}_{ij,x}(x - q_{ix}) + \mathcal{C}_{ij,y}(y - q_{iy}),$$

where $\mathcal{C}_{ij,x}$ and $\mathcal{C}_{ij,y}$ are the coefficients to be determined. We denote by $\widetilde{\mathcal{C}}_{ij,x}$ and $\widetilde{\mathcal{C}}_{ij,y}$ the unique coefficients such that the associated polynomial function $\widetilde{\boldsymbol{\phi}}_{ij} \equiv \widetilde{\boldsymbol{\phi}}_{ij}(x, y)$ interpolates ϕ_i, defined at q_i, and ψ_n, defined at v_n, $n \in S_{ij}$ (see Fig. 4, centre).

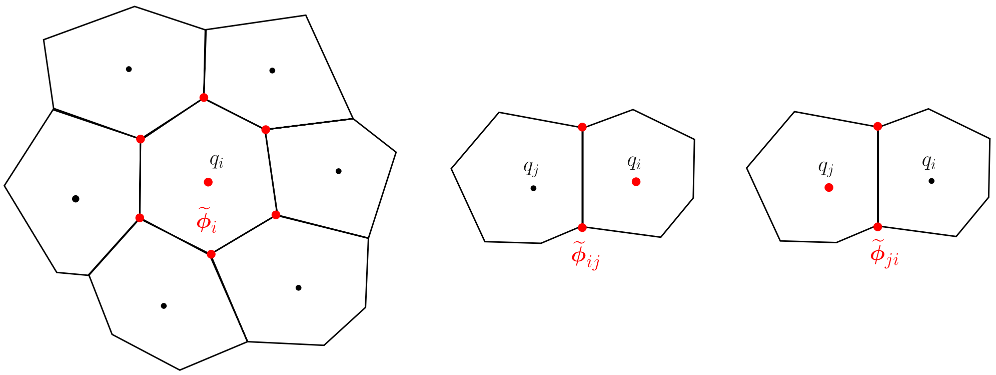

Fig. 4. First-degree polynomials for cells (left) and for edges (centre and right). The red dots correspond to the values used in the polynomials reconstructions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Notice that we also define the polynomial $\widetilde{\boldsymbol{\phi}}_{ji} \equiv \widetilde{\boldsymbol{\phi}}_{ji}(x, y)$ using the reference cell point q_j and the associated value ϕ_j (see Fig. 4, right).

For an edge $e_{i\mathrm{D}} \subset \Gamma_{\mathrm{D}}$, we proceed in the same way to provide polynomial $\widetilde{\boldsymbol{\phi}}_{i\mathrm{D}} \equiv \widetilde{\boldsymbol{\phi}}_{i\mathrm{D}}(x, y)$.

Remark 4.1. For the three-dimensional case, we shall consider polynomial function as

$$\boldsymbol{\phi}_{ij}(x, y, z) = \phi_i + \mathcal{C}_{ij,x}(x - q_{ix}) + \mathcal{C}_{ij,y}(y - q_{iy}) + \mathcal{C}_{ij,z}(z - q_{iz}).$$

For tetrahedron cells, the faces are triangles hence the polynomial reconstructions are straightforward as in the bidimensional case. For more complex faces (with more vertices) one has to introduce a least-square approximation to provide the polynomial approximation.

5. Second-order scheme

In Section 3 we compute Ψ from Φ (and eventually from $\phi_k,\ k = I + 1, \dots, I + K$) and in Section 4 we provide the local polynomial reconstructions based on vectors Ψ and Φ. We are now ready to return to the generic finite volume scheme (3) and compute the numerical approximations $\mathcal{F}_{ij}$, $\mathcal{R}_i$, and f_i.

5.1. Convective and diffusive terms

Having all the polynomial reconstructions in hand, we detail the numerical fluxes $\mathcal{F}_{ij}$ with respect to the interfaces of the cells of the mesh.

First, for an inner edge e_{ij} we define the polynomials

$$\check{\boldsymbol{\phi}}_{ij} = \check{\boldsymbol{\phi}}_{ji} = \sigma_{ij}\widetilde{\boldsymbol{\phi}}_{ij} + \sigma_{ji}\widetilde{\boldsymbol{\phi}}_{ji}. \tag{14}$$

We choose $\sigma_{ij} = \frac{|c_i|}{|c_i|+|c_j|}$ and $\sigma_{ji} = \frac{|c_j|}{|c_i|+|c_j|}$ for the sake of simplicity but situations with discontinuous diffusion coefficients for instance may require other expressions.

Remark 5.1. Let us denote by $[q_i, v_n, q_j, v_s]$ the diamond cell associated to edge e_{ij}. There exists a unique discrete gradient $\nabla\bar{\phi}_{ij}$ defined by $\nabla\bar{\phi}_{ij}\cdot q_iq_j = \phi_j - \phi_i$ and $\nabla\bar{\phi}_{ij}\cdot v_nv_s = \psi_n - \psi_s$. It is not possible to obtain such a gradient with the combination (14) except for the case when the intersection of segment $[q_i, q_j]$ with e_{ij} exactly corresponds to the midpoint.

Remark 5.2. Consider the pure diffusive problem with Dirichlet boundary condition. In [12] (see also [25]), the authors show that the choice of σ_{ij} does not guarantee the maximum principle property for the discrete solution and one has to combine the left and right one-sided gradients in a nonlinear way, namely σ_{ij} depends on the function ϕ [12, p. 2183]. Nevertheless, all the numerical tests have been performed with the constant σ_{ij} and no maximum principle violation has been reported.

We have four situations:

- for an inner edge e_{ij}, the numerical flux at the midpoint m_{ij} writes

$$\mathcal{F}_{ij} = \left[V(m_{ij})\cdot n_{ij}\right]^{+}\widetilde{\boldsymbol{\phi}}_i(m_{ij}) + \left[V(m_{ij})\cdot n_{ij}\right]^{-}\widetilde{\boldsymbol{\phi}}_j(m_{ij}) - \kappa(m_{ij})\nabla\check{\boldsymbol{\phi}}_{ij}(m_{ij})\cdot n_{ij};$$

- for a Dirichlet boundary edge $e_{i\mathrm{D}}$, the numerical flux at the midpoint $m_{i\mathrm{D}}$ writes

$$\mathcal{F}_{i\mathrm{D}} = [V(m_{i\mathrm{D}})\cdot n_{i\mathrm{D}}]^{+}\widetilde{\boldsymbol{\phi}}_i(m_{i\mathrm{D}}) + [V(m_{i\mathrm{D}})\cdot n_{i\mathrm{D}}]^{-}\phi_{\mathrm{D}}(m_{i\mathrm{D}}) - \kappa(m_{i\mathrm{D}})\nabla\widetilde{\boldsymbol{\phi}}_{i\mathrm{D}}(m_{i\mathrm{D}})\cdot n_{i\mathrm{D}};$$

- for a partial Neumann boundary edge $e_{i\mathrm{P}}$, the numerical flux at the midpoint $m_{i\mathrm{P}}$ writes
$$\mathcal{F}_{i\mathrm{P}} = V(m_{i\mathrm{P}}) \cdot n_{i\mathrm{P}} \widetilde{\boldsymbol{\phi}}_i(m_{i\mathrm{P}}) + g_{\mathrm{P}}(m_{i\mathrm{P}});$$
- for a total Neumann boundary edge $e_{i\mathrm{T}}$, the numerical flux at the midpoint $m_{i\mathrm{T}}$ writes
$$\mathcal{F}_{i\mathrm{T}} = g_{\mathrm{T}}(m_{i\mathrm{T}}).$$

5.2. Reactive and source terms

Second-order approximations $\mathcal{R}_i$ and f_i of the reactive and source terms require an extra effort because these expressions do not derive from a flux. To derive second-order approximations, we split the cell c_i into $\#\nu(i)$ triangular subcells denoted by $c_{ij}, j \in \nu(i)$, associated to edge e_{ij} (see Fig. 5).

The reactive part $\mathcal{R}_i$ will be evaluated using one of the following approximations:

$$\mathcal{R}_i = r(q_i)\phi_i, \tag{15}$$

$$\mathcal{R}_i = \frac{1}{|c_i|}\left[\sum_{j\in\nu(i)} \frac{|c_{ij}|}{3}\left(\sum_{n\in S_{ij}} r(v_n)\psi_n + r(q_i)\phi_i\right)\right]. \tag{16}$$

If c_i is a triangle, Eq. (16) turns into

$$\mathcal{R}_i = \frac{1}{3}\sum_{n\in S_i} r(v_n)\psi_n.$$

The source term is evaluated with

$$f_i = \frac{1}{|c_i|}\left[\sum_{j\in\nu(i)} \frac{|c_{ij}|}{3}\left(\sum_{n\in S_{ij}} f(v_n) + f(q_i)\right)\right],$$

which can be simplified if c_i is a triangular cell to

$$f_i = \frac{1}{3}\sum_{n\in S_i} f(v_n).$$

Remark 5.3. For the particular $q_i = m_i$, one can use the very simple quadrature rule $f_i = f(q_i)$ to derive a second-order approximation of the mean value of f over the cell c_i. It is important to note that this simple quadrature rule does not provide a second-order approximation for a generic point q_i.

Remark 5.4. One has to slightly modify formula (16) for the three-dimensional case setting

$$\mathcal{R}_i = \frac{1}{|c_i|}\left[\sum_{j\in\nu(i)} \frac{|c_{ij}|}{4}\left(\sum_{n\in S_{ij}} r(v_n)\psi_n + r(q_i)\phi_i\right)\right], \tag{17}$$

where S_{ij} is the set of the vertices of the face f_{ij}. If c_i is a tetrahedron, Eq. (17) turns into

$$\mathcal{R}_i = \frac{1}{4}\sum_{n\in S_i} r(v_n)\psi_n.$$

5.3. Residual scheme

Since $\mathcal{F}_{ij}$, $\mathcal{R}_i$, and f_i linearly depend on vector Φ, we define the affine operator $\Phi \to \mathcal{G}_i(\Phi)$ for each cell c_i, $i = 1, \dots, I$, as

$$\mathcal{G}_i(\Phi) = \sum_{j\in\nu(i)} \frac{|e_{ij}|}{|c_i|}\mathcal{F}_{ij}(\Phi) + \mathcal{R}_i(\Phi) - f_i(\Phi),$$

which corresponds to the finite volume scheme cast (3) in the residual form.

Gathering all the components $\mathcal{G}_i(\Phi)$ of the residual in vector $\mathcal{G}(\Phi)$, we obtain an affine operator from $\mathbb{R}^I$ into $\mathbb{R}^I$ such that vector $\Phi^\star$, solution of the problem $\mathcal{G}(\Phi) = 0_I$, provides a constant piecewise approximation of the problem. We obtain a matrix-free scheme and the affine problem is solved by applying a GMRES procedure as explained in [1].

6. Extension to time-dependent problems

We dedicate this section to the adaptation of the presented cell–vertex scheme to solve time-dependent convection–diffusion–reaction problems. As for the steady-state problems, let us denote the spatial domain by Ω, an open bounded

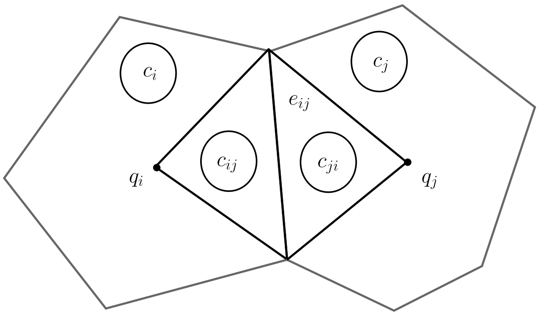

Fig. 5. Representation of two subcells, c_{ij} and c_{ji}.

polygonal domain of $\mathbb{R}^2$ with boundary $\Gamma = \Gamma_{\mathrm{D}} \cup \Gamma_{\mathrm{T}} \cup \Gamma_{\mathrm{P}}$, and let $\left[t^0, t^{\mathrm{f}}\right]$ be the time domain. We denote by $\phi \equiv \phi(x, y, t)$ the solution of the time-dependent convection–diffusion–reaction equation, given by

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (V\phi - \kappa \nabla \phi) + r\phi = f, \quad \text{in } \Omega \times \left[t^0, t^{\mathrm{f}}\right], \tag{18}$$

where the velocity $V = (u, v) \equiv (u(x, y, t), v(x, y, t))$, the diffusion coefficient $\kappa \equiv \kappa(x, y, t)$, the reaction coefficient $r \equiv r(x, y, t)$, and the source term $f \equiv f(x, y, t)$ are given regular functions. Eq. (18) is equipped with $\phi_{\mathrm{D}} \equiv \phi_{\mathrm{D}}(x, y, t)$ on Γ_{D}, $g_{\mathrm{P}} \equiv g_{\mathrm{P}}(x, y, t)$ on Γ_{P}, and $g_{\mathrm{T}} \equiv g_{\mathrm{T}}(x, y, t)$ on Γ_{T} with respect to Dirichlet, partial Neumann, and total Neumann time-dependent boundary conditions, respectively (see Section 2). An initial condition is also required,

$$\phi(\cdot, \cdot, t^0) \equiv \phi^0, \quad \text{in } \Omega,$$

where $\phi^0 \equiv \phi^0(x, y)$ is a given function.

We introduce vector $\Phi(t) = (\phi_i(t))_{i=1,\dots,I}$ where $\phi_i(t)$ is an approximation of $\phi(q_{ix}, q_{iy}, t)$ while $\psi_n(t),\ n = 1, \dots, N$, is computed from $\Phi(t)$ using the cell to vertex interpolation time-parameterized method explained in Section 3. To design the finite volume scheme for time-dependent problems, we apply the method of lines starting by integrating Eq. (18) over cell c_i to provide the semi-discretization in space

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{M}_i(t, \Phi(t)) + \sum_{j\in\nu(i)} \frac{|e_{ij}|}{|c_i|}\mathcal{F}_{ij}(t, \Phi(t)) + \mathcal{R}_i(t, \Phi(t)) - f_i(t) = \mathcal{O}(h^2), \tag{19}$$

where $\mathcal{M}_i(t, \Phi(t))$ is an approximation of the mean value of ϕ over cell c_i at time t, and $\mathcal{F}_{ij}(t, \Phi(t))$, $\mathcal{R}_i(t, \Phi(t))$, and $f_i(t)$ are computed as explained in Section 5 considering a fixed time value t.

We compute $\mathcal{M}_i(t, \Phi(t))$ using one of the following approximations:

$$\mathcal{M}_i = \phi_i(t), \tag{20}$$

$$\mathcal{M}_i = \frac{1}{|c_i|}\left[\sum_{j\in\nu(i)} \frac{|c_{ij}|}{3}\left(\sum_{n\in S_{ij}} \psi_n(t) + \phi_i(t)\right)\right], \tag{21}$$

following the notations already introduced in Section 5.

Note that $\mathcal{M}_i(t, \Phi(t))$ corresponds to the reactive part when the reaction coefficient is $r = 1$ and relations (20) and (21) derive from (15) and (16), respectively, (see Section 5.2). In the same way, Eq. (21) can be substituted by

$$\mathcal{M}_i = \frac{1}{3}\sum_{n\in S_i} \psi_n(t)$$

when cell c_i is a triangle.

We rewrite Eq. (19) in the residual form considering an extension of operator $\mathcal{G}_i(\Phi)$ to a time-dependent situation,

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{M}_i(t, \Phi(t)) + \mathcal{G}_i(t, \Phi(t)) = \mathcal{O}(h^2), \tag{22}$$

where

$$\mathcal{G}_i(t, \Phi(t)) = \sum_{j\in\nu(i)} \frac{|e_{ij}|}{|c_i|}\mathcal{F}_{ij}(\Phi(t)) + \mathcal{R}_i(\Phi(t)) - f_i(t).$$

To proceed with the time discretization, let K be a positive integer and consider the time subdivision $t^k = k\Delta T,\ k = 0, \dots, K$, where $\Delta t = \frac{t^f}{K}$. We now apply the Crank–Nicholson method to Eq. (22) which leads to

$$\frac{\mathcal{M}_i(\Phi^{k+1}) - \mathcal{M}_i(\Phi^k)}{\Delta t} + \frac{\mathcal{G}_i(t^{k+1}, \Phi^{k+1}) + \mathcal{G}_i(t^k, \Phi^k)}{2} = \mathcal{O}(h^2), \tag{23}$$

where vector $\Phi^k,\ k = 1, \dots, K$, gathers the approximations $\phi_i^k,\ i = 1, \dots, I$ of ϕ at the reference cell point q_i at time t^k. Note that we substitute vector $\Phi(t)$ with Φ^k in Eqs. (20) and (21) to compute $\mathcal{M}_i(\Phi^k)$.

Since $\mathcal{G}_i$ and M_i linearly depend on vector Φ^k, we define the affine operator $(\Phi^k, \Phi^{k+1}) \to \mathcal{H}_i(\Phi^k, \Phi^{k+1}),\ k = 0, \dots, K-1$, for each cell $c_i,\ i = 1, \dots, I$, and for each time interval $\left[t^n, t^{n+1}\right]$ as

$$\mathcal{H}_i(t^k, t^{k+1}, \Phi^k, \Phi^{k+1}) = \frac{\mathcal{M}_i(\Phi^{k+1}) - \mathcal{M}_i(\Phi^k)}{\Delta t} + \frac{\mathcal{G}_i(t^{k+1}, \Phi^{k+1}) + \mathcal{G}_i(t^k, \Phi^k)}{2},$$

which corresponds to the finite volume scheme cast (23) in the residual form. Gathering all i components $\mathcal{H}_i(t^k, t^{k+1}, \Phi^k, \Phi^{k+1})$ provides the residual operator $\mathcal{H}(t^k, t^{k+1}, \Phi^k, \Phi^{k+1})$.

We consider the affine operator from $\mathbb{R}^I$ into $\mathbb{R}^I$ such that for t^k, t^{k+1}, and vector Φ^k we associate the operator $\Phi \to \mathcal{H}(t^k, t^{k+1}, \Phi^k, \Phi)$, and we seek vector $\Phi = \Phi^{k+1}$ solution of $\mathcal{H}(t^k, t^{k+1}, \Phi^k, \Phi) = 0_I$. For t^0 vector Φ^0 is given by the initial condition ϕ^0 setting $\phi_i^0 = \phi^0(q_{ix}, q_{iy}),\ i = 1, \dots, I$.

We would like to highlight that for each time step we obtain a matrix-free scheme and the affine problem is solved by applying a GMRES procedure as explained in [1].

7. Numerical tests

In this section we present several tests to quantitatively and qualitatively assess the robustness and accuracy of the proposed method. In order to test the implementation of the method we check situations for which we manufacture a solution. All the simulations have been carried out on the academic domain $\Omega =]0, 1[^2$ (expect when explicitly mentioned).

Given the numerical approximation $\Phi^\star = (\phi_i^\star)_{i=1,\dots,I}$ of ϕ at the reference cell points of a given mesh $\mathcal{T}$, we evaluate the error using the L^1- and the L^∞-norms, given by

$$E_1(\mathcal{T}) = \sum_{i=1}^{I} |\phi_i^\star - \phi(q_i)|\,|c_i| \quad \text{and} \quad E_\infty(\mathcal{T}) = \max_{i=1}^{I} |\phi_i^\star - \phi(q_i)|, \tag{24}$$

respectively. We also compute the convergence rate of the L^γ-norm error, $\gamma = 1, \infty$, obtained with two different meshes $\mathcal{T}_1$ and $\mathcal{T}_2$ consisting of I_1 and I_2 cells, respectively, as

$$O_\gamma(\mathcal{T}_1, \mathcal{T}_2) = 2\frac{|\log(E_\gamma(\mathcal{T}_1)/E_\gamma(\mathcal{T}_2))|}{|\log(I_1/I_2)|}.$$

The notation $\mathbb{P}_0$ and $\mathbb{P}_1$ means that we employ the $\mathbb{P}_0$ and the $\mathbb{P}_1$ interpolations, respectively, to compute Ψ (see Sections 3.2.1 and 3.2.2). In all tests (except when explicitly mentioned), we assume that $q_\ell = m_\ell,\ \ell = 1, \dots, I+K$, the weights are $\omega_{ni} = 1$, and the target coefficients θ_{ni} are given by (12).

7.1. Convection–diffusion problem with Dirichlet condition

We consider the simple linear convection–diffusion problems with homogeneous Dirichlet condition (see Fig. 6, left). For the sake of simplicity, we assume a constant diffusion coefficient $\kappa = 1$ and a constant velocity $V = (u, v)$ where we set $V = (1, 2)$ for the low Péclet number case (diffusive regime) and $V = (100, 100)$ for the large Péclet number case (convective regime). The exact solution is given by $\phi(x, y) = C\alpha(x)\beta(y)$ with $C \in \mathbb{R}$ and

$$\alpha(x) = \frac{1}{u}\left(x - \frac{\exp(ux) - 1}{\exp(u) - 1}\right), \qquad \beta(y) = \frac{1}{v}\left(y - \frac{\exp(vy) - 1}{\exp(v) - 1}\right),$$

while the right-hand side is $f(x, y) = C(\alpha(x) + \beta(y))$ with the Dirichlet boundary condition $\phi_D(x, y) = 0$ on $\Gamma_D = \Gamma$. In order to normalize the exact solution such that the maximum of ϕ in $\bar{\Omega}$ is 1, we take $C = 65$ for the low Péclet case and $C = 11\,236$ for the large Péclet case (see Fig. 6, centre and right).

To perform the simulations and compute the convergence rates, we consider successive finer unstructured meshes considering triangular Delaunay meshes and deformed quadrilateral meshes. For the low Péclet number case the meshes are regular (see Fig. 7, left and centre) while we use locally refined triangular Delaunay meshes for the simulations with the large Péclet number case in order to suit well the boundary layer induced by the homogeneous Dirichlet condition (see Fig. 7, right). Scheme robustness and accuracy assessment with deformed meshes are important to check the method capacity to handle complex meshes still preserving second-order convergence rates. These meshes were computed from structured meshes where we randomly moved each inner vertex with a specific deformation factor (see [1]). In the present experience, we choose a 30% deformation.

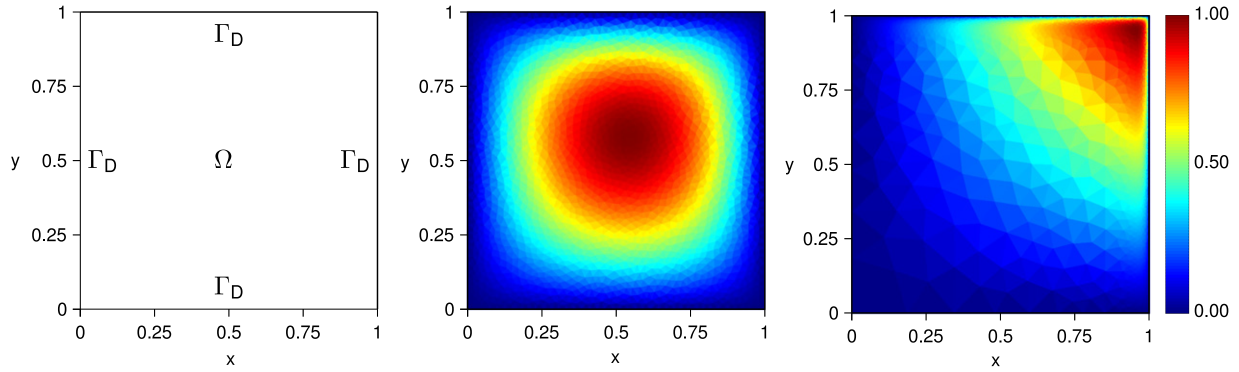

Fig. 6. Domain and boundary partitions (left) and exact solutions for the low Péclet number case (centre) and for the large Péclet number case (right) with fine meshes.

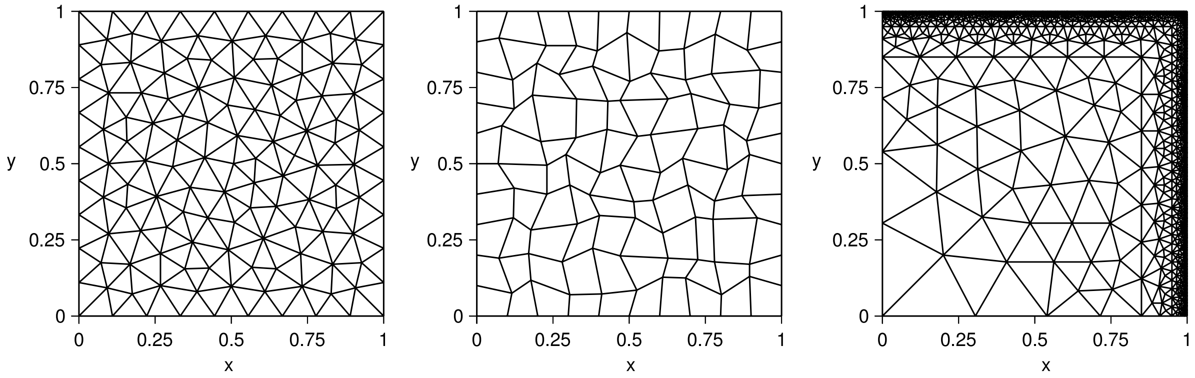

Fig. 7. Uniform triangular Delaunay mesh (left), deformed quadrilateral mesh (centre), and locally refined triangular Delaunay mesh (right).

Table 1
Errors and convergence rates for the low Péclet case using regular triangular Delaunay meshes.

I	$\mathbb{P}_0$				$\mathbb{P}_1$			
	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
944	1.06E−03	–	3.43E−03	–	6.70E−04	–	3.01E−03	–
4 038	7.55E−04	0.46	2.17E−03	0.63	1.44E−04	2.12	7.89E−04	1.84
16 374	7.48E−04	0.01	1.98E−03	0.13	3.61E−05	1.98	2.18E−04	1.84
44 050	7.05E−04	0.12	1.79E−03	0.21	1.30E−05	2.06	7.94E−05	2.04

Table 2
Errors and convergence rates for the low Péclet case using deformed meshes.

I	$\mathbb{P}_0$				$\mathbb{P}_1$			
	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
1 600	2.63E−02	–	6.55E−02	–	3.22E−04	–	1.99E−03	–
6 400	2.72E−02	–	6.68E−02	–	8.44E−05	1.93	5.50E−04	1.85
16 900	2.75E−02	–	6.66E−02	–	3.16E−05	2.03	2.23E−04	1.86
44 100	2.79E−02	–	6.74E−02	–	1.23E−05	1.97	8.80E−05	1.94

7.1.1. *Low Péclet number with m_i as the reference cell point*

We report in Table 1 (Delaunay mesh) and Table 2 (deformed mesh) the errors and the convergence rates for the low Péclet number case using the $\mathbb{P}_0$ and the $\mathbb{P}_1$ interpolations.

For the Delaunay meshes, an effective second-order accuracy is achieved with the $\mathbb{P}_1$ interpolation whereas the $\mathbb{P}_0$ method does not converge. When dealing with deformed meshes, the second-order convergence with the $\mathbb{P}_1$ reconstruction is preserved but no convergence with $\mathbb{P}_0$ is achieved. These two examples demonstrate that the $\mathbb{P}_1$ technique guarantees the second-order convergence and definitively disqualify the $\mathbb{P}_0$ interpolation.

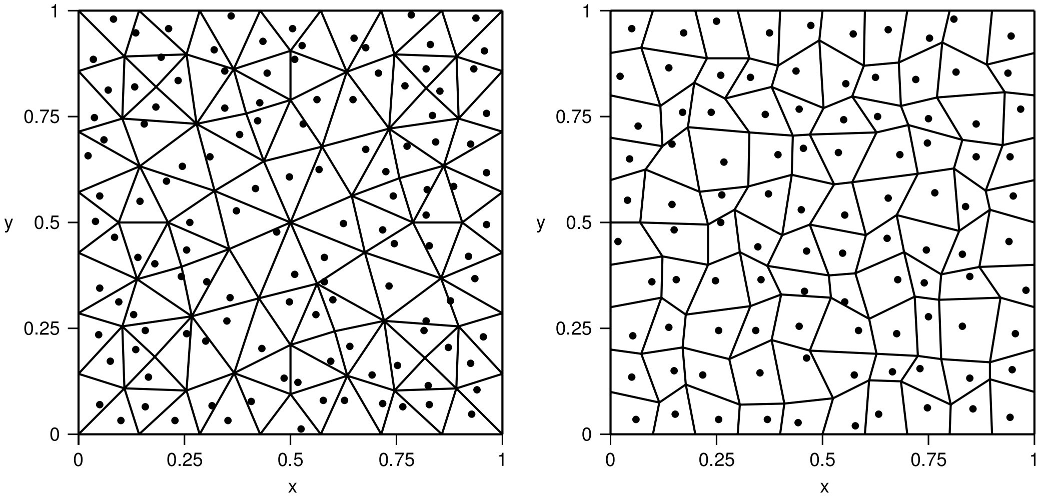

Fig. 8. Reference cell points with random locations obtained with $\alpha = 0.9$ for a triangular Delaunay mesh (left) and a deformed quadrilateral mesh (right).

Table 3
Errors and convergence rates for the low Péclet case using triangular Delaunay meshes and random locations of the reference cell points.

I	$\alpha = 0.5$				$\alpha = 0.7$				$\alpha = 0.9$			
	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
944	1.63E−03	–	1.28E−02	–	2.44E−03	–	2.12E−02	–	3.75E−03	–	3.44E−02	–
4 038	4.63E−04	1.73	2.69E−03	2.14	6.99E−04	1.72	4.90E−03	2.02	1.07E−03	1.73	8.89E−03	1.86
16 374	1.19E−04	1.95	7.61E−04	1.81	1.86E−04	1.89	1.33E−03	1.86	2.92E−04	1.85	2.35E−03	1.90
44 050	4.53E−05	1.95	3.53E−04	1.56	6.90E−05	2.01	6.23E−04	1.54	1.05E−04	2.07	1.23E−03	1.31

Table 4
Errors and convergence rates for the low Péclet case using deformed meshes and random locations of the reference cell points.

I	$\alpha = 0.5$				$\alpha = 0.7$				$\alpha = 0.9$			
	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
1 600	4.64E−04	–	2.37E−03	–	5.80E−04	–	4.04E−03	–	7.42E−04	–	6.92E−03	–
6 400	1.33E−04	1.80	8.09E−04	1.55	1.76E−04	1.72	1.16E−03	1.80	2.31E−04	1.68	1.61E−03	2.11
16 900	6.48E−05	1.48	3.70E−04	1.61	8.55E−05	1.48	5.05E−04	1.71	1.13E−04	1.48	7.05E−04	1.70
44 100	2.36E−05	2.11	1.66E−04	1.67	3.14E−05	2.09	2.19E−04	1.74	4.14E−05	2.09	2.89E−04	1.86

7.1.2. Low Péclet with random reference cell points

We now consider simulations with a random location of the reference cell point, *i.e.* any point in c_i is an admissible location of q_i. To do so, an aleatory displacement of the mass centre is performed, controlled by a given deformation factor. More precisely, for each cell c_i, let the point $p_i \in c_i$ be a random point given by

$$p_i = \frac{\sum\limits_{n \in S_i} \xi_{in} v_n}{\sum\limits_{n \in S_i} \xi_{in}},$$

where $\xi_{in} \in [0, 1]$ is a random variable following the uniform law. We then define $q_i = m_i + \alpha\,(p_i - m_i)$, with $\alpha \in [0, 1]$ the deformation factor. Notice that $\alpha = 1$ gives $q_i = p_i$ and the point q_i can belong to the boundary of the cell, leading to an ill-conditioned problem.

We consider the low Péclet solution with triangular Delaunay meshes and deformed meshes but with random locations q_i taking α equals to 0.5, 0.7, and 0.9 to avoid ill-conditioned situations (see Fig. 8).

Tables 3 and 4 provide the convergence rates for Delaunay and deformed meshes, respectively, using the $\mathbb{P}_1$ interpolation. We get an approximated second-order accuracy even with 90% of deformation ($\alpha = 0.9$) for the L^1-norm. The convergence rates are not so straightforward since the cell collocation point q_i randomly changes from cell to cell and from mesh to mesh. Additionally, we obtain more accurate approximations with low deformation factors as expected.

7.1.3. Low Péclet number with weighted reconstructions

The quadratic functional (10) depends on two sets of coefficients: the target coefficients θ_{ni} and the weights ω_{ni}. The questions we tackle now is the impact of target coefficients and weights for a very simple situation such as the Low Péclet

Table 5
Errors and convergence rates for the low Péclet case with uniform (left panel) and non-uniform (right panel) weights using the mass centre as the location of the reference cell points.

I	Unweighted				Weighted			
	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
944	6.70E−04	–	3.01E−03	–	6.71E−04	–	3.01E−03	–
4 038	1.44E−04	2.12	7.89E−04	1.84	1.44E−04	2.12	7.89E−04	1.84
16 374	3.61E−05	1.98	2.18E−04	1.84	3.61E−05	1.98	2.18E−04	1.84
44 050	1.30E−05	2.06	7.94E−05	2.04	1.30E−05	2.06	7.94E−05	2.04

Table 6
Errors and convergence rates for the low Péclet case with uniform (left panel) and non-uniform (right panel) weights using random reference cell points taking $\alpha = 0.9$.

I	Unweighted				Weighted			
	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
1 600	7.42E−04	–	6.92E−03	–	7.48E−04	–	6.76E−03	–
6 400	2.31E−04	1.68	1.61E−03	2.11	2.30E−04	1.70	1.63E−03	2.05
16 900	1.13E−04	1.48	7.05E−04	1.70	1.13E−04	1.46	7.35E−04	1.64
44 100	4.14E−05	2.09	2.89E−04	1.86	4.14E−05	2.10	2.92E−04	1.92

Table 7
Errors and convergence rates for the large Péclet test using refined triangular Delaunay meshes.

I	E_1	O_1	E_∞	O_∞
3 448	1.40E−03	–	8.00E−03	–
8 918	3.94E−04	2.67	3.03E−03	2.04
23 110	1.67E−04	1.81	1.22E−03	1.92
67 872	5.57E−05	2.03	5.42E−04	1.50

problem. To this end, we test several sets of weights and target coefficients and assess the accuracy sensitivity with respect to these parameters. We recall that all the previous numerical tests have been carried out with $\omega_{ni} = 1$ and the target coefficients θ_{ni} were given by (12). We first consider the weighted vertex reconstruction setting

$$\omega_{ni} = \frac{\frac{1}{|v_n c_i|}}{\sum\limits_{j\in\mu(n)} \frac{1}{|v_n c_j|}} \tag{25}$$

and the target coefficients θ_{ni} as (12). We report in Table 5 the errors and the convergence rates using regular Delaunay meshes with $q_i = m_i$. The left panel presents the errors with a uniform distribution of the weights ($\omega_{ni} = 1$), unweighted case, whereas the right panel gives the results with the weighted interpolation. In the same way, we report in Table 6 the results using deformed meshes with random locations q_i taking $\alpha = 0.9$.

The tables clearly demonstrate that the parameters of the reconstruction procedure have no impact on the accuracy since we exactly obtain the same errors for the Delaunay mesh case and very few differences with the deformed mesh case. Due to this negative result, we have tested an alternative choice to (25) setting $\omega_{ni} = \frac{|c_i|}{\sum_{j\in\mu(n)} |c_j|}$ but the numerical simulations provide exactly the same errors. The weights $\omega_{ni} = \frac{|v_n q_i|}{\sum_{j\in\mu(n)} |v_n q_j|}$ also produce the same results both for the Delaunay and the deformed meshes in comparison with the unweighted case. In complement with this negative result, several experiences with different choices of the target coefficients θ_{ni} have been carried out (using the cell centroid–vertex distance for instance) and also provide the same errors. We conclude that for homogeneous problem (diffusion and convection are constant) and regular meshes, the original choice is appropriated.

Remark 7.1. For anisotropic problems for instance, the weighted reconstruction might be of interest to integrate the anisotropic effects but such a question is out of the scope of the present study.

Remark 7.2. The values of the weights may be of crucial importance when dealing with highly stretched meshes with surface curvature (see [28,27]).

7.1.4. Large Péclet number

We now turn to the large Péclet number case with refined triangular Delaunay meshes to carry out the simulations (see Fig. 7, right). We use the mass centre $q_i = m_i$ as the location of the reference cell point and the $\mathbb{P}_1$ reconstruction method for the vertices. Table 7 shows that the second-order accuracy is achieved even with a high convective regime and a thin boundary layer. In particular no oscillations are detected, even for coarse meshes, due to the upwind scheme used in the convective contribution. No stability condition with respect to the space parameter is required to guarantee an eligible solution.

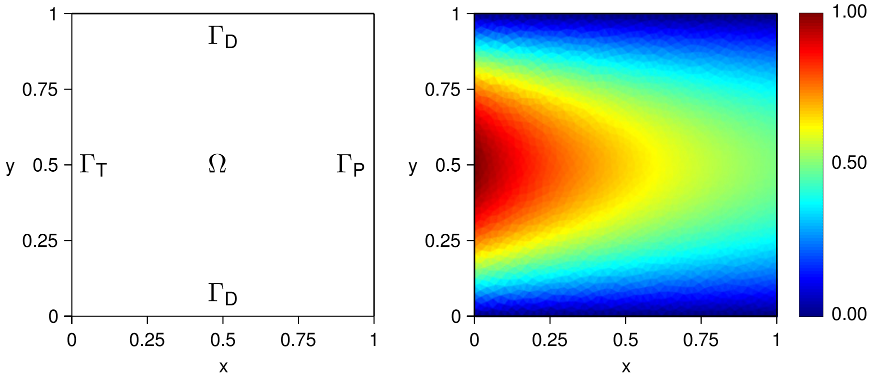

Fig. 9. Domain and boundary partitions (left) and the exact solution with a fine mesh (right).

Table 8
Errors and convergence rates with an inflow condition on the right boundary and an outflow condition on the left boundary.

I	(i)				(ii)				(iii)			
	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
944	1.88E+00	–	9.98E+01	–	2.66E+00	–	2.57E+02	–	5.62E−01	–	3.04E+01	–
4 040	9.45E−01	0.94	7.32E+01	0.43	3.04E−01	2.98	3.55E+01	2.72	2.00E−01	1.42	1.55E+01	0.93
16 364	2.87E−01	1.70	2.58E+01	1.49	4.17E−02	2.84	4.03E+00	3.11	4.24E−02	2.22	3.80E+00	2.01
44 050	1.02E−01	2.09	1.01E+01	1.89	1.72E−02	1.79	1.71E+00	1.73	1.73E−02	1.81	1.72E+00	1.60

7.2. *Convection–diffusion problem with Dirichlet and Neumann conditions*

In this subsection we check the method capacity to handle essential and natural boundary conditions for convection–diffusion problems. In particular, the impact of the total Neumann and the partial Neumann boundary conditions is studied in function of the velocity direction (inflow or outflow convective flux). Assuming the solution $\phi(x, y) = \frac{4y(1-y)}{x+1}$ (see Fig. 9, right), a normalized diffusive coefficient $\kappa = 1$, and a constant velocity vector $V = (u, 0)$, the source term is given by

$$f(x, y) = \frac{2}{x+1} - y(1-y)\left(\frac{u}{(x+1)^2} + \frac{2}{(x+1)^3}\right).$$

We prescribe the homogeneous Dirichlet conditions $\phi_{\mathrm{D}}(x, 0) = 0$ and $\phi_{\mathrm{D}}(x, 1) = 0, x \in [0, 1]$, while we impose the partial flux condition $g_{\mathrm{P}}(1, y) = y(1-y), y \in]0, 1[$, on the right side and the total flux condition $g_{\mathrm{T}}(0, y) = 4y(y-1)(1+u)$, $y \in]0, 1[$, on the left side (see Fig. 9, left). Successive regular triangular Delaunay meshes are used (see Fig. 7, left) to carry out the simulations and to compute the convergence rates.

We recall that for the inner vertices v_n we have chosen $\gamma(n) = \mu(n)$ and the vertices on Γ_{D} do not require any polynomial reconstruction. When dealing with vertices on Γ_{P} or Γ_{T} three different strategies are proposed:

(i) to use $\mu(n)$, *i.e.* no ghost cells;
(ii) to use $\gamma(n)$ and the approximations (4) and (5a);
(iii) to use $\gamma(n)$ and the approximations (4) and (5b).

In the next two subsections we assess the convergence rates depending on the velocity direction such that an inflow condition is prescribed on the right boundary and an outflow condition is prescribed on the left boundary and vice-versa.

7.2.1. *Negative velocity*

We first assume the velocity $V = (-80, 0)$ such that the partial Neumann condition is prescribed on the inflow boundary and the total Neumann condition is prescribed on the outflow boundary.

We report in Table 8 the results considering the three strategies and using the $\mathbb{P}_1$ interpolation. We obtain very low accurate results in all the cases and the errors are concentred close to the boundaries where the natural conditions are prescribed. To explain such a negative result, let us consider the extreme case where the viscosity vanishes (the Péclet number is infinite). Imposing the total Neumann condition leads to impose the flux at the boundary for an outflow condition which is not compatible for this hyperbolic situation. For large Péclet number, the total Neumann condition still provides an ill-conditioned problem and fine enough meshes are necessary such that the viscosity discrete term compensates the convective discrete term. On the other hand, to prescribe the partial Neumann condition for an inflow boundary with zero viscosity corresponds to skip the boundary condition and the problem is not well-posed. Moreover, the discretization of the convective flux gives rise to a downwind scheme which also generates instabilities. To sum up, one must not prescribe the partial Neumann condition for an inflow situation and total Neumann condition for an outflow situation.

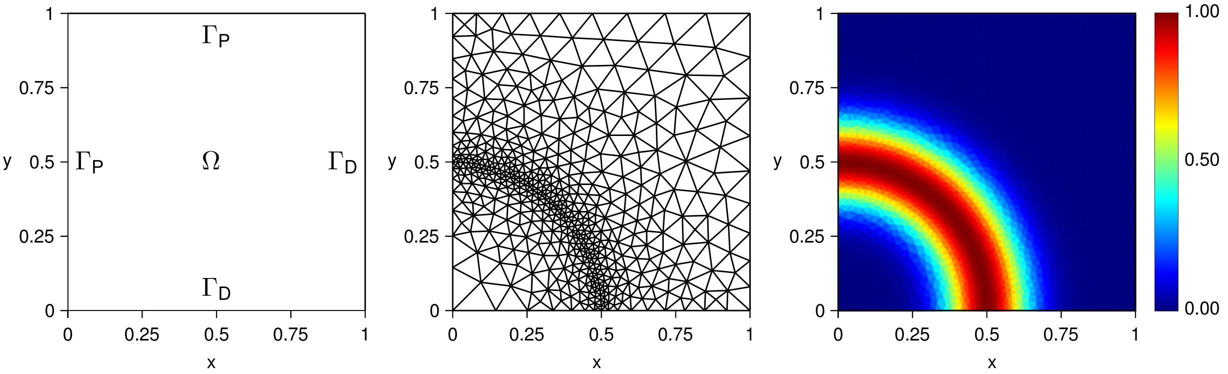

Fig. 10. Domain and boundary partitions (left), refined triangular Delaunay mesh (centre), and exact solution with a fine mesh.

Table 9
Errors and convergence rates with an inflow condition on the left boundary and an outflow condition on the right boundary.

I	(i)				(ii)				(iii)			
	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
944	1.06E−03	–	2.23E−03	–	1.17E−03	–	6.21E−03	–	1.05E−03	–	1.79E−03	–
4 040	2.54E−04	1.96	5.30E−04	1.98	2.66E−04	2.03	1.63E−03	1.84	2.53E−04	1.96	4.16E−04	2.00
16 364	6.35E−05	1.98	1.35E−04	1.96	6.53E−05	2.01	6.68E−04	1.28	6.34E−05	1.98	1.02E−04	2.01
44 050	2.42E−05	1.95	6.19E−05	1.57	2.42E−05	2.01	2.54E−04	1.95	2.40E−05	1.96	4.00E−05	1.89

Table 10
Errors and convergence rates for the pure convection case.

I	E_1	O_1	E_∞	O_∞
970	2.87E−03	–	1.86E−02	–
4 344	4.13E−04	2.58	4.57E−03	1.87
14 722	1.29E−04	1.91	1.27E−03	2.10
41 922	3.86E−05	2.30	4.54E−04	1.96

7.2.2. Positive velocity

We now assume the velocity $V = (80, 0)$ so that the total flux is prescribed on the left boundary with an inflow condition and the partial flux is prescribed on the right boundary with an outflow condition.

Table 9 shows that we obtain a second-order approximation with very accurate results. Moreover, the three techniques show very similar results for the L^1-norm error but we observe that option (iii) provides the most accurate approximation compared to the other techniques. Given that, we conclude that more accurate approximations of ϕ at the vertices on Γ_T are then achieved when we use Eq. (5b) rather than Eq. (5a).

7.3. Pure convection problem

To check the scheme capacity to deal with a pure convective problem we consider an inviscid fluid with a constant velocity $V = (-y, x)$. On the bottom side, we prescribe the Dirichlet boundary condition $\phi_D(x, 0) = \exp(-50(x-0.5)^2)$, $x \in [0, 1]$, whereas on the right side we prescribe the homogeneous Dirichlet boundary condition $\phi_D(1, y) = 0, y \in [0, 1]$. On the top and on the left sides, we assume that the outflow condition is the partial flux (see Fig. 10, left). We obtain the exact solution $\phi(x, y) = \exp(-50(r-0.5)^2)$, where $r^2 = x^2 + y^2$, from the quarter of revolution deriving from the Dirichlet boundary condition on the bottom side (see Fig. 10, right). To compute the numerical solution and the convergence rate, we use successive finer triangular Delaunay meshes with a local refinement around the circumference of radius $r = 0.5$ (see Fig. 10, centre).

Table 10 shows that we get an effective second-order accuracy with $\mathbb{P}_1$ even when dealing with a pure convective case. No oscillations are reported even in the large gradient area due to the upwinding of the convective flux and we get a second-order accuracy. The numerical experiment demonstrates the ability of the scheme to handle pure convective equations without artificial diffusion to stabilize the solution. Of course, such a scheme is not adapted to compute discontinuous solutions and stabilization procedures such as MUSCL method have to be activated.

7.4. Diffusion–reaction problem

The question of the diffusion–reaction problem is tackled in this section in order to check the scheme ability to provide a second-order of convergence even when dealing with stiff reactive problems. We recall that the treatment of the reactive

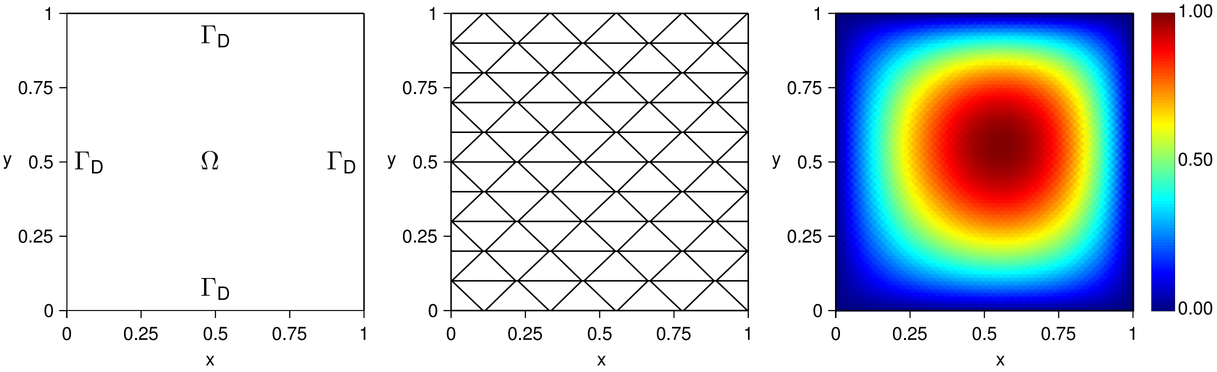

Fig. 11. Domain and boundary partitions (left), hour-glass mesh (centre), and exact solution with a fine mesh.

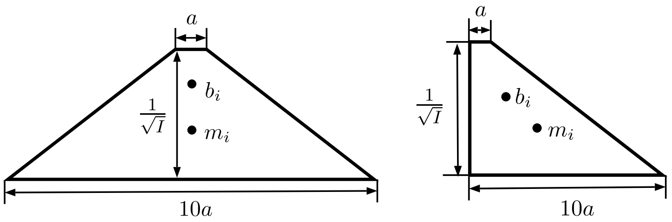

Fig. 12. Trapezoidal cells from the hour-glass meshes.

Table 11
Errors and convergence rates for the diffusion–reaction problem with random reference cell points using Eq. (15).

I	$\alpha = 0.5$				$\alpha = 0.7$				$\alpha = 0.9$			
	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
100	2.24E−02	–	7.88E−02	–	2.81E−02	–	1.02E−01	–	3.42E−02	–	1.31E−01	–
400	1.16E−02	0.95	8.92E−02	0.18	1.57E−02	0.84	1.29E−01	0.34	1.98E−02	0.79	1.70E−01	0.37
1600	5.00E−03	1.21	2.93E−02	1.61	6.93E−03	1.18	4.00E−02	1.69	8.85E−03	1.16	5.07E−02	1.74
6400	2.41E−03	1.05	1.65E−02	0.83	3.35E−03	1.05	2.31E−02	0.79	4.27E−03	1.05	2.95E−02	0.78

Table 12
Errors and convergence rates for the diffusion–reaction problem with random reference cell points using Eq. (16).

I	$\alpha = 0.5$				$\alpha = 0.7$				$\alpha = 0.9$			
	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
100	7.49E−03	–	2.61E−02	–	8.23E−03	–	3.79E−02	–	1.11E−02	–	6.64E−02	–
400	1.95E−03	1.94	6.89E−03	1.92	2.22E−03	1.89	9.72E−03	1.97	3.10E−03	1.84	1.92E−02	1.79
1600	4.61E−04	2.08	2.45E−03	1.49	5.09E−04	2.13	4.55E−03	1.10	7.04E−04	2.14	1.07E−02	0.85
6400	1.15E−04	2.00	5.17E−04	2.24	1.23E−04	2.05	9.56E−04	2.25	1.52E−04	2.21	2.11E−03	2.34

term is not the same of the diffusive and convective terms since it does not derive from a flux expression. We consider the exact solution $\phi(x, y) = 3.41xy(\exp(x) - \exp(1))(\exp(y) - \exp(1))$ with $\kappa = 1, r = 1 \times 10^6$, and homogeneous Dirichlet boundary conditions (see Fig. 11, left and right). The source term is given by

$$\begin{aligned} f(x, y) = {} & 3.41\Big(\exp(x + 1)(x + 2)y + \exp(y + 1)(y + 2)x - 2\exp(x + y)(xy + x + y)\Big) \\ & + 3.41 \times 10^6 xy(\exp(x) - \exp(1))(\exp(y) - \exp(1)). \end{aligned}$$

We study here the impact of the two proposals (15) and (16) to compute the reactive part and we consider different scenarios for the location of the reference cell points. To this end, we use a regular hour-glass mesh (see Fig. 11, centre) composed of trapezoidal cells whose mass centres do not match with the centroids (see Fig. 12). In fact, we state that the longer base is 10 times larger than the smaller base in order to get a relevant difference between the mass centres and the centroids.

We first consider successive finer hour-glass meshes where the reference cell points are random points which we compute as explained in Section 7.1.2 with 50%, 70%, and 90% of deformation. To compute the reactive part we use the approximations (15) and (16) and we report in Tables 11 and 12 the errors and the convergence rates.

We only achieve first-order convergence when we use relation (15) whereas we get an optimal and effective second-order convergence with the approximation (16). Indeed, in the finite volume context, $\mathcal{R}_i$ is supposed to be an approximation of the

Table 13
Errors and convergence rates for the diffusion–reaction problem with centroids and mass centres using Eq. (15).

I	$q_i = b_i$				$q_i = m_i$			
	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
100	2.74E−02	–	7.48E−02	–	1.68E−02	–	3.59E−02	–
400	1.15E−02	1.25	3.90E−02	0.94	4.01E−03	2.07	8.32E−03	2.11
1600	5.31E−03	1.12	1.93E−02	1.02	9.77E−04	2.04	2.00E−03	2.05
6400	2.46E−03	1.11	9.14E−03	1.08	2.41E−04	2.02	4.90E−04	2.03

Table 14
Errors and convergence rates for the 3D low Péclet number case.

I	E_1	O_1	E_∞	O_∞
8 191	5.29E−03	–	2.46E−02	–
26 795	2.25E−03	2.34	9.91E−03	2.49
65 496	1.15E−03	2.88	5.06E−03	2.91
144 043	6.67E−04	2.38	4.14E−03	0.88

mean value of $r\phi$ over cell c_i, that is, $\mathscr{R}_i \approx \frac{1}{|c_i|}\int_{c_i} r\phi \, \mathrm{d}A$. For random locations of the reference cell points, relation (15) does not provide, in the general case, a second-order approximation of the mean value and the global method order is reduced to a first-order one. On the contrary, the integration technique with the vertices proposed in relation (16) always provide a second-order approximation of the mean value, whatever points q_i are located.

We now analyse two more cases with Eq. (15): we consider $q_i = b_i$ and $q_i = m_i$. We report the results in Table 13.

Eq. (15) provides a first-order method when using centroids. Indeed, for the particular case of hour-glass meshes, the distance between the mass centres and the centroids are larger than $\mathcal{O}(h^2)$. We recover an effective second-order scheme when the location of the reference cell points are the mass centres since it provides a second-order approximation of the mean values. The two last examples highlight that the one Gauss point formula only provides the correct order with the mass centres and not with the centroids. Of course, for triangular cells the mass centre and the centroid is the same point.

7.5. 3D convection–diffusion–reaction problems

We dedicate this section to assess and evaluate the robustness and the capacity of the scheme to provide second-order accuracy when handling three-dimensional convection–diffusion–reaction problems. To this end, given the numerical approximation $\Phi^\star = (\phi_i^\star)_{i=1,\dots,I}$ of function ϕ in a mesh $\mathcal{T}$ composed of tetrahedron cells, we evaluate the L^1- and the L^∞-norm errors between the numerical and the exact solutions, given by

$$E_1(\mathcal{T}) = \sum_{i=1}^{I} |\phi_i^\star - \phi(q_i)|\,|c_i| \quad \text{and} \quad E_\infty(\mathcal{T}) = \max_{i=1}^{I} |\phi_i^\star - \phi(q_i)|,$$

respectively, where $|c_i|$ is now the volume of cell c_i. We also compute the convergence rate of the L^γ-norm error, with $\gamma = 1, \infty$, obtained with two different meshes $\mathcal{T}_1$ and $\mathcal{T}_2$ consisting of I_1 and I_2 cells, respectively, as

$$O_\gamma(\mathcal{T}_1, \mathcal{T}_2) = 3\frac{|\log(E_\gamma(\mathcal{T}_1))/(E_\gamma(\mathcal{T}_2))|}{|\log(I_1/I_2)|}.$$

7.5.1. Convection–diffusion with Low Péclet number

We consider a low Péclet number flow in the domain $\Omega =]0, 1[^3$ with a constant velocity $V = (1, 2, 3)$ and a constant diffusion coefficient $\kappa = 1$. The exact solution is given by $\phi(x, y, z) = C\alpha(x)\beta(y)\gamma(z)$ with $C \in \mathbb{R}$ and

$$\alpha(x) = x - \frac{\exp(x) - 1}{\exp(1) - 1}, \qquad \beta(y) = \frac{1}{2}\left(y - \frac{\exp(2y) - 1}{\exp(2) - 1}\right), \qquad \gamma(z) = \frac{1}{3}\left(z - \frac{\exp(3z) - 1}{\exp(3) - 1}\right),$$

which implies the source term $f(x, y, z) = C(\alpha(x)\beta(y) + \beta(y)\gamma(z) + \alpha(x)\gamma(z))$. We prescribe the homogeneous Dirichlet boundary condition $\phi_{\mathrm{D}}(x, y, z) = 0$ on $\Gamma_{\mathrm{D}} = \Gamma$. In order to normalize the exact solution such that the maximum of ϕ in $\bar{\Omega}$ is close to 1, we take $C = 625$ (see Fig. 13, right).

To carry out the simulations we use successive finer regular tetrahedral Delaunay meshes (see Fig. 13, left) and we report the errors and the convergence rates in Table 14.

We get an effective second-order accuracy and no oscillations are reported. Such an example demonstrates the capacity of the method to be adapted to the three dimensional context and to provide accurate approximations. A lot of numerical experiences (which we do not report here for the sake of simplicity) have been performed with high Péclet number and pure convective flows, where we obtained the same order of convergence of the 2D counterparts.

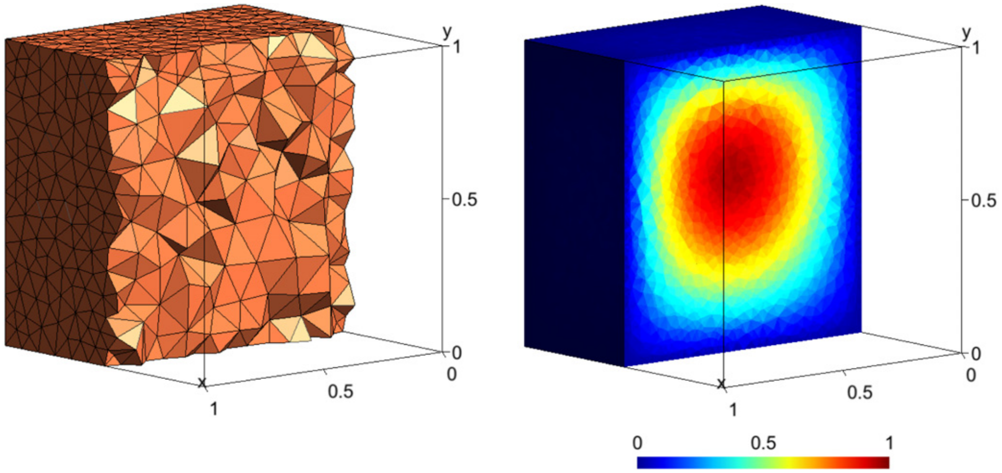

Fig. 13. Cross section of a regular tetrahedral Delaunay mesh (right) and exact solution with a fine mesh (right).

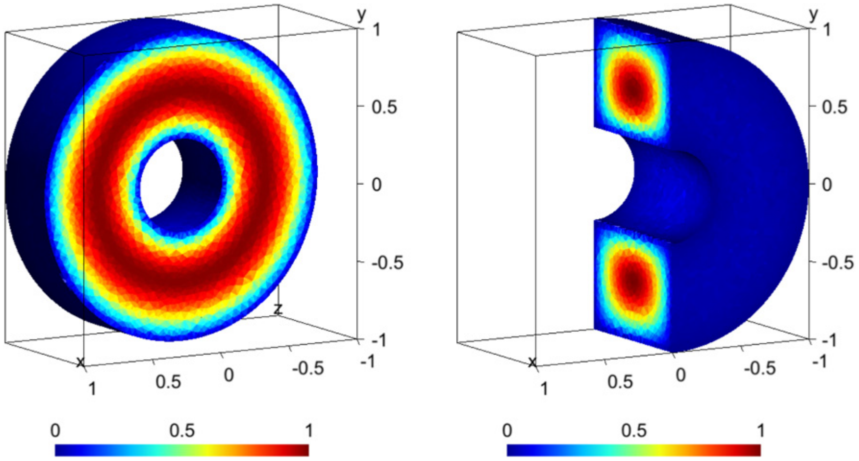

Fig. 14. Transversal (left) and longitudinal (right) cross sections of the exact solution with a fine mesh.

7.5.2. Diffusion–reaction case

In this test the domain Ω is a pipe with $0.3 \leqslant \sqrt{x^2+y^2} \leqslant 1$ and $z \in [0, 1]$. The exact solution is

$$\phi(x,y,z) = -7.16(z-z^2)\left(\frac{\ln(x^2+y^2)}{\ln 0.3} + \frac{200}{91}(x^2+y^2-1)\right),$$

the diffusion coefficient is $\kappa = 1$, and the reactive coefficient is $r = 10^6$ (see Fig. 14, left and right). The source term is given by

$$f(x,y,z) = -14.32\left(\frac{\ln(x^2+y^2)}{\ln 0.3} + \frac{200}{91}(x^2+y^2-1) - \frac{400}{91}(z-z^2)\right) + 10^6\phi(x,y,z),$$

and we prescribe the homogeneous Dirichlet boundary condition $\phi_{\mathrm{D}}(x,y,z) = 0$ on $\Gamma_{\mathrm{D}} = \Gamma$.

To carry out the simulations we use successive finer regular tetrahedral Delaunay meshes (see Fig. 15, left and right) and we consider two situations. In the first one we locate the reference cell points at the mass centres of the cells (which match with the centroids) and we report in Table 15 the errors and the convergence rates. As expected, we get an effective second-order accuracy and no oscillations are reported. Notice that in this situation the use of Eq. (15) guarantees the second-order accuracy since we are dealing with reference cell points located at the mass centres. In the second run, we consider reference cell points with random locations extending the use of Eq. (16) to the three-dimensional case given by Eq. (17). We consider $\alpha = 0.5, 0.7, 0.9$ and we report the results in Table 16. We still obtain a second-order accuracy even with a stiff source term which proves the robustness of the method. We highlight that in this experiment the use of Eq. (17) to compute the reactive term is crucial to achieve second-order approximations of the underlying solution.

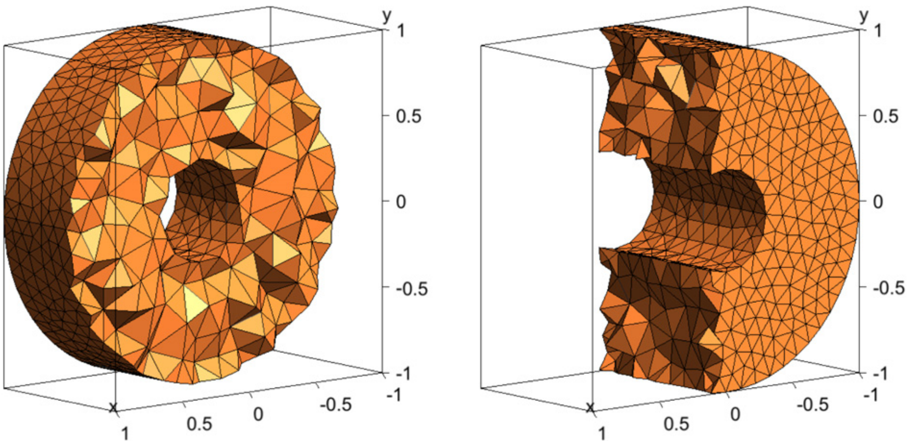

Fig. 15. Transversal (left) and longitudinal (right) cross sections of a uniform tetrahedral Delaunay mesh.

Table 15
Errors and convergence rates for the 3D diffusion–reaction case using reference cell points located at the mass centres.

I	E_1	O_1	E_∞	O_∞
7 589	3.37E−02	–	1.35E−01	–
24 936	1.58E−02	2.07	6.04E−02	2.19
61 565	8.04E−03	2.91	3.30E−02	2.61
130 058	4.64E−03	2.38	2.10E−02	1.96

Table 16
Errors and convergence rates for the 3D diffusion–reaction case using reference cell points with random locations.

I	$\alpha = 0.5$				$\alpha = 0.5$				$\alpha = 0.9$			
	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
7 589	3.74E−02	–	3.15E−01	–	3.91E−02	–	3.92E−01	–	4.31E−02	–	5.54E−01	–
24 936	1.75E−02	2.07	1.45E−01	2.13	1.84E−02	2.06	1.79E−01	2.14	2.04E−02	2.04	3.05E−01	1.63
61 565	8.97E−03	2.90	7.55E−02	2.82	9.43E−03	2.90	8.24E−02	3.36	1.04E−02	2.90	1.20E−01	4.03
130 058	5.16E−03	2.40	4.45E−02	2.28	5.41E−03	2.40	5.14E−02	2.05	5.97E−03	2.42	8.06E−02	1.73

7.6. Time-dependent convection–diffusion test

We now check the capacity of the scheme to handle time-dependent problems and provide a second-order accuracy. To this end, we consider a time-dependent convection–diffusion problem where we slightly modify the low Péclet number steady-state test solution presented in Section 7.1 by adding the time derivative term. We assume a constant diffusion coefficient $\kappa = 1$ and a constant velocity $V = (1, 2)$ with $t^0 = 0$ and $t^{\mathrm{f}} = 1$. The exact solution is given by $\phi(x, y, t) = 65\alpha(x)\beta(y)\cos(2\pi t)$ in $\Omega \times [0, 1]$ with

$$\alpha(x) = x - \frac{\exp(ux) - 1}{\exp(u) - 1}, \qquad \beta(y) = \frac{1}{2}\left(y - \frac{\exp(2y) - 1}{\exp(2) - 1}\right),$$

and the source term is $f(x, y, t) = 65(\alpha(x) + \beta(y))\cos(2\pi t) - 130\pi\alpha(x)\beta(y)\sin(2\pi t)$. We consider the homogeneous Dirichlet boundary condition $\phi_{\mathrm{D}}(x, y, t) = 0$ on $\Gamma_{\mathrm{D}} \times [0, 1]$ with $\Gamma_{\mathrm{D}} = \Gamma$, and the initial condition $\phi^0(x, y) = \phi(x, y, 0)$.

To perform the simulations and compute the convergence rates, we consider a fixed mesh while successive finer time steps $\Delta t = \frac{1}{K}$ are used with K the number of subdivisions. Given the numerical approximation $\Phi^K = (\phi_i^K)_{i=1,\dots,I}$ of a function ϕ at the final time t^K, we evaluate the error using the L^1- and the L^∞-norms given by Eq. (24). We also compute the convergence rate of the L^γ-norm error, with $\gamma = 1, \infty$, obtained with two different time steps, K_1 and K_2, as

$$O_\gamma(K_1, K_2) = \frac{|\log(E_\gamma(K_1))/(E_\gamma(K_2))|}{|\log(K_1/K_2)|}.$$

In a first run, we use a triangular Delaunay mesh with 44 050 cells and a deformed quadrilateral mesh with 44 100 cells (see Fig. 7, right and centre), and for both cases the reference cell point is the mass centre. The errors and convergence rates are reported in Table 17. As expected, we obtain a second-order accuracy in time scheme, unconditionally stable which proves the robustness and the accuracy of the method. Notice that the scheme performs very well even with deformed meshes.

Table 17
Errors and convergence rates for the time-dependent convection–diffusion test using reference cell points located at the mass centres.

K	Delaunay mesh				Deformed mesh			
	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
10	2.02E−02	–	4.57E−02	–	2.02E−02	–	4.57E−02	–
20	4.90E−03	2.04	1.11E−02	2.04	4.91E−03	2.04	1.11E−02	2.04
40	1.21E−03	2.02	2.74E−03	2.02	1.22E−03	2.01	2.75E−03	2.01
80	2.98E−04	2.03	6.65E−04	2.04	3.08E−04	1.99	6.83E−04	2.01
160	6.95E−05	2.10	1.56E−04	2.09	8.02E−05	1.94	1.76E−04	1.96

Table 18
Errors and convergence rates for the time-dependent convection–diffusion test using reference cell points with a random location.

K	E_1	O_1	E_∞	O_∞
10	2.02E−02	–	4.58E−02	–
20	4.92E−03	2.04	1.13E−02	2.03
40	1.23E−03	2.00	2.96E−03	1.93
80	3.14E−04	1.97	9.22E−04	1.68

In a second run, we use only a deformed quadrilateral mesh with 44 100 cells and the reference cell points in a random location with 90% of deformation. We report the results in Table 18. As in the previous test, the scheme manages well very crude situations where the location of the reference cell points are far from the mass centres and second-order convergence rates in time are achieved.

The same simulations were carried out with triangular Delaunay meshes and similar results were achieved.

8. Conclusion

An extension of the original Frink's method has been proposed and implemented to perform the cell-to-vertex reconstruction both for two- and three-dimensional geometries. The vertex values are computed via linear combinations of the closest cell values where the coefficients are determined by a functional minimization. The method enables to associate the cell value at any location point inside the cell still preserving both the second-order accuracy and the robustness, even for pure convection equations. In particular, efficient discretizations of the reactive term or time-dependent problems are proposed to achieve a second-order approximation, even if the location of the reference cell points are far from the mass centre. Several numerical experiences have been carried out to assess the method performance and demonstrate the capacity to handle a wide range of situations in the context of the convection–diffusion–reaction equation.

Acknowledgements

This research was financed by FEDER Funds through Programa Operacional Factores de Competitividade—COMPETE and by Portuguese Funds through FCT—Fundação para a Ciência e a Tecnologia, within the Projects PEst-OE/MAT/UI0013/2014 and PTDC/MAT/121185/2010. The second author was also financed by project FCT-ANR/MAT-NAN/0122/2012.

References

[1] S. Clain, G.J. Machado, J.M. Nóbrega, R.M.S. Pereira, A sixth-order finite volume method for the convection–diffusion problem with discontinuous coefficients, Comput. Methods Appl. Mech. Engrg. 267 (2013) 43–64.

[2] S. Diot, R. Loubère, S. Clain, The MOOD method in the three-dimensional case: very-high-order finite volume method for hyperbolic systems, Internat. J. Numer. Methods Fluids 73 (4) (2013) 362–392.

[3] A. Hidalgo, M. Dumbser, ADER Schemes for nonlinear systems of stiff advection–diffusion–reaction equations, J. Sci. Comput. 48 (1–3) (2011) 173–189.

[4] L. Ivan, C.P.T. Groth, High-order solution-adaptative central essentially non-oscillatory (CENO) method for viscous flows, AIAA Paper 2011-367, 2011.

[5] C. Ollivier-Gooch, High-order ENO schemes for unstructured Meshes based on least-squares reconstruction, AIAA Paper 97-0540, 1997.

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

[7] F. Brezzi, K. Lipnikov, M. Shashkov, Convergence of mimetic finite difference methods for diffusion problems on polyhedral meshes, SIAM J. Numer. Anal. 43 (2005) 1872–1896.

[8] C. Berthon, Y. Coudière, V. Desveaux, Second-order MUSCL schemes based on dual mesh gradient reconstrution (DMGR), ESAIM Math. Model. Numer. Anal. 48 (2014) 583–602.

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

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

[11] R. Eymard, T. Gallouët, R. Herbin, Finite volume approximation of elliptic problems and convergence of an approximate gradient, Appl. Numer. Math. 37 (2001) 31–53.

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

[13] P. Chandrashekar, A. Garg, Vertex-centroid finite volume scheme on tetrahedral grids for conservation laws, Comput. Math. Appl. 65 (2013) 58–74.

[14] N.T. Frink, Recent progress toward a three-dimensional unstructured Navier–Stokes flow solver, AIAA Paper 94-0061, 1994.

[15] P. Jawahar, H. Kamath, A high-resolution procedure for Euler and Navier–Stokes computations on unstructured Grids, J. Comput. Phys. 164 (2000) 165–203.
[16] G. Manzini, A. Russo, A finite volume method for advection–diffusion problems in convection-dominated regimes, Comput. Methods Appl. Mech. Engrg. 197 (2008) 1242–1261.
[17] R.D. Rausch, J.T. Batina, H.T.Y. Yang, Spatial adaptation procedure on unstructured meshes for accurate unsteady aerodynamic flow computation, AIAA Paper 91-1106, 1991.
[18] E. Bertolazzi, G. Manzini, Least square-based finite volumes for solving the advection–diffusion of contaminants in porous media, Appl. Numer. Math. 51 (2004) 451–461.
[19] E. Bertolazzi, G. Manzini, A unified treatment of boundary conditions in least-square based finite-volume methods, Comput. Math. Appl. 49 (2005) 1755–1765.
[20] E. Bertolazzi, G. Manzini, On vertex reconstructions for cell-centered finite volume approximations of 2D anisotropic diffusion problems, Math. Models Methods Appl. Sci. 17 (1) (2007) 1–32.
[21] N.T. Frink, Three-dimensional upwind scheme for solving the Euler equations on unstructured tetrahedral grids (Ph.D. dissertation), Virginia Polytechnic Institute and State University, 1991.
[22] N.T. Frink, Upwind scheme for solving the Euler equations on unstructured tetrhedral meshes, AIAA J. 1 (1992) 70–77.
[23] D.G. Holmes, S.D. Connel, Solution of the 2D Navier–Stokes equations on unstructured adaptive grids, AIAA Paper 89-1392, 1989.
[24] Z. Sheng, G. Yuan, A finite volume scheme for diffusion equations on distorted quadrilateral meshes, Transport Theory Statist. Phys. 37 (2008) 171–207.
[25] Y. Coudière, J.P. Vila, P. Villedieu, Convergence rate of a finite volume scheme for a two dimensional convection–diffusion problem, M2AN Math. Model. Numer. Anal. 3 (33) (1999) 493–516.
[26] W.J. Coirier, K.G. Powell, Solution-adaptive Cartesian cell approach for viscous and inviscid flows, AIAA J. 34 (1996) 938–945.
[27] B. Diskin, J.L. Thomas, Comparison of node-centered and cell-centered unstructured finite-volume discretizations: inviscid fluxes, AIAA 2010-1079.
[28] D.J. Mavriplis, Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes, NASA/CR-2003-212683 NIA Report No. 2003-06, 2003.