Arbitrary order positivity preserving finite-volume schemes for 2D elliptic problems

Xavier Blanc, Francois Hermeline, Emmanuel Labourasse, Julie Patela

Arbitrary order positivity preserving finite-volume schemes for 2D elliptic problems

Xavier Blanc[1], Francois Hermeline[2,3], Emmanuel Labourasse[2,3], and Julie Patela[1,2]

[1]Université Paris Cité, Sorbonne Université, CNRS, Laboratoire Jacques-Louis Lions, F-75013 Paris, France.
[2]CEA, DAM, DIF, F-91297 Arpajon, France.
[3]Université Paris-Saclay, CEA DAM DIF, Laboratoire en Informatique Haute Performance pour le Calcul et la Simulation, 91297 Arpajon, France.

August 5, 2024

Abstract

The positivity preservation is very important in most applications solving elliptic problems. Many schemes preserving positivity has been proposed but are at most second-order convergent. Besides, in general, high-order schemes do not preserve positivity. In the present paper, we propose an arbitrary-order positivity preserving method for elliptic problems in 2D. We show how to adapt our method to the case of a discontinuous and/or tensor-valued diffusion coefficient, while keeping the expected order of convergence. We assess the new scheme on several test problems.

Keywords— Finite volume method, elliptic problem, anisotropic diffusion, positivity preserving, high order

Contents

1 Introduction

This paper describes a follow-up of two recently published works [6, 7]. In the former work, we designed a positivity preserving and arbitrary-order numerical method for an elliptic equation in 1D. In the latter one, we showed that the approach used in 1D extends to second-order accurate methods in 2D. Our goal in this paper is to propose the first arbitrary-order positivity preserving finite volume method for elliptic problems in 2D. Furthermore this method allows polygonal meshes of almost any shape to be used.

The model we consider is

$$\begin{cases} -\boldsymbol{\nabla}\cdot(\boldsymbol{\kappa}\boldsymbol{\nabla}\bar{u}) + \lambda\bar{u} = f & \text{in} \quad \Omega, \\ \bar{u} = g_D & \text{on} \quad \Gamma_D, \\ \boldsymbol{\kappa}\boldsymbol{\nabla}\bar{u}\cdot\mathbf{n} = g_N & \text{on} \quad \Gamma_N, \end{cases} \tag{1}$$

where Ω is a bounded open domain of $\mathbb{R}^2$ with $\partial\Omega = \Gamma_D \cup \Gamma_N$ ($\Gamma_D \cap \Gamma_N = \emptyset$), and $\mathbf{n} \in \mathbb{R}^2$ is the outgoing unit normal vector. The data are such that $f \in L^2(\Omega)$, $g_D \in H^{1/2}(\Gamma_D)$, $g_N \in L^2(\Gamma_N)$, $\lambda \in \mathbb{R}^+$ (if $\lambda = 0$, then $|\Gamma_D| > 0$), and $\boldsymbol{\kappa} \in L^\infty(\Omega)^{2,2}$. The tensor-valued diffusion coefficient $\boldsymbol{\kappa}$ satisfies the uniform ellipticity condition:

$$\forall \mathbf{x} \in \Omega,\ \forall \xi \in \mathbb{R}^2,\ \kappa_{\min}\|\xi\|^2 \leq \xi^t \boldsymbol{\kappa}(\mathbf{x})\xi. \tag{2}$$

where $\kappa_{\min}$ is a strictly positive coefficient. Under the above conditions, one can prove (see for instance [39], Chapter 7, in the case of Neumann or Dirichlet boundary conditions, extension to mixed boundary conditions is straightforward) that system (1) has a unique solution in $H^1(\Omega)$ which satisfies a positiveness principle, i.e. if $f \geq 0$ and $g \geq 0$, then $\bar{u} \geq 0$. One often refers to positivity preserving in the literature for this principle.

For the applications we have in mind, such as inertial confinement fusion simulations, we need to be able to solve problem (1) on (almost) arbitrary meshes. The reason for this is twofold. First, the domain Ω can be very distorted. Second, problem (1) is coupled to the Euler equations, which is descretized using a Lagrangian finite volume scheme (see [14, 31, 37]). We thus have no control on the quality of the mesh. Further, a fundamental property of the hydrodynamics scheme is to be conservative, in order to reproduce as precisely as possible singular solutions, such as shocks. Thus, the diffusion scheme applied to (1) should be conservative too, in order to preserve this property. As a consequence, the positivity of the solution cannot be recovered by merely truncating negative values: such a strategy is incompatible with conservativity.

This is why a large amount of work has been devoted to the design of positivity preserving schemes since the seminal works of [5, 34]. Among other publications, let us cite recent works [12, 13, 44, 46, 49, 51, 52, 57] and references therein about this topic. However, none of these methods is arbitrarily high-order accurate. The most advanced work in this direction is [52], which achieved third-order accuracy.

Some methods are particularly well-suited for achieving arbitrary high-order for elliptic problems. Let us cite for instance the finite-element method [17], the Virtual Element method [4], the Discontinuous Galerkin method [19], and the Hybrid High-Order method (HHO) [22]. However, very few (see [2, 3, 11, 50] and references therein) can enforce the positiveness of the unknown without imposing severe constraints on the mesh, and none of them achieve a convergence order higher than two.

In [16, 28, 35, 36, 47], Discontinuous Galerkin (DG) schemes that satisfy the maximum principle are presented for solving different types of parabolic models. These methods are based on the seminal work [55], and use limiters to preserve the positivity of the mean values, at the discrete level. An explicit integration in time is performed, leading to a parabolic constraint on the timestep which is not affordable in our context. Another approach to achieve positivity relies on the change of variable $\bar{u} = e^{\bar{v}}$. Equation (1) is transformed into a nonlinear equation. This idea has been used for instance in [8, 20, 40] to produce high-order positivity preserving methods. A DG method is proposed in [8, 20], which preserves the positivity of the solution of the Fischer-Kolmogoroff-Petrovsky-Piscounov equation. In [40], a similar scheme is designed in the framework of the HHO method. These two methods are arbitrary-order accurate in space, and allow for general polygonal meshes. They can be considered as alternatives to the present work.

A reason for not using these methods in our context, is that their coupling with other models can be problematic since the degrees of freedom of the different discrete operators approximations do not match. Besides, the method presented in this paper is cell-centered (with piecewise constant unknowns), which is the simplest possible set of unknowns.

To our knowledge, the method we propose here is the first arbitrary-order positivity preserving finite volume type scheme for discretizing the elliptic equation (1) on general polygonal meshes. The diffusion coefficient can be tensor-valued and/or discontinuous. We show that the arbitrary high-order accuracy is preserved even with a discontinuous

diffusion coefficient as long as discontinuities are known and coincide with edges of the mesh. We recall the main steps of the proposed method (see also [7]):

1. Integration of the equation over each cell of the mesh.
2. Transformation of this surface integral into a sum of fluxes using the divergence theorem.
3. Approximation of the fluxes using a Gauss quadrature rule on each edge of the cell.
4. Taylor expansion of the solution $\bar{u}$ in the neighborhood of each Gauss quadrature point of each edge along *two* independent privileged directions in order to obtain an approximation of $\boldsymbol{\nabla}\bar{u}$ involving the values of $\bar{u}$ and its derivatives at certain suitably chosen points, in this case the center of mass and vertices of the cell.
5. Using this Taylor expansion, estimation of $(\boldsymbol{\kappa}\boldsymbol{\nabla}\bar{u})\cdot\mathbf{n} = (\boldsymbol{\nabla}\bar{u})\cdot(\boldsymbol{\kappa}^t\mathbf{n})$.
6. Calculation of the values of $\bar{u}$ at vertices by a polynomial interpolation formula in the neighborhood of the Gauss quadrature points of each cell edge.
7. Calculation of the values of derivatives of $\bar{u}$ at centers of mass and vertices of the neighboring cells by differentiating this polynomial interpolation.
8. Transformation of the scheme into a positivity preserving non-linear approximation with a *two-point* flux structure.
9. Resolution of the non-linear system by the Picard iteration method.

The paper is structured as follows. Definitions and notations are given in Section 2. The proposed arbitrarily high-order Finite-Volume method is described in Section 3. Then, we explain how the scheme is modified to enforce the positivity in Section 4. In Section 5, we prove some nice properties of the method. Finally the arbitrary high-order accuracy and the positivity of the method are assessed in Section 6 on a range of benchmarks including cases with highly anisotropic and discontinuous diffusion coefficients.

2 Definitions and notations

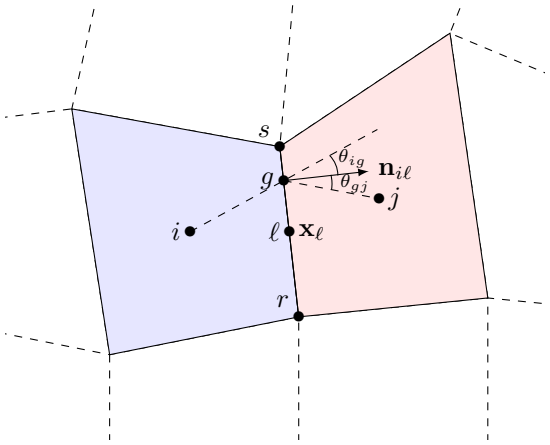


Figure 1: Example of a mesh with our notations.

Given an arbitrary mesh the cells of which are numbered from 1 to n, consider a cell denoted i and its neighbor j (see Figure 1). The center of mass of i (resp. j) is denoted by $\mathbf{x}_i$ (resp. $\mathbf{x}_j$), their common edge is ℓ and the vertices of ℓ are $\mathbf{x}_r$ and $\mathbf{x}_s$. The position of the center of the edge ℓ is $\mathbf{x}_\ell$. We denote by $\mathbf{x}_g$ a Gauss quadrature point located on the edge ℓ. The length of ℓ is $|\ell|$ and the volume of a cell i is V_i. The normal vector $\mathbf{n}_{i\ell}$ is the unit vector which is orthogonal to the edge ℓ and outgoing for the cell i. Let us emphasize that if ℓ is shared by cells i and j then $\mathbf{n}_{j\ell} = -\mathbf{n}_{i\ell}$. Let θ_{ig} (resp. θ_{gj}) be the angle between $\mathbf{x}_g - \mathbf{x}_i$ (resp. $\mathbf{x}_j - \mathbf{x}_g$) and $\mathbf{n}_{i\ell}$.

By abuse of notations, we denote by

$$\sum_{\ell\in i} \quad \left(\text{resp.} \quad \sum_{g\in\ell}\right)$$

any sum on all edges ℓ (resp. on all Gauss quadrature points g) of cell i (resp. edge ℓ).

We define $h = \min_{\ell} |\ell|$. We assume that there exists a constant θ_0 independent of h such that, for all g,

$$|\theta_0| < \frac{\pi}{2}, \quad \cos(\theta_0) < \cos(\theta_{ig}), \quad \cos(\theta_0) < \cos(\theta_{gj}). \tag{\textbf{H}}$$

This condition is close to imposing that the cells are star-shaped. However, since θ_0 does not depend on h, it is slightly stronger. Note that the higher the order, the more Gauss points there are, and the more restrictive this assumption becomes.

In the case where (**H**) is not satisfied, we replace $\mathbf{x}_i$ by the midpoint of an inner diagonal of i or by any interior point for which i is star-shaped (right-hand side of Figure 2).

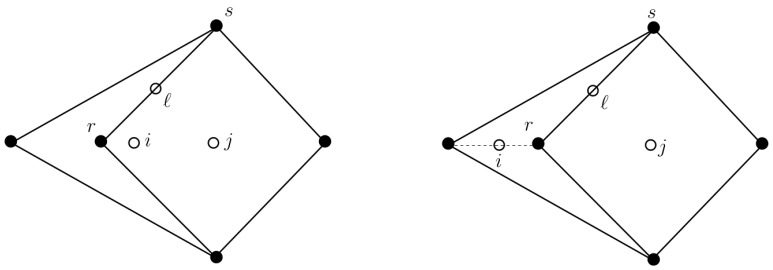


Figure 2: A non convex cell i and a convex cell j such that $\mathbf{x}_i$ and $\mathbf{x}_j$ are not separated by the line defined by edge ℓ.

Given $\mathbf{v} = (v_i)$ a vector in $\mathbb{R}^n$ we will denote respectively its Euclidian, L^2 and L^∞ norms by

$$\|\mathbf{v}\| = \left(\sum_{i=1}^{n} v_i^2\right)^{1/2}, \quad \|\mathbf{v}\|_2 = \left(\sum_{i=1}^{n} V_i v_i^2\right)^{1/2}, \quad \|\mathbf{v}\|_\infty = \max_{1\le i\le n} |v_i|$$

and we use the compact notation $\mathbf{v} > \mathbf{0}$ (resp. $\mathbf{v} \ge \mathbf{0}$) if, for all i, $v_i > 0$ (resp. $v_i \ge 0$).

3 Finite volume formulation

In this part, we detail the construction of a linear arbitrary order scheme for problem (1). Up to some details, this construction is similar to that proposed in [42] (see also [18, 21]). In particular, in [21] some proofs are provided concerning the accuracy order of the linear scheme. These proofs can be easily adapted to our derivation.

To simplify the presentation we suppose that $\boldsymbol{\kappa}$ is isotropic : $\boldsymbol{\kappa} = \kappa \boldsymbol{I}$, with $\kappa > \kappa_{\min}$. It is worth noting that the full anisotropic case can be immediately dealt with by remarking that $(\boldsymbol{\kappa}\boldsymbol{\nabla}\bar{u}) \cdot \mathbf{n} = (\boldsymbol{\nabla}\bar{u}) \cdot (\boldsymbol{\kappa}^t \mathbf{n})$ and by replacing $\mathbf{n}$ by $\boldsymbol{\kappa}^t \mathbf{n}$ in what follows. Moreover we assume that the discontinuities of κ coincide with edges of the mesh.

3.1 Approximation of the interior fluxes

The first step to design a finite volume scheme consists in integrating (1) on cell i

$$-\int_i \boldsymbol{\nabla} \cdot \kappa \boldsymbol{\nabla}\bar{u} + \int_i \lambda \bar{u} = \int_i f.$$

Thanks to the divergence formula we obtain

$$-\sum_{\ell \in i} \int_\ell \kappa \boldsymbol{\nabla}\bar{u} \cdot \mathbf{n}_{i\ell} + \int_i \lambda \bar{u} = \int_i f. \tag{3}$$

Using a k-th order accurate Gauss's quadrature formula for approximating the flux through the edge ℓ

$$\bar{\mathcal{F}}_{i\ell} = \int_\ell \kappa \boldsymbol{\nabla}\bar{u} \cdot \mathbf{n}_{i\ell}$$

we have

$$-\sum_{\ell \in i} |\ell| \sum_{g \in \ell} \omega_g \kappa(\mathbf{x}_g) \left(\boldsymbol{\nabla}\bar{u}\right)(\mathbf{x}_g) \cdot \mathbf{n}_{i\ell} + \int_i \lambda \bar{u} = \int_i f + \mathcal{O}(h^k),$$

where $\omega_g > 0$ and $\mathbf{x}_g$ are respectively the weights and the points of the quadrature. Thus we have to approximate

$$\kappa(\mathbf{x}_g)\left(\boldsymbol{\nabla}\bar{u}\right)(\mathbf{x}_g)\cdot\mathbf{n}_{i\ell}.$$

Suppose that $\bar{u}\in W^{1,\infty}(\Omega)$ and denote :

$$N_q^p=\frac{1}{p!}\binom{p}{q}=\frac{1}{q!(p-q)!}.$$

A Taylor expansion at order k in the neighborhood of $\mathbf{x}_g$ gives

$$\bar{u}(\mathbf{x})=\bar{u}(\mathbf{x}_g)+(\mathbf{x}-\mathbf{x}_g)\cdot\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)+\sum_{p=2}^{k}\sum_{q=0}^{p}N_q^p\frac{\partial^p\bar{u}}{\partial x^q\partial y^{p-q}}(\mathbf{x}_g)(x-x_g)^q(y-y_g)^{p-q}+\mathcal{O}\left(\|\mathbf{x}-\mathbf{x}_g\|^{k+1}\right). \tag{4}$$

Let $\bar{\mathbf{u}}$ be the vector

$$\bar{\mathbf{u}}=(\bar{u}_i)_{1\le i\le n}, \tag{5}$$

with $\bar{u}_i$ the mean value of $\bar{u}$ in cell i

$$\bar{u}_i=\frac{1}{V_i}\int_i\bar{u}(\mathbf{x}).$$

Integrating (4) on cells i, j and dividing respectively by their volume V_i, V_j provides

$$\bar{u}_i=\bar{u}(\mathbf{x}_g)+(\mathbf{x}_i-\mathbf{x}_g)\cdot\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)+\frac{1}{V_i}\sum_{p=2}^{k}\sum_{q=0}^{p}N_q^p\frac{\partial^p\bar{u}}{\partial x^q\partial y^{p-q}}(\mathbf{x}_g)\int_i(x-x_g)^q(y-y_g)^{p-q}+\mathcal{O}\left(h^{k+1}\right),$$

$$\bar{u}_j=\bar{u}(\mathbf{x}_g)+(\mathbf{x}_j-\mathbf{x}_g)\cdot\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)+\frac{1}{V_j}\sum_{p=2}^{k}\sum_{q=0}^{p}N_q^p\frac{\partial^p\bar{u}}{\partial x^q\partial y^{p-q}}(\mathbf{x}_g)\int_j(x-x_g)^q(y-y_g)^{p-q}+\mathcal{O}\left(h^{k+1}\right).$$

hence

$$(\mathbf{x}_g-\mathbf{x}_i)\cdot\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)=\bar{u}(\mathbf{x}_g)-\bar{u}_i+\bar{r}_{gi},$$

$$(\mathbf{x}_j-\mathbf{x}_g)\cdot\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)=\bar{u}_j-\bar{u}(\mathbf{x}_g)+\bar{r}_{gj}$$

with

$$\bar{r}_{gi}=\frac{1}{V_i}\sum_{p=2}^{k}\sum_{q=0}^{p}N_q^p\frac{\partial^p\bar{u}}{\partial x^q\partial y^{p-q}}(\mathbf{x}_g)\int_i(x-x_g)^q(y-y_g)^{p-q}+\mathcal{O}\left(h^{k+1}\right),$$

$$\bar{r}_{gj}=-\frac{1}{V_j}\sum_{p=2}^{k}\sum_{q=0}^{p}N_q^p\frac{\partial^p\bar{u}}{\partial x^q\partial y^{p-q}}(\mathbf{x}_g)\int_j(x-x_g)^q(y-y_g)^{p-q}+\mathcal{O}\left(h^{k+1}\right)$$

Using respectively $\mathbf{x}=\mathbf{x}_r$ and $\mathbf{x}=\mathbf{x}_s$ in the Taylor expansion (4), we obtain

$$\bar{u}(\mathbf{x}_r)=\bar{u}(\mathbf{x}_g)+(\mathbf{x}_r-\mathbf{x}_g)\cdot\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)+\sum_{p=2}^{k}\sum_{q=0}^{p}N_q^p\frac{\partial^p\bar{u}}{\partial x^q\partial y^{p-q}}(\mathbf{x}_g)(x_r-x_g)^q(y_r-y_g)^{p-q}+\mathcal{O}\left(h^{k+1}\right),$$

$$\bar{u}(\mathbf{x}_s)=\bar{u}(\mathbf{x}_g)+(\mathbf{x}_s-\mathbf{x}_g)\cdot\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)+\sum_{p=2}^{k}\sum_{q=0}^{p}N_q^p\frac{\partial^p\bar{u}}{\partial x^q\partial y^{p-q}}(\mathbf{x}_g)(x_s-x_g)^q(y_s-y_g)^{p-q}+\mathcal{O}\left(h^{k+1}\right).$$

Subtracting these equalities gives

$$(\mathbf{x}_s-\mathbf{x}_r)\cdot\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)=\bar{u}(\mathbf{x}_s)-\bar{u}(\mathbf{x}_r)+\bar{r}_{rs}$$

with

$$\bar{r}_{rs}=-\sum_{p=2}^{k}\sum_{q=0}^{p}N_q^p\frac{\partial^p\bar{u}}{\partial x^q\partial y^{p-q}}(\mathbf{x}_g)\left((x_s-x_g)^q(y_s-y_g)^{p-q}-(x_r-x_g)^q(y_r-y_g)^{p-q}\right)+\mathcal{O}\left(h^{k+1}\right).$$

Thus, we have the system

$$\begin{cases} \boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)\cdot(\mathbf{x}_g-\mathbf{x}_i)=\bar{u}(\mathbf{x}_g)-\bar{u}_i+\bar{r}_{gi},\\ \boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)\cdot(\mathbf{x}_j-\mathbf{x}_g)=\bar{u}_j-\bar{u}(\mathbf{x}_g)+\bar{r}_{gj},\\ \boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)\cdot(\mathbf{x}_s-\mathbf{x}_r)=\bar{u}(\mathbf{x}_s)-\bar{u}(\mathbf{x}_r)+\bar{r}_{rs}. \end{cases} \tag{6}$$

We can decompose the unit normal vector $\mathbf{n}_{i\ell}$ both in the basis $((\mathbf{x}_g-\mathbf{x}_i),(\mathbf{x}_s-\mathbf{x}_r))$ and $((\mathbf{x}_j-\mathbf{x}_g),(\mathbf{x}_s-\mathbf{x}_r))$

$$\mathbf{n}_{i\ell}=\alpha_{gi}\frac{\mathbf{x}_g-\mathbf{x}_i}{\|\mathbf{x}_g-\mathbf{x}_i\|}+\beta_{gi}\frac{\mathbf{x}_s-\mathbf{x}_r}{\|\mathbf{x}_s-\mathbf{x}_r\|}=\alpha_{gj}\frac{\mathbf{x}_j-\mathbf{x}_g}{\|\mathbf{x}_j-\mathbf{x}_g\|}+\beta_{gj}\frac{\mathbf{x}_s-\mathbf{x}_r}{\|\mathbf{x}_s-\mathbf{x}_r\|}$$

with

$$\alpha_{gi}=\frac{\|\mathbf{x}_g-\mathbf{x}_i\|}{(\mathbf{x}_g-\mathbf{x}_i)\cdot\mathbf{n}_{i\ell}}>0,\qquad \alpha_{gj}=\frac{\|\mathbf{x}_j-\mathbf{x}_g\|}{(\mathbf{x}_j-\mathbf{x}_g)\cdot\mathbf{n}_{i\ell}}>0, \tag{7}$$

$$\beta_{gi}=\frac{\|\mathbf{x}_s-\mathbf{x}_r\|\mathbf{n}_{i\ell}\cdot(\mathbf{x}_g-\mathbf{x}_i)^{\perp}}{(\mathbf{x}_s-\mathbf{x}_r)\cdot(\mathbf{x}_g-\mathbf{x}_i)^{\perp}},\qquad \beta_{gj}=\frac{\|\mathbf{x}_s-\mathbf{x}_r\|\mathbf{n}_{i\ell}\cdot(\mathbf{x}_j-\mathbf{x}_g)^{\perp}}{(\mathbf{x}_s-\mathbf{x}_r)\cdot(\mathbf{x}_j-\mathbf{x}_g)^{\perp}}. \tag{8}$$

That is, in view of Figure 1

$$\alpha_{gi}=\frac{1}{\cos(\theta_{ig})},\qquad \beta_{gi}=\frac{\sin(\theta_{ig})}{\cos(\theta_{ig})},\qquad \alpha_{gj}=\frac{1}{\cos(\theta_{gj})},\qquad \beta_{gj}=\frac{\sin(\theta_{gj})}{\cos(\theta_{gj})}.$$

According to assumption (**H**) these values are well defined. Owing to the definition we choose for $\mathbf{x}_i$ (see Section 2 and Figure 2), we have the inequalities $\alpha_{gi}>0$, $\alpha_{gj}>0$, which are mandatory for positiveness of the scheme (see Section 4).

Thus, we have the expression of the gradient in the direction of the normal vector seen by the cell i, j, respectively denoted by $\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)_i\cdot\mathbf{n}_{i\ell}$, $\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)_j\cdot\mathbf{n}_{i\ell}$

$$\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)_i\cdot\mathbf{n}_{i\ell}=\alpha_{gi}\frac{\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)\cdot(\mathbf{x}_g-\mathbf{x}_i)}{\|\mathbf{x}_g-\mathbf{x}_i\|}+\beta_{gi}\frac{\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)\cdot(\mathbf{x}_s-\mathbf{x}_r)}{\|\mathbf{x}_s-\mathbf{x}_r\|},$$

$$\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)_j\cdot\mathbf{n}_{i\ell}=\alpha_{gj}\frac{\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)\cdot(\mathbf{x}_j-\mathbf{x}_g)}{\|\mathbf{x}_j-\mathbf{x}_g\|}+\beta_{gj}\frac{\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)\cdot(\mathbf{x}_s-\mathbf{x}_r)}{\|\mathbf{x}_s-\mathbf{x}_r\|},$$

that is to say, using (6)

$$\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)_i\cdot\mathbf{n}_{i\ell}=\alpha_{gi}\frac{\bar{u}(\mathbf{x}_g)-\bar{u}_i+\bar{r}_{gi}}{\|\mathbf{x}_g-\mathbf{x}_i\|}+\beta_{gi}\frac{\bar{u}(\mathbf{x}_s)-\bar{u}(\mathbf{x}_r)+\bar{r}_{rs}}{\|\mathbf{x}_s-\mathbf{x}_r\|}, \tag{9}$$

$$\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)_j\cdot\mathbf{n}_{i\ell}=\alpha_{gj}\frac{\bar{u}_j-\bar{u}(\mathbf{x}_g)+\bar{r}_{gj}}{\|\mathbf{x}_j-\mathbf{x}_g\|}+\beta_{gj}\frac{\bar{u}(\mathbf{x}_s)-\bar{u}(\mathbf{x}_r)+\bar{r}_{rs}}{\|\mathbf{x}_s-\mathbf{x}_r\|}, \tag{10}$$

If κ is continuous on a Gauss point $\mathbf{x}_g$ of an edge ℓ we define

$$\kappa_{gi}=\kappa_{gj}=\kappa(\mathbf{x}_g)$$

while if it is not we define

$$\kappa_{gi}=\lim_{\mathbf{x}\in i\to\mathbf{x}_g}\kappa(\mathbf{x}),\quad \kappa_{gj}=\lim_{\mathbf{x}\in j\to\mathbf{x}_g}\kappa(\mathbf{x}).$$

Thanks to the continuity of the flux

$$\kappa_{gi}\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)_i\cdot\mathbf{n}_{i\ell}=\kappa_{gj}\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)_j\cdot\mathbf{n}_{i\ell},$$

we obtain

$$\begin{aligned}\bar{u}(\mathbf{x}_g)=\frac{1}{\frac{\kappa_{gi}\alpha_{gi}}{\|\mathbf{x}_g-\mathbf{x}_i\|}+\frac{\kappa_{gj}\alpha_{gj}}{\|\mathbf{x}_j-\mathbf{x}_g\|}}\Big(&\frac{\kappa_{gj}\alpha_{gj}}{\|\mathbf{x}_j-\mathbf{x}_g\|}(\bar{u}_j+\bar{r}_{gj})+\frac{\kappa_{gi}\alpha_{gi}}{\|\mathbf{x}_g-\mathbf{x}_i\|}(\bar{u}_i-\bar{r}_{gi})\\ &+\frac{\kappa_{gj}\beta_{gj}}{\|\mathbf{x}_s-\mathbf{x}_r\|}(\bar{u}(\mathbf{x}_s)-\bar{u}(\mathbf{x}_r)+\bar{r}_{rs})-\frac{\kappa_{gi}\beta_{gi}}{\|\mathbf{x}_s-\mathbf{x}_r\|}(\bar{u}(\mathbf{x}_s)-\bar{u}(\mathbf{x}_r)+\bar{r}_{rs})\Big).\end{aligned}$$

Inserting this value into (9) or (10) results in

$$\kappa_{gi}\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)_i\cdot\mathbf{n}_{i\ell}=\kappa_{gj}\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)_j\cdot\mathbf{n}_{i\ell}=\left(\frac{\kappa_{gi}\kappa_{gj}\alpha_{gi}\alpha_{gj}}{\|\mathbf{x}_j-\mathbf{x}_g\|\kappa_{gi}\alpha_{gi}+\|\mathbf{x}_g-\mathbf{x}_i\|\kappa_{gj}\alpha_{gj}}\right)(\bar{u}_j-\bar{u}_i+\bar{r}_{gj}+\bar{r}_{gi})$$
$$+\left(\frac{\kappa_{gi}\kappa_{gj}\alpha_{gi}\beta_{gj}\|\mathbf{x}_j-\mathbf{x}_g\|}{\|\mathbf{x}_s-\mathbf{x}_r\|\left(\|\mathbf{x}_j-\mathbf{x}_g\|\kappa_{gi}\alpha_{gi}+\|\mathbf{x}_g-\mathbf{x}_i\|\kappa_{gj}\alpha_{gj}\right)}\right)(\bar{u}(\mathbf{x}_s)-\bar{u}(\mathbf{x}_r)+\bar{r}_{rs})$$
$$+\left(\frac{\kappa_{gi}\kappa_{gj}\alpha_{gj}\beta_{gi}\|\mathbf{x}_g-\mathbf{x}_i\|}{\|\mathbf{x}_s-\mathbf{x}_r\|\left(\|\mathbf{x}_j-\mathbf{x}_g\|\kappa_{gi}\alpha_{gi}+\|\mathbf{x}_g-\mathbf{x}_i\|\kappa_{gj}\alpha_{gj}\right)}\right)(\bar{u}(\mathbf{x}_s)-\bar{u}(\mathbf{x}_r)+\bar{r}_{rs}).$$

Let us assume that we have at our disposal an approximation $\mathbf{u}=(u_i)_{1\leq i\leq n}$ of $\bar{\mathbf{u}}=(\bar{u}_i)_{1\leq i\leq n}$. From $\mathbf{u}$ we can find a high-order polynomial approximation $P_i(\mathbf{x})$ of $\bar{u}$ in each cell i while respecting the discontinuity lines of the diffusion coefficient κ (see Section 3.3). So, the numerical flux $\mathcal{F}_{i\ell}(\mathbf{u})$ is defined by

$$\mathcal{F}_{i\ell}(\mathbf{u})=|\ell|\sum_{g\in\ell}\omega_g\left[\left(\frac{\kappa_{gi}\kappa_{gj}\alpha_{gi}\alpha_{gj}}{\|\mathbf{x}_j-\mathbf{x}_g\|\kappa_{gi}\alpha_{gi}+\|\mathbf{x}_g-\mathbf{x}_i\|\kappa_{gj}\alpha_{gj}}\right)(u_j-u_i+r_{gj}(\mathbf{u})+r_{gi}(\mathbf{u}))\right.$$
$$+\left(\frac{\kappa_{gi}\kappa_{gj}\alpha_{gi}\beta_{gj}\|\mathbf{x}_j-\mathbf{x}_g\|}{\|\mathbf{x}_s-\mathbf{x}_r\|\left(\|\mathbf{x}_j-\mathbf{x}_g\|\kappa_{gi}\alpha_{gi}+\|\mathbf{x}_g-\mathbf{x}_i\|\kappa_{gj}\alpha_{gj}\right)}\right)(P_j(\mathbf{x}_s)-P_j(\mathbf{x}_r)+s_{gj}(\mathbf{u}))$$
$$\left.+\left(\frac{\kappa_{gi}\kappa_{gj}\alpha_{gj}\beta_{gi}\|\mathbf{x}_g-\mathbf{x}_i\|}{\|\mathbf{x}_s-\mathbf{x}_r\|\left(\|\mathbf{x}_j-\mathbf{x}_g\|\kappa_{gi}\alpha_{gi}+\|\mathbf{x}_g-\mathbf{x}_i\|\kappa_{gj}\alpha_{gj}\right)}\right)(P_i(\mathbf{x}_s)-P_i(\mathbf{x}_r)+s_{gi}(\mathbf{u}))\right]$$

with

$$\begin{cases}r_{gi}(\mathbf{u})=\dfrac{1}{V_i}\displaystyle\sum_{p=2}^{k}\sum_{q=0}^{p}N_q^p\frac{\partial^p P_i}{\partial x^q\partial y^{p-q}}(\mathbf{x}_g)\int_i(x-x_g)^q(y-y_g)^{p-q},\\ r_{gj}(\mathbf{u})=-\dfrac{1}{V_j}\displaystyle\sum_{p=2}^{k}\sum_{q=0}^{p}N_q^p\frac{\partial^p P_j}{\partial x^q\partial y^{p-q}}(\mathbf{x}_g)\int_j(x-x_g)^q(y-y_g)^{p-q},\\ s_{gi}(\mathbf{u})=-\displaystyle\sum_{p=2}^{k}\sum_{q=0}^{p}N_q^p\frac{\partial^p P_i}{\partial x^q\partial y^{p-q}}(\mathbf{x}_g)\left((x_s-x_g)^q(y_s-y_g)^{p-q}-(x_r-x_g)^q(y_r-y_g)^{p-q}\right),\\ s_{gj}(\mathbf{u})=-\displaystyle\sum_{p=2}^{k}\sum_{q=0}^{p}N_q^p\frac{\partial^p P_j}{\partial x^q\partial y^{p-q}}(\mathbf{x}_g)\left((x_s-x_g)^q(y_s-y_g)^{p-q}-(x_r-x_g)^q(y_r-y_g)^{p-q}\right).\end{cases}\tag{11}$$

Finally we obtain in a more compact form the following approximation of the flux through the edge ℓ

$$\mathcal{F}_{i\ell}(\mathbf{u})=\gamma_\ell(u_j-u_i)+r_{i\ell}(\mathbf{u})\tag{12}$$

with

$$\begin{cases}\gamma_\ell=|\ell|\displaystyle\sum_{g\in\ell}\omega_g\left(\frac{\kappa_{gi}\kappa_{gj}\alpha_{gi}\alpha_{gj}}{\|\mathbf{x}_j-\mathbf{x}_g\|\kappa_{gi}\alpha_{gi}+\|\mathbf{x}_g-\mathbf{x}_i\|\kappa_{gj}\alpha_{gj}}\right)\geq 0,\\ r_{i\ell}(\mathbf{u})=|\ell|\displaystyle\sum_{g\in\ell}\omega_g\left[\left(\frac{\kappa_{gi}\kappa_{gj}\alpha_{gi}\alpha_{gj}}{\|\mathbf{x}_j-\mathbf{x}_g\|\kappa_{gi}\alpha_{gi}+\|\mathbf{x}_g-\mathbf{x}_i\|\kappa_{gj}\alpha_{gj}}\right)(r_{gi}(\mathbf{u})+r_{gj}(\mathbf{u}))\right.\\ \quad+\left(\frac{\kappa_{gi}\kappa_{gj}\alpha_{gj}\beta_{gi}\|\mathbf{x}_g-\mathbf{x}_i\|}{\|\mathbf{x}_s-\mathbf{x}_r\|\left(\|\mathbf{x}_j-\mathbf{x}_g\|\kappa_{gi}\alpha_{gi}+\|\mathbf{x}_g-\mathbf{x}_i\|\kappa_{gj}\alpha_{gj}\right)}\right)(P_i(\mathbf{x}_s)-P_i(\mathbf{x}_r)+s_{gi}(\mathbf{u}))\\ \quad\left.+\left(\frac{\kappa_{gi}\kappa_{gj}\alpha_{gi}\beta_{gj}\|\mathbf{x}_j-\mathbf{x}_g\|}{\|\mathbf{x}_s-\mathbf{x}_r\|\left(\|\mathbf{x}_j-\mathbf{x}_g\|\kappa_{gi}\alpha_{gi}+\|\mathbf{x}_g-\mathbf{x}_i\|\kappa_{gj}\alpha_{gj}\right)}\right)(P_j(\mathbf{x}_s)-P_j(\mathbf{x}_r)+s_{gj}(\mathbf{u}))\right].\end{cases}$$

The property $\gamma_\ell\geq 0$ is a consequence of $\omega_g\geq 0$ and $\alpha_{gi},\alpha_{gj}\geq 0$ (Equation (7)).

Remark 3.1. *The same construction can be made in the cell j. The only difference is that $\mathbf{n}_{i\ell}$ is replaced by $\mathbf{n}_{j\ell}=-\mathbf{n}_{i\ell}$. Thus, we have by construction that $r_{i\ell}=-r_{j\ell}$ and moreover that $\mathcal{F}_{i\ell}=-\mathcal{F}_{j\ell}$.*

Remark 3.2. *In the present article, we consider a scalar-valued diffusion coeffient $\boldsymbol{\kappa}$. The extension to the case of a tensor-valued diffusion coeffient $\boldsymbol{\kappa}$ is straightforward and goes as follows. First, formula* (3) *becomes*

$$-\sum_{\ell\in i}\int_\ell\boldsymbol{\nabla}\bar{u}\cdot(\boldsymbol{\kappa}^t\mathbf{n}_{i\ell})+\int_i\lambda\bar{u}=\int_i f.$$

Second, we decompose the normal vector $\boldsymbol{\kappa}^t\mathbf{n}_{i\ell}$ *both in the basis* $((\mathbf{x}_g-\mathbf{x}_i),(\mathbf{x}_s-\mathbf{x}_r))$ *and in the basis* $((\mathbf{x}_j-\mathbf{x}_g),(\mathbf{x}_s-\mathbf{x}_r))$

$$\boldsymbol{\kappa}^t\mathbf{n}_{i\ell} = \alpha_{gi}\frac{\mathbf{x}_g-\mathbf{x}_i}{\|\mathbf{x}_g-\mathbf{x}_i\|} + \beta_{gi}\frac{\mathbf{x}_s-\mathbf{x}_r}{\|\mathbf{x}_s-\mathbf{x}_r\|} = \alpha_{gj}\frac{\mathbf{x}_j-\mathbf{x}_g}{\|\mathbf{x}_j-\mathbf{x}_g\|} + \beta_{gj}\frac{\mathbf{x}_s-\mathbf{x}_r}{\|\mathbf{x}_s-\mathbf{x}_r\|}$$

Thus, the coefficient defined by (7) *and* (8) *becomes*

$$\alpha_{gi} = \frac{\|\mathbf{x}_g-\mathbf{x}_i\|\boldsymbol{\kappa}^t\mathbf{n}_{i\ell}\cdot\mathbf{n}_{i\ell}}{(\mathbf{x}_g-\mathbf{x}_i)\cdot\mathbf{n}_{i\ell}} > 0, \qquad \beta_{gi} = \frac{\|\mathbf{x}_s-\mathbf{x}_r\|\boldsymbol{\kappa}^t\mathbf{n}_{i\ell}\cdot(\mathbf{x}_g-\mathbf{x}_i)^\perp}{(\mathbf{x}_s-\mathbf{x}_r)\cdot(\mathbf{x}_g-\mathbf{x}_i)^\perp}$$

and

$$\alpha_{gj} = \frac{\|\mathbf{x}_j-\mathbf{x}_g\|\boldsymbol{\kappa}^t\mathbf{n}_{i\ell}\cdot\mathbf{n}_{i\ell}}{(\mathbf{x}_j-\mathbf{x}_g)\cdot\mathbf{n}_{i\ell}} > 0, \qquad \beta_{gj} = \frac{\|\mathbf{x}_s-\mathbf{x}_r\|\boldsymbol{\kappa}^t\mathbf{n}_{i\ell}\cdot(\mathbf{x}_j-\mathbf{x}_g)^\perp}{(\mathbf{x}_s-\mathbf{x}_r)\cdot(\mathbf{x}_j-\mathbf{x}_g)^\perp}.$$

The fact that α_{gi}, α_{gj} *are positive is a direct consequence of the ellipticity condition* (2) *satisfied by* $\boldsymbol{\kappa}$.

3.2 Approximation of the boundary fluxes

In this section we use the boundary conditions to estimate the boundary fluxes.

3.2.1 Neumann boundary condition

Integrating the Neumann boundary condition on an edge $\ell \subset \Gamma_N$, we have

$$\int_\ell \kappa\boldsymbol{\nabla}\bar{u}\cdot\mathbf{n}_{i\ell} = \int_\ell g_N,$$

that is to say

$$\bar{\mathcal{F}}_{i\ell} = |\ell|\sum_{g\in\ell}\omega_g g_N(\mathbf{x}_g) + \mathcal{O}(h^k),$$

we thus impose this equation on the numerical flux

$$\mathcal{F}_{i\ell}(\mathbf{u}) = |\ell|\sum_{g\in\ell}\omega_g g_N(\mathbf{x}_g).$$

3.2.2 Dirichlet boundary condition

Taking into account the Dirichlet boundary condition $\bar{u}(\mathbf{x}_g) = g_D(\mathbf{x}_g)$ in (9) we have, for $g \in \ell \subset \Gamma_D$,

$$\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)\cdot\mathbf{n}_{i\ell} = \alpha_{gi}\frac{g_D(\mathbf{x}_g)-\bar{u}_i+\bar{r}_{gi}}{\|\mathbf{x}_g-\mathbf{x}_i\|} + \beta_{gi}\frac{\bar{u}(\mathbf{x}_s)-\bar{u}(\mathbf{x}_r)+\bar{r}_{rs}}{\|\mathbf{x}_s-\mathbf{x}_r\|}.$$

By mimicking the expression of this exact flux, the numerical one is defined by

$$\mathcal{F}_\ell(\mathbf{u}) = |\ell|\sum_{g\in\ell}\omega_g\kappa_g\left(\frac{\alpha_{gi}}{\|\mathbf{x}_g-\mathbf{x}_i\|}\left(g_D(\mathbf{x_g})-u_i+r_{gi}(\mathbf{u})\right) + \frac{\beta_{gi}}{\|\mathbf{x}_s-\mathbf{x}_r\|}\left(P_i(\mathbf{x}_s)-P_i(\mathbf{x}_r)+s_{gi}(\mathbf{u})\right)\right)$$

with $r_{gi}(\mathbf{u})$ and $s_{gi}(\mathbf{u})$ given in (11). In a more compact form we have

$$\mathcal{F}_{i\ell}(\mathbf{u}) = -\gamma_\ell u_i + \sum_{g\in\ell}\left(\frac{\omega_g\kappa_g\alpha_{gi}|\ell|}{\|\mathbf{x}_g-\mathbf{x}_i\|}g_D(\mathbf{x_g})\right) + r_{i\ell}(\mathbf{u})$$

with

$$\begin{cases} \gamma_\ell = \displaystyle\sum_{g\in\ell}\left(\frac{\omega_g\kappa_g\alpha_{gi}|\ell|}{\|\mathbf{x}_g-\mathbf{x}_i\|}\right) \geq 0, \\ r_{i\ell}(\mathbf{u}) = |\ell|\displaystyle\sum_{g\in\ell}\omega_g\kappa_g\left(\frac{\alpha_{gi}}{\|\mathbf{x}_g-\mathbf{x}_i\|}r_{gi}(\mathbf{u}) + \frac{\beta_{gi}}{\|\mathbf{x}_s-\mathbf{x}_r\|}(P_i(\mathbf{x}_s)-P_i(\mathbf{x}_r)+r_{rs}(\mathbf{u}))\right), \end{cases}$$

where

$$r_{rs} = -\sum_{p=2}^{k}\sum_{q=0}^{p}N_q^p\frac{\partial^p P_i}{\partial x^q\partial y^{p-q}}(\mathbf{x}_g)\left((x_s-x_g)^q(y_s-y_g)^{p-q}-(x_r-x_g)^q(y_r-y_g)^{p-q}\right).$$

3.3 High-order reconstruction by interpolation

For a polynomial of degree k, we have to calculate

$$\frac{(k+1)(k+2)}{2}$$

coefficients, so at least $(k+1)(k+2)/2$ neighboring cells of the cell are required. However, it is well known [54], that a larger number of cells is highly desirable for stability purpose. Following [23, 30], we use a stencil of size $(k+1)(k+2)$, to be sure to have enough information in each direction. When it is possible, the stencil will be centered on the cell, but the closer the cell is to the boundary or the discontinuity of κ, the more the stencil will be shifted so as not to cross the discontinuity.

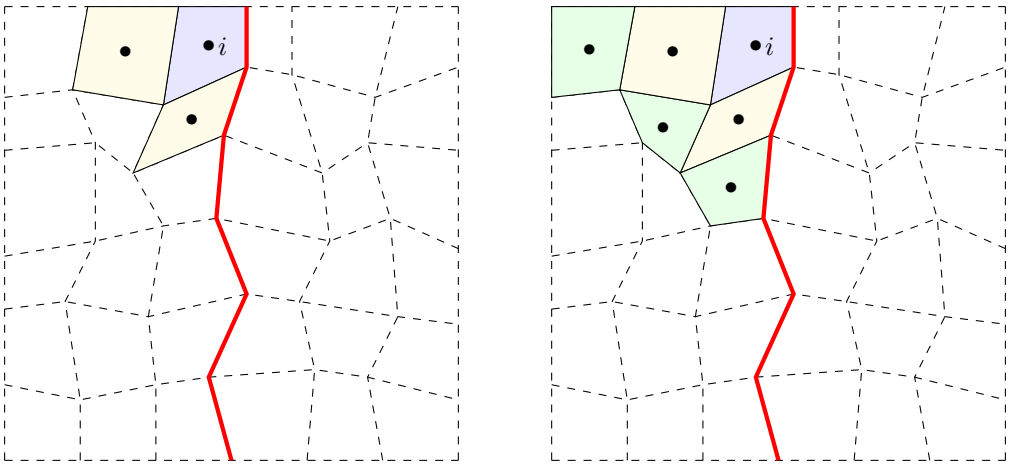


Figure 3: Construction of the stencil for the cell i with a discontinuity (in red)

To be more precise, the construction of the stencil $\mathcal{S}_i = \{0, ..., p\}$ associated with a cell i is illustrated on Figure 3. For the sake of simplicity, we have assumed that the cells involved in the stencil have been renumbered. First the cell i itself (in blue) is added to the stencil and then we add the cells that share, at least, an edge with the cell i (in yellow). If the number of cells we have already selected is not sufficient (in our case, $(k+1)(k+2)$ cells for a polynomial of order k), we add the cells that have, at least, an edge linked to the cells that we have just been added to the stencil (in green) and so on until we have enough cells. In all the above process, we impose that the stencil does not cross any discontinuity of κ (see Figure 3). Note that this method only works if there are enough cells in the domain to build the stencil. If the coefficient κ is discontinuous, there must be enough cells in all sub-domains generated by the discontinuity. When this is not the case, the only strategy we are able to propose is to refine the mesh.

Let $u_0, ..., u_p$ denote the $p+1$ values of $\mathbf{u}$ used for the calculation, with $p \geq 2$. The polynomial is of the form

$$P(\mathbf{x}) = \sum_{m=0}^{k} \sum_{n=0}^{k-m} a_{m,n}(\mathbf{u})(x - x_i)^m (y - y_i)^n.$$

The coefficients of the polynomial $P(\mathbf{x})$ are assumed to satisfy

$$\frac{1}{V_j} \int_j P(\mathbf{x}) = u_j, \quad \forall j \in \mathcal{S}_i.$$

This leads to the following system

$$\underbrace{\begin{pmatrix} 1 & \frac{1}{V_0}\int_0 (x-x_i) & \frac{1}{V_0}\int_0 (y-y_i) & \dots & \frac{1}{V_0}\int_0 (x-x_i)^k & \frac{1}{V_0}\int_0 (y-y_i)^k \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 1 & \frac{1}{V_p}\int_p (x-x_i) & \frac{1}{V_p}\int_p (y-y_i) & \dots & \frac{1}{V_p}\int_p (x-x_i)^k & \frac{1}{V_p}\int_p (y-y_i)^k \end{pmatrix}}_{=:\mathbf{M}} \underbrace{\begin{pmatrix} a_{0,0} \\ a_{1,0} \\ a_{0,1} \\ \vdots \\ a_{k,0} \\ a_{0,k} \end{pmatrix}}_{=:\mathbf{a}} = \underbrace{\begin{pmatrix} u_0 \\ \vdots \\ u_p \end{pmatrix}}_{=:\mathbf{d}}.$$

Since the matrix $\mathbf{M}$ has more rows than columns we have to use the least square method so that the vector $\mathbf{a}$ is computed as the solution to the linear system: $\mathbf{M}^t\mathbf{M}\mathbf{a} = \mathbf{M}^t\mathbf{d}$. To improve the condition-number of this Least-Squares problem, we rewrite it as

$$\begin{cases} \mathbf{M}\mathbf{G}\mathbf{y} = \mathbf{d}, \\ \mathbf{a} = \mathbf{G}\mathbf{y}, \end{cases}$$

where $\mathbf{G}$ is the diagonal matrix defined by

$$G_{kk} = \left(\frac{1}{V_i}\right)^{\frac{d_k}{2}},$$

with d_k the degree of the k-th monomial of P (see Section 2.1.4 of [15]). We use the Givens method (see [27] p.206 and following) to solve this least-square problem, which avoids the direct inversion of $\mathbf{M}^t\mathbf{M}$.

In this process, we do not enforce the continuity of u at the vertices. Indeed, a priori, $P_j(\mathbf{x}_s) \neq P_i(\mathbf{x}_s)$ for $i \neq j$.

4 Positivity preservation

A method borrowed from [25, 26, 51, 56] and developed in the framework of 2D diffusion on arbitrary meshes can be used to make the scheme positivity preserving. The flux (12) can be rewritten as follows

$$\mathcal{F}_{i\ell}(\mathbf{u}) = \gamma_\ell(u_j - u_i) + r_{i\ell}(\mathbf{u})^+ - r_{i\ell}(\mathbf{u})^-,$$

with

$$r_{i\ell}(\mathbf{u})^+ = \frac{|r_{i\ell}(\mathbf{u})| + r_{i\ell}(\mathbf{u})}{2} \geq 0 \quad \text{and} \quad r_{i\ell}(\mathbf{u})^- = \frac{|r_{i\ell}(\mathbf{u})| - r_{i\ell}(\mathbf{u})}{2} \geq 0.$$

Let us assume that $\mathbf{u} > \mathbf{0}$, the flux then reads as

$$\mathcal{F}_{i\ell}(\mathbf{u}) = \left(\gamma_\ell + \frac{r_{i\ell}(\mathbf{u})^+}{u_j}\right) u_j - \left(\gamma_\ell + \frac{r_{i\ell}(\mathbf{u})^-}{u_i}\right) u_i,$$

and the coefficients $\left(\gamma_\ell + \frac{r_{i\ell}(\mathbf{u})^+}{u_j}\right)$ and $\left(\gamma_\ell + \frac{r_{i\ell}(\mathbf{u})^-}{u_i}\right)$ are positive. We end up with a two-point flux structure, which is very favorable for the resolution of the system. However note that this system is non-symmetric and non-linear since its coefficients depend on $\mathbf{u}$.

Remark 4.1. *We have also*

$$\mathcal{F}_{j\ell}(\mathbf{u}) = \left(\gamma_\ell + \frac{r_{j\ell}(\mathbf{u})^+}{u_i}\right) u_i - \left(\gamma_\ell + \frac{r_{j\ell}(\mathbf{u})^-}{u_j}\right) u_j.$$

Using that $r_{j\ell}(\mathbf{u}) = -r_{i\ell}(\mathbf{u})$ (see Remark 3.1), it implies $r_{j\ell}(\mathbf{u})^+ = r_{i\ell}(\mathbf{u})^-$ and $r_{j\ell}(\mathbf{u})^- = r_{i\ell}(\mathbf{u})^+$. Hence,

$$\gamma_\ell + \frac{r_{i\ell}(\mathbf{u})^-}{u_i} = \gamma_\ell + \frac{r_{j\ell}(\mathbf{u})^+}{u_i} \quad , \quad \gamma_\ell + \frac{r_{i\ell}(\mathbf{u})^+}{u_j} = \gamma_\ell + \frac{r_{j\ell}(\mathbf{u})^-}{u_j}. \tag{13}$$

This yields

$$\mathcal{F}_{j\ell}(\mathbf{u}) = \left(\gamma_\ell + \frac{r_{i\ell}(\mathbf{u})^-}{u_i}\right) u_i - \left(\gamma_\ell + \frac{r_{i\ell}(\mathbf{u})^+}{u_j}\right) u_j = -\mathcal{F}_{i\ell}(\mathbf{u}).$$

This is expected since the scheme is equivalent to (12) as long as $\forall i \in [\![1, n]\!]$, $u_i \neq 0$.

The definition of the nonlinear scheme requires $u_i > 0, \forall i \in [\![1, n]\!]$, but at the limit $h \to 0$, u_i may vanish. In order to circumvent this difficulty, it is possible to add a term $\eta > 0$ to the denominator in the flux. Then, the flux is given by

$$\mathcal{F}_{i\ell}(\mathbf{u}) = -\mathcal{F}_{j\ell}(\mathbf{u}) = \left(\gamma_\ell + \frac{r_{i\ell}(\mathbf{u})^+}{u_j + \eta}\right) u_j - \left(\gamma_\ell + \frac{r_{i\ell}(\mathbf{u})^-}{u_i + \eta}\right) u_i. \tag{14}$$

In addition, relation (13) becomes

$$\gamma_\ell + \frac{r_{i\ell}(\mathbf{u})^-}{u_i + \eta} = \gamma_\ell + \frac{r_{j\ell}(\mathbf{u})^+}{u_i + \eta} \quad , \quad \gamma_\ell + \frac{r_{i\ell}(\mathbf{u})^+}{u_j + \eta} = \gamma_\ell + \frac{r_{j\ell}(\mathbf{u})^-}{u_j + \eta}. \tag{15}$$

We tried both implementations (with and without η) and found negligible differences on the results. The only sensitivity we found is that, in difficult problems, adding η may lead to a reduction of the number of fixed-point iterations. Note that in theory, η is no more negligible when $u \to 0$. However we did not observe any difference in the results even for problems with homogeneous Dirichlet boundary conditions. All the problems in Section 6 are run with $\eta = 10^{-15}$. In practice, this parameter should be scaled with respect to the problem considered.

4.1 Matrix form

The scheme reads as (recall that we have assumed that λ is constant)

$$-\sum_{\ell\in i}\mathcal{F}_{i\ell}(\mathbf{u})+\lambda V_i u_i = V_i f_i. \tag{16}$$

Consider a mesh the cells of which are numbered from 1 to n. Denoting

$$\mathbf{u}=(u_i)_{1\le i\le n},\quad \mathbf{b}=(b_i)_{1\le i\le n},\quad \mathbf{A}=(A_{ij})_{1\le i,j\le n},$$

we can write (16) as the matrix-vector product

$$\mathbf{A}(\mathbf{u})\mathbf{u}=\mathbf{b}, \tag{17}$$

with

$$\begin{cases} A_{ii}(\mathbf{u}) = \displaystyle\sum_{\ell\in i,\ell\notin\Gamma_N}\left(\gamma_\ell+\frac{r_{i\ell}(\mathbf{u})^-}{u_i+\eta}\right)+V_i\lambda, & \\ A_{ij}(\mathbf{u}) = -\displaystyle\sum_{\ell\in i\cap j}\left(\gamma_\ell+\frac{r_{i\ell}(\mathbf{u})^+}{u_j+\eta}\right) & i\neq j \end{cases} \tag{18}$$

and

$$\mathbf{b}_i = V_i f_i + \sum_{\ell\in i,\ell\in\Gamma_D}\left(r_{i\ell}(\mathbf{u})^+ + \sum_{g\in\ell}\left(\frac{\omega_g\kappa_g\alpha_{gi}|\ell|}{\|\mathbf{x}_g-\mathbf{x}_i\|}\right)g_D(\mathbf{x}_g)\right) + \sum_{\ell\in i,\ell\in\Gamma_N}|\ell|\sum_{g\in\ell}\omega_g g_N(\mathbf{x}_g). \tag{19}$$

Remark 4.2. *In* (16) *and* (19)*, an approximation of order k is required to compute $f_i=\frac{1}{V_i}\int_i f$. Moreover, the formula for the zero order term $\lambda V_i u_i$ is of order k if λ is assumed to be a constant.*
If λ is not a constant, the integral $\int_i\lambda\bar{u}$ in Equation (3) *is approximated using a k-th order accurate Gauss's quadrature formula, which gives $\sum_{\tilde{g}}\omega_{\tilde{g}}\lambda(\mathbf{x}_{\tilde{g}})\bar{u}(\mathbf{x}_{\tilde{g}})$, where $\omega_{\tilde{g}}>0$ and $\mathbf{x}_{\tilde{g}}$ are respectively the weights and the points of the quadrature in cell i. To be more precise, we proceed as follows.*
Integrating (4) *on cell i and dividing by its volume V_i gives*

$$\bar{u}_i=\bar{u}(\mathbf{x}_{\tilde{g}})+(\mathbf{x}_i-\mathbf{x}_{\tilde{g}})\cdot\boldsymbol{\nabla}\bar{u}(\mathbf{x}_{\tilde{g}})+\frac{1}{V_i}\sum_{p=2}^{k}\sum_{q=0}^{p}N_q^p\frac{\partial^p\bar{u}}{\partial x^q\partial y^{p-q}}(\mathbf{x}_{\tilde{g}})\int_i(x-x_{\tilde{g}})^q(y-y_{\tilde{g}})^{p-q}+\mathcal{O}\left(h^{k+1}\right),$$

which implies

$$\int_i\lambda\bar{u}=\sum_{\tilde{g}}\omega_{\tilde{g}}\lambda(\mathbf{x}_{\tilde{g}})\left(\bar{u}_i+\bar{r}_{i\tilde{g}}\right),$$

where

$$\bar{r}_{i\tilde{g}}=-(\mathbf{x}_i-\mathbf{x}_{\tilde{g}})\cdot\boldsymbol{\nabla}\bar{u}(\mathbf{x}_{\tilde{g}})+\frac{1}{V_i}\sum_{p=2}^{k}\sum_{q=0}^{p}N_q^p\frac{\partial^p\bar{u}}{\partial x^q\partial y^{p-q}}(\mathbf{x}_{\tilde{g}})\int_i(x-x_{\tilde{g}})^q(y-y_{\tilde{g}})^{p-q}+\mathcal{O}\left(h^{k+1}\right).$$

The scheme thus reads as

$$\sum_{\ell\in i}\left(\gamma_\ell(u_i-u_j)+r_{i\ell}(\mathbf{u})\right)+\sum_{\tilde{g}}\omega_{\tilde{g}}\lambda(\mathbf{x}_{\tilde{g}})\left(u_i+r_{i\tilde{g}}\right)=V_i f_i,$$

where $r_{i\ell}(\mathbf{u})$ is defined in Section 3.1, and

$$r_{i\tilde{g}}=-(\mathbf{x}_i-\mathbf{x}_{\tilde{g}})\cdot\boldsymbol{\nabla}P_i(\mathbf{x}_{\tilde{g}})+\frac{1}{V_i}\sum_{p=2}^{k}\sum_{q=0}^{p}N_q^p\frac{\partial^p P_i}{\partial x^q\partial y^{p-q}}(\mathbf{x}_{\tilde{g}})\int_i(x-x_{\tilde{g}})^q(y-y_{\tilde{g}})^{p-q}.$$

The above equation is valid for inner cells only, but can be easily adapted for boundary cells, as we explained in Section 3.2.

Remark 4.3. *Assuming that $f\ge 0$ and $g\ge 0$, all the components of the right hand side $\mathbf{b}$ are non-negative. Assuming moreover that f and g are not both identically zero, then at least one component of $\mathbf{b}$ is positive.*

4.2 Picard iteration method

In order to solve (17) we use a Picard iteration method. We start with an initial guess $\mathbf{u}^0 > 0$, compute the matrix $\mathbf{A}(\mathbf{u}^0)$ and solve $\mathbf{A}(\mathbf{u}^0)\mathbf{u}^1 = \mathbf{b}$. Repeating this process, we build a sequence $(\mathbf{u}^\nu)$ that, if it converges to a positive vector, tends to a solution of the scheme. We stop the algorithm when the difference $\mathbf{u}^{\nu+1} - \mathbf{u}^\nu$ between two successive iterates is small enough. To summarize, the following algorithm is used

$$\begin{aligned} &\nu = 0 \\ &A(\mathbf{u}^0)\mathbf{u}^1 = \mathbf{b} \\ &\text{While } \frac{\|\mathbf{u}^{\nu+1} - \mathbf{u}^\nu\|_2}{\|\mathbf{u}^\nu\|_2} > \varepsilon \\ &\quad A(\mathbf{u}^\nu)\mathbf{u}^{\nu+1} = \mathbf{b} \\ &\quad \nu = \nu + 1, \end{aligned} \tag{20}$$

where ε is a small stopping criterium. Unfortunately, we are unable to prove that the above algorithm converges. Nevertheless, we prove in Section 5.1 below that the scheme is well defined at each iteration of the algorithm, as soon as the initial guess $\mathbf{u}^0$ is positive.

This procedure implies that the flux depends now on $\mathbf{u}^\nu$ and $\mathbf{u}^{\nu+1}$. Accordingly, we define

$$\mathcal{F}_{i\ell}(\mathbf{u}^\nu, \mathbf{u}^{\nu+1}) = \left(\gamma_\ell + \frac{r_{i\ell}(\mathbf{u}^\nu)^+}{u_j^\nu + \eta}\right) u_j^{\nu+1} - \left(\gamma_\ell + \frac{r_{i\ell}(\mathbf{u}^\nu)^-}{u_i^\nu + \eta}\right) u_i^{\nu+1}. \tag{21}$$

Note that relation (15) still holds, because the coefficients depend only on $\mathbf{u}^\nu$, hence

$$\mathcal{F}_{i\ell}(\mathbf{u}^\nu, \mathbf{u}^{\nu+1}) = -\mathcal{F}_{j\ell}(\mathbf{u}^\nu, \mathbf{u}^{\nu+1}). \tag{22}$$

5 Properties

5.1 Well-posedness of the Picard iteration method

Consider the definition of an M-matrix (see for instance [43])

Definition 5.1. *An $n \times n$ matrix $\mathbf{A}$ that can be expressed in the forme $\mathbf{A} = s\mathbf{I} - \mathbf{B}$, where $\mathbf{B} = (b_{ij})_{1\le i,j\le n}$ with $b_{ij} \ge 0$, $1 \le i, j \le n$, and $s \ge \rho(\mathbf{B})$, the maximum of the moduli of the eigenvalues of $\boldsymbol{B}$, is called an M-matrix.*

We use the following lemma

Lemma 5.2. *A matrix $\mathbf{A} = (A_{ij})_{1\le i,j\le n}$ is an M-matrix if it satisfies the following inequalities*

$$\forall i \ne j, \quad A_{ij} \le 0, \quad \text{and} \quad \forall i, \quad \sum_{j=1}^n A_{ij} \ge 0.$$

Moreover, if the last inequality is strict, we say that $\mathbf{A}$ is a strict M-matrix.

Proposition 5.3. *Assume that $\mathbf{u} > \mathbf{0}$. Then the matrice $\mathbf{A}$ defined by (18) is such that $\mathbf{A}^t$ is a strict M-matrix.*

Proof. The matrix $\mathbf{A}$ satisfies

$$\forall i \ne j, \quad A_{ij} \le 0 \quad \text{and} \quad \forall j, \quad \sum_{i=1}^n A_{ij} > 0.$$

Indeed we have, for all j

$$\sum_{i=1}^n A_{ij} = \sum_{i=1}^n \left(\sum_{\ell\in i, \ell\notin\Gamma_N} \left(\gamma_\ell + \frac{r_{i\ell}(\mathbf{u})^-}{u_i+\eta}\right) - \sum_{\ell\in i\cap j} \left(\gamma_\ell + \frac{r_{i\ell}(\mathbf{u})^+}{u_j+\eta}\right)\right) + \lambda_j V_j.$$

Thanks to the relation (15), only the boundary terms and the mass term remain, for all j

$$\sum_{i=1}^n A_{ij} = \sum_{i=1}^n \sum_{\ell\in(i\cap\Gamma_D)} \left(\gamma_\ell + \frac{r_{i\ell}(\mathbf{u})^-}{u_i+\eta}\right) + \lambda_j V_j > 0.$$

□

Proposition 5.4. *Assume that $f \ge 0$, $g \ge 0$, and either $\|f\|_{L^2(\Omega)} > 0$ or $\|g\|_{L^2(\partial\Omega)} > 0$. Assume moreover that $\mathbf{u}^0 > \mathbf{0}$. Then, the algorithm (20) defines a sequence $(\mathbf{u}^\nu)_{\nu\ge0}$ such, that for all ν, $\mathbf{u}^\nu > \mathbf{0}$.*

To prove this property, we need to introduce the concept of irreducible matrix: see [48, Definition 1.15].

Definition 5.5. *An $n \times n$ matrix $\mathbf{A}$ is **reducible** if there exits an $n \times n$ permutation matrix $\mathbf{P}$ such that*

$$\mathbf{P}\mathbf{A}\mathbf{P}^t = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{A}_{22} \end{bmatrix},$$

*where $\mathbf{A}_{11}$, $\mathbf{A}_{12}$, $\mathbf{A}_{22}$ are respectively $r \times r$, $r \times (n-r)$ and $(n-r) \times (n-r)$ sub-matrices with $1 \leq r < n$. If no such permutation matrix exists, then $\mathbf{A}$ is **irreducible**.*

The matrice $\mathbf{A}$ defined by (18) is irreducible thanks to the following Lemma (see [48, Theorem 1.17]).

Lemma 5.6. *To any $n \times n$ matrix $\mathbf{A}$ we associate the graph of nodes $1, 2, ..., n$ and of directed edges connecting $\mathbf{x}_i$ to $\mathbf{x}_j$ if $A_{ij} \neq 0$. Then $\mathbf{A}$ is irreducible if and only if for any pair $i \neq j$ there exists a chain of edges that allows to go from $\mathbf{x}_i$ to $\mathbf{x}_j$,*

$$A_{i,k1} \neq 0 \rightarrow A_{k1,k2} \neq 0 \rightarrow \cdots \rightarrow A_{km,j} \neq 0.$$

With these definitions we can make use of the following theorem (see [48], Corollary 3.20).

Theorem 5.7. *If $\mathbf{A}$ is an irreducible strict M-matrix, then it is invertible and, for all i, j ($1 \leq i, j \leq n$), $(A^{-1})_{ij} > 0$.*

We are now in position to prove Proposition 5.4.

Proof of Proposition 5.4. We argue by induction on the index ν. We assume that $\mathbf{u}^\nu > 0$. Hence $(\mathbf{A}(\mathbf{u}^\nu))^t$ is a strict M-matrix (see Proposition 5.3). It is easy to check that $(\mathbf{A}(\mathbf{u}^\nu))^t$ is also irreducible. Thus, applying Theorem 5.7, $(\mathbf{A}(\mathbf{u}^\nu))^t$ is invertible and all the entries of $(\mathbf{A}(\mathbf{u}^\nu))^{-t}$ are positive. Consequently, all the entries of $(\mathbf{A}(\mathbf{u}^\nu))^{-1}$ are positive. Using Remark 4.3, we know that all components of $\mathbf{b}$ are non-negative. Moreover, because of the assumption that either $\|f\|_{L^2(\Omega)} > 0$ or $\|g\|_{L^2(\partial\Omega)} > 0$, at least one component of $\mathbf{b}$ is positive. We thus have, for all i ($1 \leq i \leq n$)

$$u_i^{\nu+1} = \sum_{j=1}^{n} (\mathbf{A}(\mathbf{u}^\nu))_{ij}^{-1} b_j > 0,$$

since all terms of this sum are non-negative, with one at least that does not vanish. □

Proposition 5.4 shows that the condition $\mathbf{u}^\nu > \mathbf{0}$ remains satisfied during the Picard iteration method, which allows to define $\mathbf{A}(\mathbf{u}^\nu)$ for all $\nu \geq 0$.

Remark 5.8. *The scheme preserves positivity if the inversion of the linear system is exact. The above proof assumes that the solution of the system $\mathbf{A}\mathbf{u} = \mathbf{b}$ is calculated exactly. Obviously, in practice, this is not the case, hence, the numerical solution may be negative. In the tests we have carried out, the error is small enough not to produce negative-valued solution. However, in rare cases, the inversion of the system led to a solution with negative components, causing the calculation to stop (see for example Section 6.2.1 below). This error can be reduced by working on the condition number of the matrix or on methods for solving linear systems, which is a perspective. Besides, these issues are related to the implementation (and not the scheme itself). Further work is required to improve robustness.*

5.2 Conservation

Proposition 5.9. *Assume that $\mathbf{u}^0 > \mathbf{0}$ and consider homogeneous Neumann boundary conditions, then the scheme defined by (16) is conservative at each fixed-point iteration, that is to say*

$$\forall \nu, \qquad \sum_{i=1}^{n} V_i \lambda u_i^{\nu+1} = \sum_{i=1}^{n} V_i f_i.$$

Proof. Owing to Proposition 5.4, we know that $\forall \nu$, $\mathbf{u}^\nu$ is well defined and positive. The sum can be rewritten by inverting the sum on the cells and on the faces. Besides, the sum can be separated into boundary terms and non-boundary-terms

$$\sum_{i=1}^{n} \left(-\sum_{\ell \in i} \mathcal{F}_{i\ell}(\mathbf{u}^\nu, \mathbf{u}^{\nu+1}) \right) = -\sum_{\ell \in \Gamma} \mathcal{F}_{i\ell}(\mathbf{u}^\nu, \mathbf{u}^{\nu+1}) - \sum_{\ell \notin \Gamma} \left(\mathcal{F}_{i\ell}(\mathbf{u}^\nu, \mathbf{u}^{\nu+1}) + \mathcal{F}_{j\ell}(\mathbf{u}^\nu, \mathbf{u}^{\nu+1}) \right).$$

According to (22), we have

$$\mathcal{F}_{i\ell}(\mathbf{u}^\nu, \mathbf{u}^{\nu+1}) + \mathcal{F}_{j\ell}(\mathbf{u}^\nu, \mathbf{u}^{\nu+1}) = 0,$$

for inner faces. In addition,

$$\mathcal{F}_{i\ell}(\mathbf{u}^\nu, \mathbf{u}^{\nu+1}) = 0,$$

for boundary faces. Thus, the scheme is conservative at each fixed-point iteration. □

6 Numerical experiments

Given $\Omega =]0,1[^2$, κ a diffusion coefficient and g a function defined on $\partial\Omega$, consider Problem (1) with $\lambda = 0$ and $\Gamma_N = \emptyset$

$$\begin{cases} -\boldsymbol{\nabla}\cdot(\kappa\boldsymbol{\nabla}\bar{u}) = f & \text{in} \quad \Omega, \\ \bar{u} = g & \text{on} \quad \partial\Omega. \end{cases} \tag{23}$$

In addition to Cartesian meshes we will use the two following types of meshes (see Figure 4):

1. deformed meshes, the deformation of which from the Cartesian mesh is given by

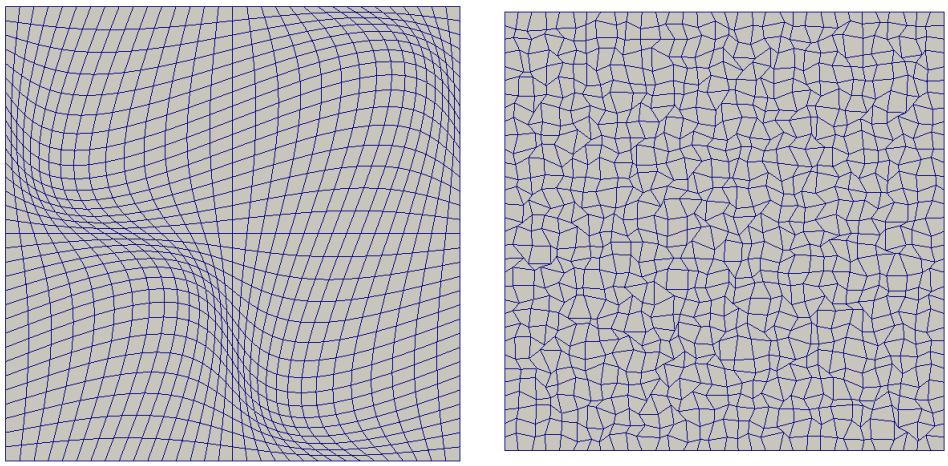
$$(x,y) \to (x + 0.1\sin(2\pi x)\sin(2\pi y), y + 0.1\sin(2\pi x)\sin(2\pi y)),$$

2. randomly deformed meshes, the deformation of which from the unit Cartesian mesh with cells of size Δx is given by

$$(x,y) \to 0.1(x,y) + 0.9(x + 0.45a\Delta x, y + 0.45b\Delta x),$$

where a, b are random numbers distributed according to the uniform law on $[-1,1]$.

In the tests, we use a sequence of successively refined deformed or randomly deformed meshes. In such a process, the deformations are applied independently on each level of refinement.



(a) A deformed mesh (b) A randomly deformed mesh

Figure 4: Examples of deformed meshes.

The L^2-error on the solution and the L^2-error on the fluxes used in the following tests are respectively given by

$$\frac{\|\mathbf{u}-\bar{\mathbf{u}}\|_2}{\|\bar{\mathbf{u}}\|_2}, \quad \frac{\left(\sum_\ell \left(\mathcal{F}_{i\ell}(\mathbf{u}) - |\ell|\sum_{g\in\ell}\omega_g\kappa_g\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)\cdot\mathbf{n}_{i\ell}\right)^2\right)^{1/2}}{\left(\sum_\ell \left(|\ell|\sum_{g\in\ell}\omega_g\kappa_g\boldsymbol{\nabla}\bar{u}(\mathbf{x}_g)\cdot\mathbf{n}_{i\ell}\right)^2\right)^{1/2}},$$

where we recall that $\bar{\mathbf{u}}$ is defined by (5). Numerically, an approximation of order k is required to compute $\bar{u}_i$.

We also use the H^1 semi-norm error defined by

$$\frac{\|\boldsymbol{\nabla}_h\mathbf{u} - \boldsymbol{\nabla}\bar{u}\|_2}{\|\boldsymbol{\nabla}\bar{u}\|_2},$$

where

$$\|\boldsymbol{\nabla}\bar{u}\|_2 = \left(\sum_i V_i \|\boldsymbol{\nabla}\bar{u}(\mathbf{x}_i)\|^2\right)^{1/2}, \quad \|\boldsymbol{\nabla}_h\mathbf{u} - \boldsymbol{\nabla}\bar{u}\|_2 = \left(\sum_i V_i \|\boldsymbol{\nabla}P_i(\mathbf{x}_i) - \boldsymbol{\nabla}\bar{u}(\mathbf{x}_i)\|^2\right)^{1/2},$$

P_i being the polynomial obtained by reconstruction with the approximated values of the solution $\mathbf{u}$.

For all the tests, the stopping criterion ε and the initial guess $\mathbf{u}^0$ of the fixed-point algorithm (20) are $\varepsilon = 10^{-12}$ and $u_i^0 = 1, \forall i$. We use the linear solver GMRES with the preconditioner ILU (see [38], Chapter 7.4) with the convergence criterion is 10^{-14}.

6.1 Numerical accuracy assessment

In this section we present numerical results for diffusion problems of type (23) with analytical solutions. The first (resp. second) case involves a discontinuous (resp. anisotropic) diffusion coefficient. Numerical convergence rates are evaluated using the L^2 norm of the solution as well the L^2 norm of the fluxes and the H^1 semi-norm. We perform a convergence study for these problems with a sequence of successively refined deformed meshes as that shown in Figure 4a. For the sake of brevity we present only the results on this type of mesh. We obtain similar results on randomly deformed meshes as that shown on Figure 4b. We will also skip the case of continuous scalar diffusion coefficient, as it is simpler than the discontinuous and anisotropic cases. We present some tests with Dirichlet boundary conditions and $\lambda = 0$ but we obtained similar results with Neumann boundary conditions and/or $\lambda \neq 0$.

6.1.1 Discontinuous diffusion coefficient

Recall that we have assumed the possible discontinuities of the diffusion coefficient κ coincide with edges of the mesh. Given

$$\kappa(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad x \leq \dfrac{1}{2} \\ 2 & \text{if} \quad x > \dfrac{1}{2} \end{cases}, \quad f(\mathbf{x}) = 2\pi^2 \cos(\pi x)\cos(\pi y) + 20, \quad g(\mathbf{x}) = 0,$$

the function

$$\bar{u}(\mathbf{x}) = \begin{cases} \cos(\pi x)\cos(\pi y) - 10x^2 + 12 & \text{if} \quad x \leq \dfrac{1}{2}, \\ \dfrac{1}{2}\cos(\pi x)\cos(\pi y) - 5x^2 + \dfrac{43}{4} & \text{if} \quad x > \dfrac{1}{2}, \end{cases}$$

is solution to (23). Results are summarized in Figure 5 which shows that all schemes are k-th-order accurate in the L^2 norm of the solution as well the L^2 norm of the fluxes and the H^1 semi-norm. We can note that there is a superconvergence for odd orders.

We see that, even if $\nabla\bar{u}$ is discontinuous in this problem, we are able to achieve an arbitrary order of accuracy. The point for this is to design a stencil that do not cross discontinuities of κ, as explained in Section 3.3.

6.1.2 Anisotropic diffusion coefficient

Given

$$\boldsymbol{\kappa}(\mathbf{x}) = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \quad f(\mathbf{x}) = 3\pi^2 \sin(\pi x)\sin(\pi y), \quad g(\mathbf{x}) = 0,$$

the function $\bar{u}(\mathbf{x}) = \sin(\pi x)\sin(\pi y)$ is solution to (23). Results are summarized in Figure 6 which shows that all schemes are k-th-order accurate in the L^2 norm, the L^2 norm of the fluxes and the H^1 semi-norm. We can note that there is a superconvergence for odd orders. Of course, similar results have been obtained for a scalar-valued diffusion coefficient κ.

Table 1 (resp. Table 2) gives the minimum number of cells per direction required to achieve an accuracy of 10^{-5} (resp. 10^{-9}) on the L^2-error, with the number of iterations of the fixed point algorithm and the time of execution. As expected, the number of cells needed to achieve the desired precision (first column) is a decreasing function of the order. The second column gives the number of fixed point iterations required to satisfy the stagnation criterion. This number has the same order of magnitude whatever the order. It tends to be decreasing with respect to the order k for small values of k, then increases again. This may sound counter-intuitive but it is a good point. The more interesting column is the last one giving the total computational cost of the method. This computational time is a trade-off between the algorithmic complexity and the precision of the method, which both increase with the order. We notice that, in general, execution time decreases as the order increases. For a large error setpoint value (10^{-5}), the optimal choice of scheme is the third-order one. However, when decreasing the error setpoint value (10^{-9}) higher-order schemes perform better, and the optimal order becomes seven. Note that, in Table 2, we have omitted the first and second order lines, because it is too demanding in computational resources to achieve this precision. We anticipate that small values of the error setpoint will favor the

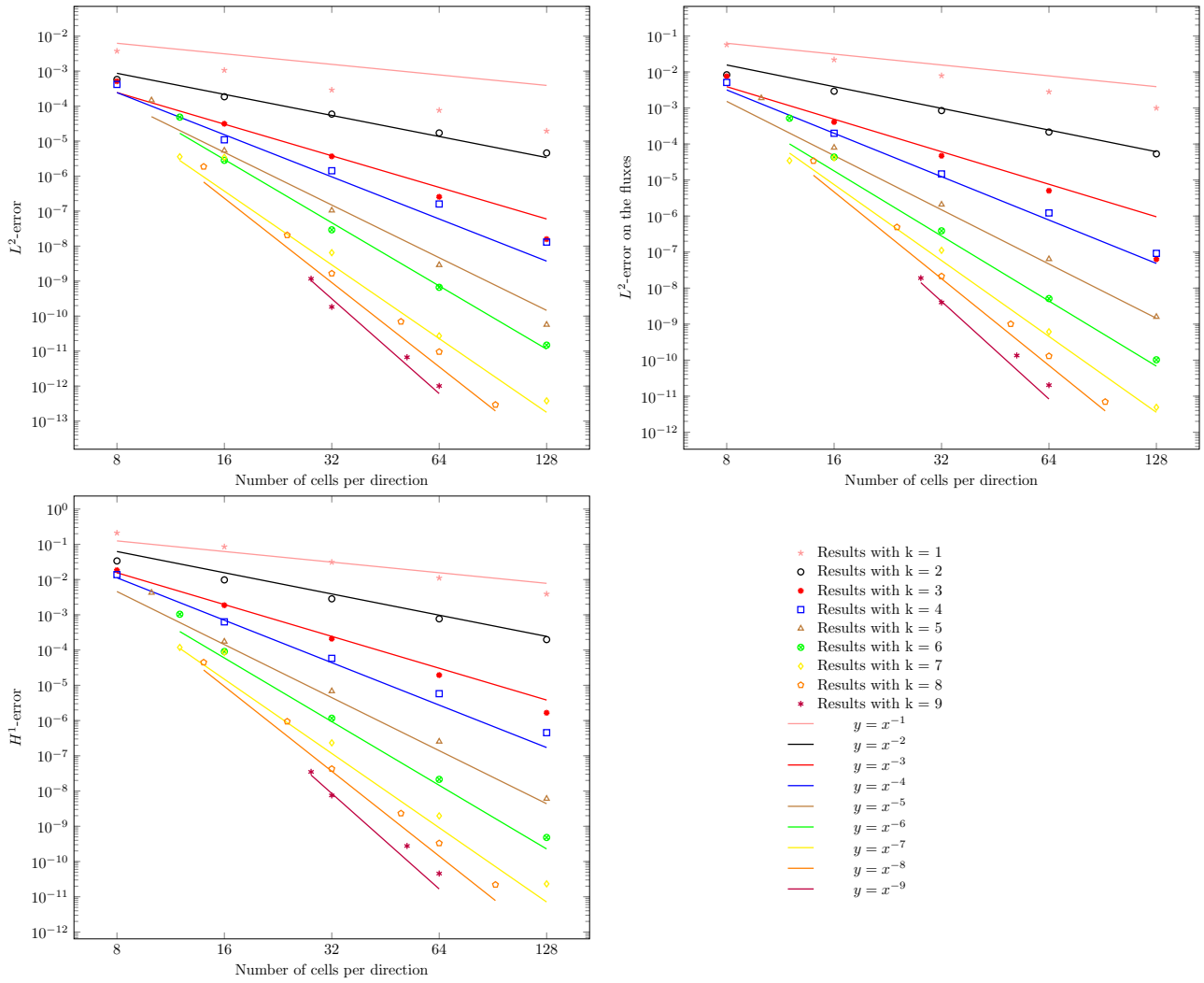


Figure 5: L^2-error on the solution (top left), L^2-error on the fluxes (top right) and H^1 semi-norm error (bottom left) for problem of Section 6.1.1.

Scheme	Number of cells per direction	Number of iterations	Execution time (ratio)
k= 1	168	98	1.00
k= 2	212	116	2.61
k= 3	31	61	0.08
k= 4	31	56	0.16
k= 5	19	46	0.16
k= 6	14	57	0.21
k= 7	16	70	0.94
k= 8	10	80	0.70

Table 1: Minimum number of cells to reach a precision on the L^2-error of 10^{-5} with the time of execution and the number of iterations of the fixed point algorithm for order 1 to 8 for problem of Section 6.1.2.

highest orders. We obtain speed-ups of factors up to ten in term of computational time to reach the desired precision. We also observed that odd orders perform better than even orders. This confirms what we notice on Figures 5 and 6: a super-convergence is achieved for odd orders. We also observe that for a fixed mesh size, the error decreases as k grows.

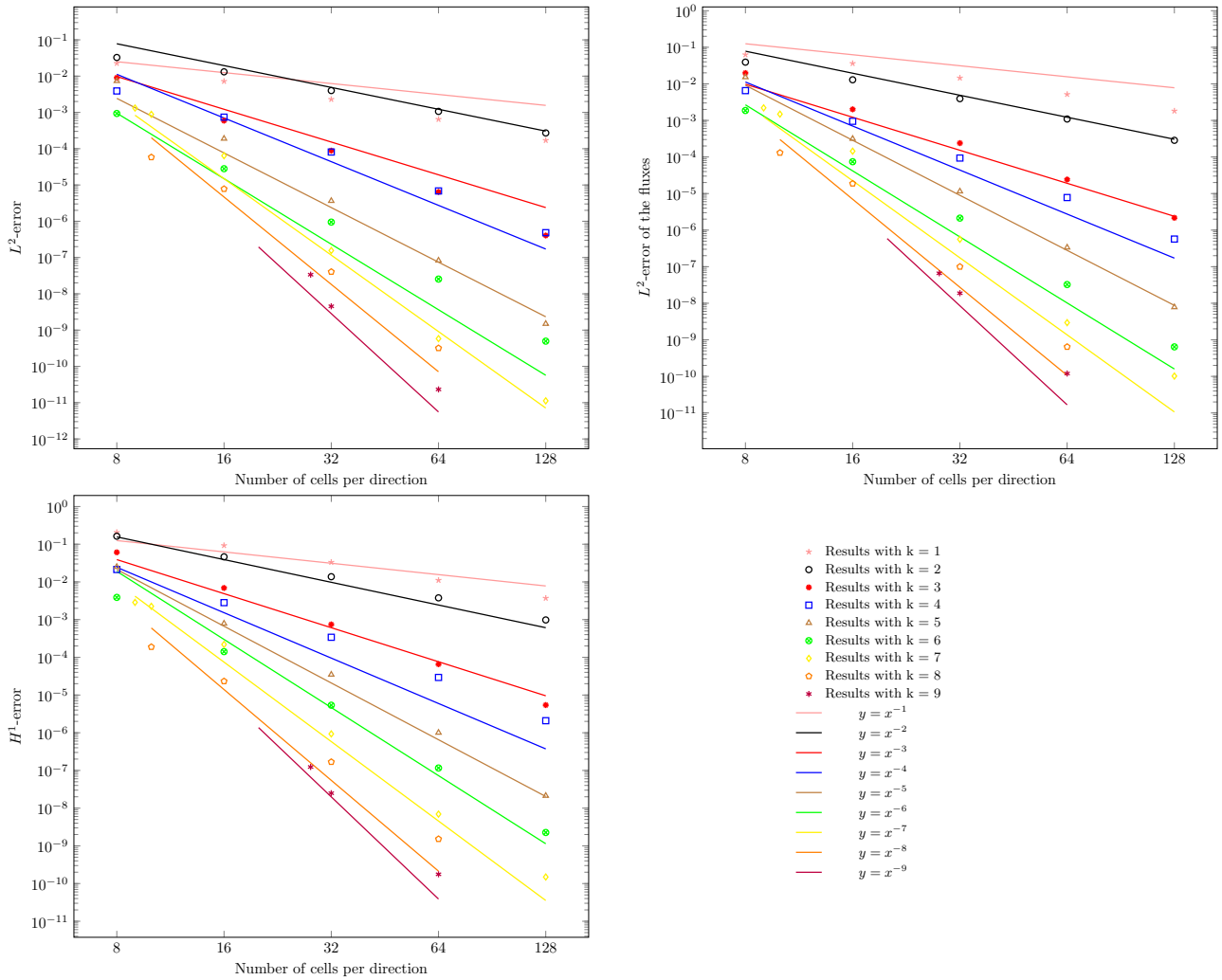


Figure 6: L^2-error on the solution (top left), L^2-error on the fluxes (top right) and H^1 semi-norm error (bottom left) for problem of Section 6.1.2.

Scheme	Number of cells per direction	Number of iterations	Execution time (ratio)
k= 3	323	135	1.00
k= 4	343	135	2.50
k= 5	93	122	0.55
k= 6	76	134	0.75
k= 7	46	90	0.54
k= 8	40	76	0.64
k= 9	30	75	0.76

Table 2: Minimum number of cells to reach a precision on the L^2-error of 10^{-9} with the time of execution and the number of iterations of the fixed point algorithm for order 3 to 9 for problem of Section 6.1.2.

Table 3 gives the L^2-error on a deformed mesh of 32×32 cells with the time of execution and the number of iterations of the fixed point algorithm for order 1 to 9. As expected, the L^2-error is globally a decreasing function of the order (with some exceptions for even orders), while execution time increases with the order. Besides, the number of iterations of the fixed point algorithm increases with the order but at a very slow rate. We can see that, for this mesh, increasing

Scheme	L^2-error	Number of iterations	Execution time (ratio)
k = 1	2.30×10^{-3}	46	1.00
k = 2	4.00×10^{-3}	55	1.62
k = 3	8.81×10^{-5}	62	4.69
k = 4	8.10×10^{-5}	58	9.45
k = 5	3.65×10^{-6}	63	33.96
k = 6	9.47×10^{-7}	67	66.80
k = 7	1.56×10^{-7}	70	208.45
k = 8	4.05×10^{-8}	67	361.07
k = 9	4.53×10^{-9}	78	936.57

Table 3: L^2-error on a deformed mesh of 32×32 cells with the time of execution and the number of iterations of the fixed point algorithm for order 1 to 9 for problem of Section 6.1.2.

the order quickly reduces the error.

6.2 Positivity preserving assessment

We propose a challenging benchmark borrowed from [53] to compare a non-positivity preserving scheme, which can give nonpositive solutions (in this case the usual DDFV scheme), with our positivity preserving high-order scheme which always gives nonnegative solutions. For this test we have used Cartesian meshes.

6.2.1 Tensor-valued coefficient κ and square domain with a square hole

Consider the square domain with a square hole $\Omega =]0,1[^2 \setminus \left[\frac{4}{9}, \frac{5}{9}\right]^2$, $f(\mathbf{x}) = 0$ in Ω and $g(\mathbf{x}) = 0$ (resp. $g(\mathbf{x}) = 2$) on the external (resp. internal) boundary. We choose

$$\boldsymbol{\kappa} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 10^4 \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \quad \theta = \frac{\pi}{6}.$$

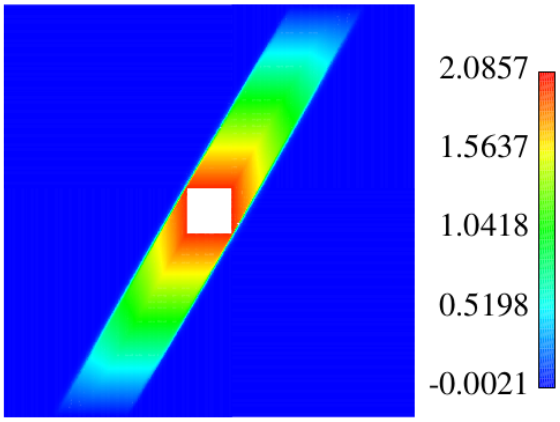


Figure 7: Numerical solution obtained with the DDFV scheme on a highly refined mesh (1310720 cells of size $\Delta x = 1/1152$).

We compare the results obtained with the positivity preserving high-order schemes on a Cartesian mesh with 2000 cells of size $\Delta x = 1/45$. The stopping criterion of the fixed point algorithm is $\varepsilon = 10^{-12}$, except for order 6 for which $\varepsilon = 10^{-10}$ and for order 7, 8 for which $\varepsilon = 10^{-6}$ to reduce the computing time.

As explained in Remark 5.8, the precision of the inversion of the linear system sometimes leads to negative entries in the solution vector $\mathbf{u}$. In general, this can be fixed by using the result of a low-order calculation as the initial guess of the high-order calculation. This procedure is also favorable regarding the computation time. It significantly reduces the

Positivity preserving scheme	Minimum	Maximum
Order 1	3.5×10^{-28}	1.96
Order 2	1.8×10^{-21}	1.96
Order 3	5.8×10^{-27}	1.98
Order 4	1.3×10^{-29}	1.97
Order 5	4.1×10^{-27}	1.97
Order 6	1.4×10^{-26}	1.98
Order 7	2.9×10^{-24}	1.98
Order 8	3.4×10^{-19}	1.98

Table 4: Minimum and maximum of the numerical solution to the problem of section 6.2.1 for a Cartesian mesh with 2000 cells of size $\Delta x = 1/45$.

overall cost of the simulation. However, we encountered one case for which this fix was not sufficient. For the test of order 5, for a Cartesian mesh with 86 cells per direction, we did not manage to run the simulation. We think that this is a severe issue for this kind of methods which is in general not addressed in the papers. However, as suggested by a reviewer, adding a time evolution term to our problem, and achieving the steady state of the problem

$$\begin{cases} \dfrac{\partial \bar{u}}{\partial t} - \boldsymbol{\nabla} \cdot (\boldsymbol{\kappa} \boldsymbol{\nabla} \bar{u}) = 0 & \text{in} \quad \Omega \times [0,T], \\ \bar{u} = g_D & \text{on} \quad \partial\Omega \times [0,T], \\ \bar{u}(0) = 1 & \text{in} \quad \Omega, \end{cases}$$

instead of solving (1) with a small enough time step, solves our convergence issue. This trick allows us to complete all the simulations we run. An example is shown in Figure 9. However, we are not able to predict *a priori* the time step required to complete the simulation. This is why, in the near future, we plane to work on the linear system inversion.

Even for a highly refined mesh (1310720 squares of size $\Delta x = 1/1152$) the solution obtained with the usual (non-positivity preserving) DDFV scheme (see Figure 7) has negative values up to -2.1×10^{-3}. On the other hand the high-order solutions obtained with the positivity preserving scheme remain always positive whatever the order: see Figure 8 and Table 4 which gives the minimum and the maximum of each solution calculated with a Cartesian mesh (2000 cells of size 1/45), up to order 6. We also observe on Figures 8 and 9 that the solution for $k = 3$ is closer to the converged solution (see Figure 7) than the solution for $k = 1$. This is reminiscent of the convergence with respect to the order we pointed out in Section 6.1.

6.2.2 Fokker-Planck type diffusion equation

This benchmark is a simplified version of the one from [32]. Given $\Omega =]-50, 50[^2$ and $T = 250$, we are looking for the function $\bar{u} = \bar{u}(\mathbf{x}, t)$, solution to the *simplified* Fokker-Planck equation[1]

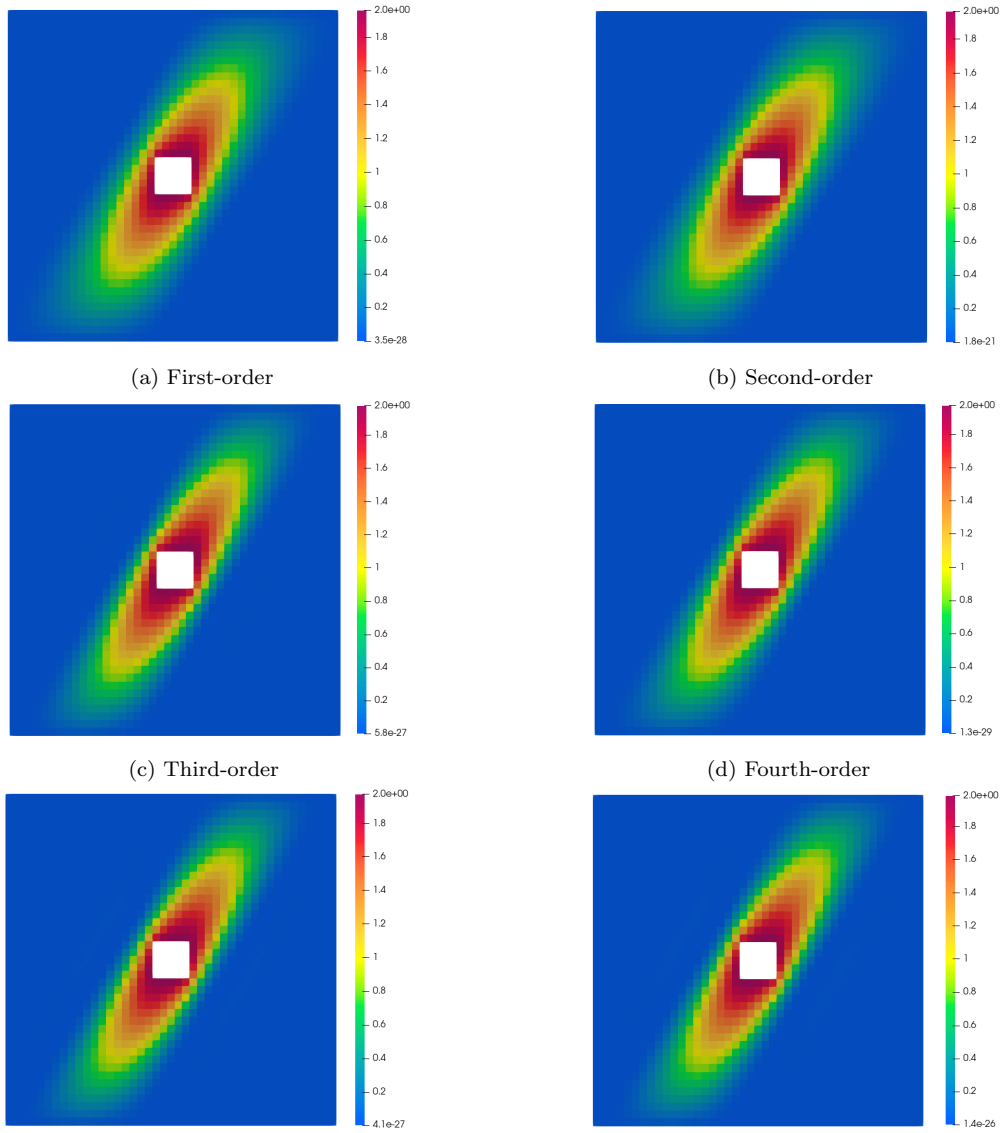
$$\begin{cases} \dfrac{\partial \bar{u}}{\partial t} - \boldsymbol{\nabla} \cdot (\boldsymbol{\kappa} \boldsymbol{\nabla} \bar{u}) = 0 & \text{in} \quad \Omega \times [0,T], \\ \boldsymbol{\kappa} \boldsymbol{\nabla} \bar{u} \cdot \mathbf{n} = 0 & \text{on} \quad \partial\Omega \times [0,T], \\ \bar{u}(0) = \bar{u}^0 & \text{in} \quad \Omega, \end{cases} \tag{24}$$

where the diffusion coefficient $\boldsymbol{\kappa} = \boldsymbol{\kappa}(\mathbf{x})$ and the initial condition $\bar{u}^0$ are given by

$$\boldsymbol{\kappa}(\mathbf{x}) = \mathbf{I} - \frac{1}{\|\mathbf{x}\|^2} \mathbf{x} \otimes \mathbf{x}, \quad \bar{u}^0(\mathbf{x}) = \frac{1}{\pi} \exp(-\|\mathbf{x} - \mathbf{x}_o\|^2)$$

[1] The *full* Fokker-Planck equation would read as

$$\frac{\partial \bar{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\mathbf{x} \bar{u}) - \boldsymbol{\nabla} \cdot (\boldsymbol{\kappa} \boldsymbol{\nabla} \bar{u}) = 0.$$



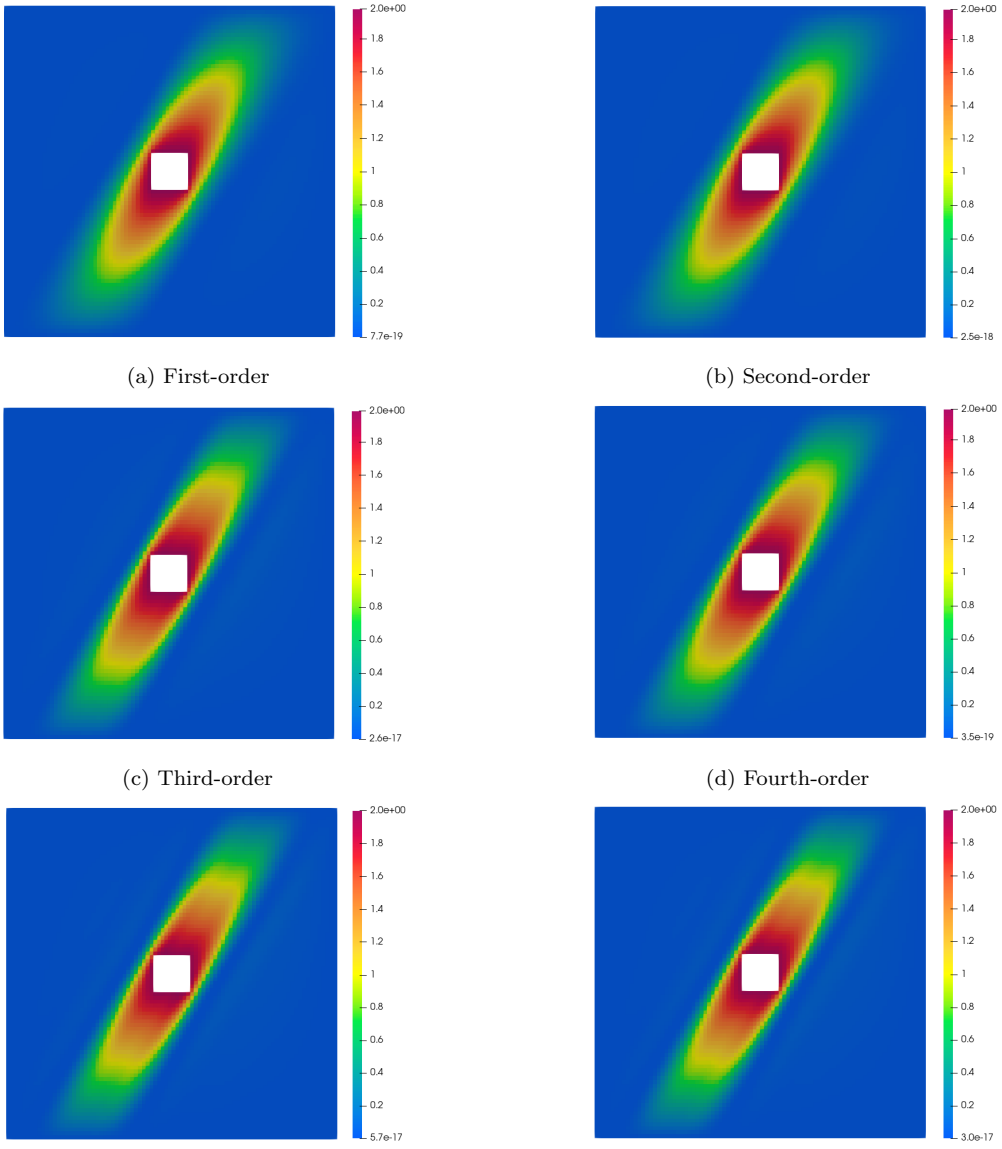
(a) First-order (b) Second-order

(c) Third-order (d) Fourth-order

(e) Fifth-order (f) Sixth-order

Figure 8: Numerical solution to the problem of section 6.2.1 obtained with positivity preserving schemes of order 1 to 6 for a Cartesian mesh (2000 cells of size 1/45).

with $\mathbf{x}_o = (-20, 20)$. Note that $\boldsymbol{\kappa}$ is degenerated: it does not satisfy (2), hence the theoretical results of the preceding Sections do not apply to the present case. It follows in particular that the well-posedness of the fixed-point algorithm (see Section 5.1) is no longer ensured. However, $\bar{u}$ should remain positive, and the non-positivity preserving DDFV scheme produces non-physical negative values. We will see that our positivity preserving scheme fixes it. This diffusion tensor correspond to a degenerate diffusion problem along the circle of radius $20\sqrt{2}$. The existence and uniqueness of a solution in $W^{1,\infty}([0\ T], W^{2,\infty}(\Omega))$ is nevertheless proven in this context: see [33, 29, 41].
The backward Euler scheme is used for time integration. To limit the calculation time, the stopping criterion of the fixed point algorithm is $\varepsilon = 10^{-5}$. Figure 11 displays the numerical solutions obtained with the Cartesian mesh of 200^2 cells. Table 5 gives the minima and maxima of the DDFV solution and the numerical solution obtained with the positivity preserving schemes up to order 6. We observe that the minima of the solutions to positivity preserving schemes always remain non negative, as expected. Compared to the solutions obtained with the DDFV scheme, given by Figures 10 and the solutions obtained by the positivity preserving DDFV schemes, given in [7], the positivity preserving arbitrary order schemes are more diffusive. However, we can note that the higher is the order, the less diffusive is the scheme.



(a) First-order (b) Second-order

(c) Third-order (d) Fourth-order

(e) Fifth-order (f) Sixth-order

Figure 9: Numerical solutions obtained with positivity preserving schemes of order 1 to 6 for a Cartesian mesh (cells of size 1/90). Simulations at order 5 and 6 are obtained in reaching the steady state of an unsteady problem.

7 Concluding remarks

This paper proposes an arbitrary-order positivity preserving Finite Volume scheme for the elliptic problem (1) on general 2D meshes. The new non-linear method we have detailed here is arbitrary-order convergent even for anisotropic and/or discontinuous diffusion coefficients on deformed meshes. Furthermore it allows to deal with all boundary conditions (Dirichlet, Neumann). This scheme uses a polynomial reconstruction involving values in neighboring cells to evaluate the secondary unknowns at the Gauss quadrature points. We have adapted the non-linear process from [51] to enforce positivity preservation. We have assessed numerically both its accuracy and positivity preservation.

Numerical performance could be improved. Indeed, the convergence of the fixed-point algorithm is not guaranteed and may be very slow. This is observed in particular in test cases where the classical DDFV scheme gives negative solutions. Techniques for accelerating this fixed point could be explored, such as Anderson acceleration (see [1, 45]) or the ϵ-algorithm (see [9, 10]). We highlight the fact that these issues are related to the implementation (and not the scheme itself). Further work is required to improve robustness.

The next step is to extend the method to non-linear diffusion (with a diffusion coefficient depending on the unknown)

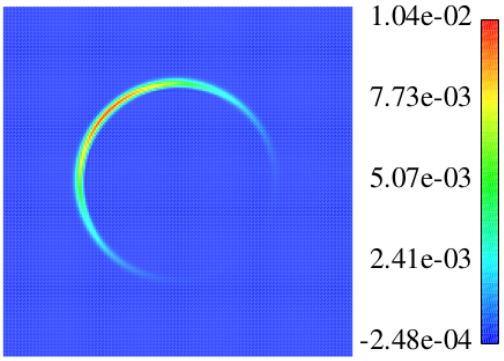


Figure 10: DDFV solution to problem (24) at time $T = 250$ on the 200×200 cells Cartesian mesh.

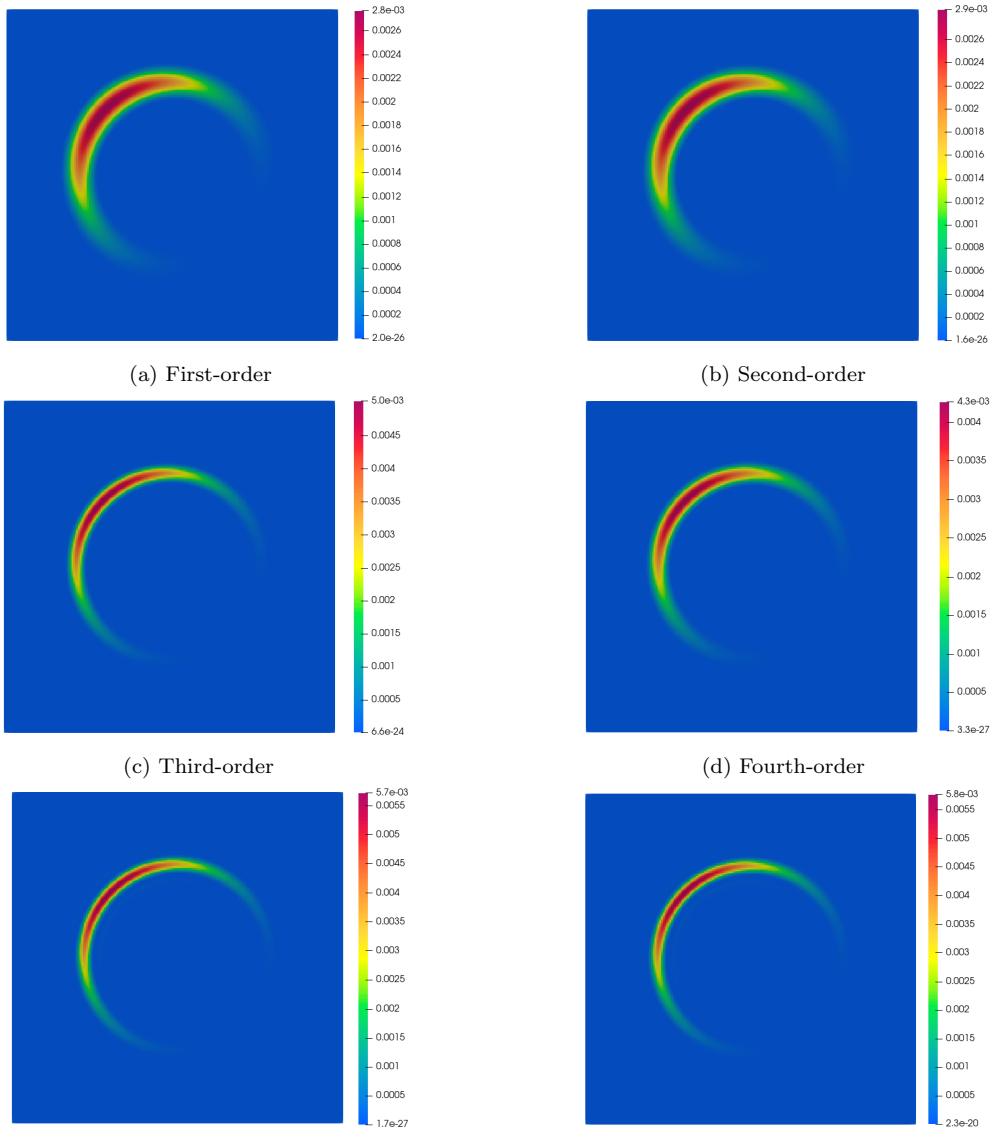
Scheme	Minimum of the solution	Maximum of the solution
DDFV	-2.48×10^{-4}	1.04×10^{-2}
Positivity preserving order 1	2.0×10^{-26}	0.28×10^{-2}
Positivity preserving order 2	1.6×10^{-26}	0.29×10^{-2}
Positivity preserving order 3	6.6×10^{-24}	0.50×10^{-2}
Positivity preserving order 4	3.2×10^{-27}	0.43×10^{-2}
Positivity preserving order 5	1.7×10^{-27}	0.57×10^{-2}
Positivity preserving order 6	2.3×10^{-20}	0.58×10^{-2}

Table 5: Minimum and maximum of the solution to the problem of section 6.2.2 at time $T = 250$ on the 200×200 cells Cartesian mesh.

and to arbitrary order unsteady diffusion, taking inspiration from [24] for example. The extension of the scheme to the three-dimensional case, based on the same ideas, is the subject of ongoing works.

8 Acknowledgements

We thank the anonymous reviewers for pointing out recent relevant references on positivity preserving methods, and suggesting improvements of the approach we propose. Thanks are also due to S. Del Pino for his help and support.



(a) First-order (b) Second-order

(c) Third-order (d) Fourth-order

(e) Fifth-order (f) Sixth-order

Figure 11: Numerical solutions to problem (24) at time $T = 250$ obtained with positivity preserving schemes of order 1 to 6 for a Cartesian mesh with 200 cells per direction

References

[1] D. G. M. Anderson. Iterative procedures for nonlinear integral equations. *J. ACM*, 12:547–560, 1965.

[2] G. Barrenechea, V. John, and P. Knobloch. An algebraic flux correction scheme satisfying the discrete maximum principle and linearity preservation on general meshes. *Math. Mod. Meth. Appl. Sci.*, 27(03):525–548, 2017.

[3] G. Barrenechea, V. John, and P. Knobloch. Finite element methods respecting the discrete maximum principle for convection-diffusion equations. *arXiv preprint arXiv:2204.07480*, 2023.

[4] L. Beirão da Veiga, F. Brezzi, L. D. Marini, and A. Russo. Virtual element method for general second-order elliptic problems on polygonal meshes. *Math. Mod. Meth. Appl. Sci.*, 26(04):729–750, 2016.

[5] E. Bertolazzi and G. Manzini. A second-order maximum principle preserving finite volume method for steady convection-diffusion problems. *SIAM J. Numer. Anal.*, 43(5):2172–2199, 2005.

[6] X. Blanc, F. Hermeline, E. Labourasse, and J. Patela. Arbitrary-order monotonic finite-volume schemes for 1D elliptic problems. *Comp. Appl. Math.*, 42(4):195, 2023.

[7] X. Blanc, F. Hermeline, E. Labourasse, and J. Patela. Monotonic diamond and DDFV type finite-volume schemes for 2D elliptic problems. *Communications in computational physics*, 34:456–502, 2023.

[8] F. Bonizzoni, M. Braukhoff, A. Jungel, and I. Perugia. A structure-preserving discontinuous Galerkin scheme for the Fischer-KPP equation. *Numerische Mathematik*, 146:119–157, 2020.

[9] C. Brezinski. *Accélération de la convergence en analyse numérique.* Lecture notes in mathematics. Springer-Verlag, 1977.

[10] C. Brezinski. *Algorithmes d'accélération de la convergence: étude numérique.* Collection Langages et algorithmes de l'informatique. Technip, 1978.

[11] E. Burman and A. Ern. Discrete maximum principle for Galerkin approximations of the Laplace operator on arbitrary meshes. *C. R. Math.*, 338(8):641–646, 2004.

[12] J.-S. Camier and F. Hermeline. A monotone nonlinear finite volume method for approximating diffusion operators on general meshes. *Int. J. Numer. Meth. Engng*, 107:496–519, 2016.

[13] C. Cancès and C. Guichard. Numerical analysis of a robust free energy diminishing finite volume scheme for parabolic equations with gradient structure. *Found. Comput. Math.*, 17:1525–1584, 2017.

[14] G. Carré, S. Del Pino, B. Després, and E. Labourasse. A cell-centered Lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. *J. Comput. Phys.*, 228(14):5160–5183, 2009.

[15] G. Carré, S. Del Pino, K. Pichon Gostaf, E. Labourasse, and A. Shapeev. Polynomial Least-Squares reconstruction for semi-Lagrangian Cell-Centered Hydrodynamic Schemes. *ESAIM-Proc*, 28:100–116, 2009.

[16] F. Cavalli, G. Naldi, and I. Perugia. Discontinuous Galerkin Approximation of Relaxation Models for Linear and Nonlinear Diffusion Equations. *SIAM Journal on Scientific Computing*, 34(1):A105–A136, 2012.

[17] P. Ciarlet. *The Finite Element Method for elliptic problems*, volume 40. SIAM, Philadelphia, 2002.

[18] S. Clain, G. Machado, J. Nóbrega, and R. Pereira. A sixth-order finite volume method for multidomain convection–diffusion problem with discontinuous coefficients. *Computer Methods in Applied Mechanics and Engineering*, 267:43–64, 2013.

[19] B. Cockburn, G. E. Karniadakis, and C-W. Shu. *Discontinuous Galerkin methods: theory, computation and applications*, volume 11. Springer Science & Business Media, 2012.

[20] M. Corti, F. Bonizzoni, and P. F. Antonietti. Structure preserving polytopal discontinuous Galerkin methods for the numerical modelling of neurodegenerative diseases. *arXiv preprint arXiv:2308.00547v1*, 2023.

[21] Y. Coudière and R. Turpault. Very high order finite volume methods for cardiac electrophysiology. *Computers & Mathematics with Applications*, 74(4):684–700, 2017.

[22] D. A. Di Pietro and J. Droniou. *The Hybrid High-Order method for polytopal meshes*, volume 19. Springer, 2020.

[23] M. Dumbser, W. Boscheri, M. Semplice, and G. Russo. Central weighted ENO schemes for hyperbolic conservation laws on fixed and moving unstructured meshes. *SIAM J. Sci. Comput.*, 39(6):A2564–A2591, 2017.

[24] A. Ern and J.-L. Guermond. Invariant-Domain Preserving High-Order Time Stepping: II. IMEX Schemes. *SIAM Journal on Scientific Computing*, 45(5):A2511–A2538, 2023.

[25] Y. Gao, G. Yuan, S. Wang, and X. Hang. A finite volume element scheme with a monotonicity correction for anisotropic diffusion problems on general quadrilateral meshes. *J. Comput. Phys.*, 407:109143, 2020.

[26] Z. Gao and J. Wu. A second-order positivity-preserving finite volume scheme for diffusion equations on general meshes. *SIAM J. Sci. Comput.*, 37(1):A420–A438, 2015.

[27] G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd Ed.).* Johns Hopkins University Press, USA, 1996.

[28] L. Guo and Y. Yang. Positivity preserving high-order local discontinuous Galerkin method for parabolic equations with blow-up solutions. *Journal of Computational Physics*, 289:181–195, 2015.

[29] K. Igari. Degenerate parabolic differential equations. *Publ. Res. Inst. Math. Sci.*, 9:493–504, 1973/74.

[30] M. Käser and A. Iske. ADER schemes on adaptive triangular meshes for scalar conservation laws. *J. Comput. Phys.*, 205(2):486–508, 2005.

[31] E. Labourasse. *Contribution to the numerical simulation of radiative hydrodynamics.* Habilitation à diriger des recherches, Sorbonne university, December 2021.

[32] O. Larroche. An efficient explicit numerical scheme for diffusion-type equations with a highly inhomogeneous and highly anisotropic diffusion tensor. *J. Comput. Phys.*, 223:436–450, 2007.

[33] C. Le Bris and P.-L. Lions. *Parabolic Equations with Irregular Data and Related Issues.* De Gruyter, Berlin, Boston, 2019.

[34] C. Le Potier. Schéma volumes finis monotone pour des opérateurs de diffusion fortement anisotropes sur des maillages de triangles non structurés. *C. R. Math.*, 341(12):787–792, 2005.

[35] H. Liu and Z. Wang. An entropy satisfying discontinuous Galerkin method for nonlinear Fokker-planck equations. *J. Sci. Comput.*, 68(05):1217–1240, 2016.

[36] H. Liu and H. Yu. Maximum-principle-satisfying third order discontinuous Galerkin schemes for Fokker-Planck equations. *SIAM J. Sci. Comput.*, 36(05):A2296–A2325, 2014.

[37] P.-H. Maire. A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. *J. Comput. Phys.*, 228(7):2391–2425, 2009.

[38] G. Meurant. *Computer solution of large linear systems*. Elsevier, 1999.

[39] C. Miranda. *Partial differential equations of elliptic type*, volume 2. Springer Science & Business Media, 2012.

[40] J. Moatti. A skeletal high-order-structure preserving scheme for advection-diffusion equations. *arXiv preprint arXiv:2303.12062v1*, 2023.

[41] O. A. Oleĭnik. On the smoothness of solutions of degenerating elliptic and parabolic equations. *Dokl. Akad. Nauk SSSR*, 163:577–580, 1965.

[42] C. Ollivier-Gooch and V. A. , Michael. A high-order-accurate unstructured mesh finite-volume scheme for the advection–diffusion equation. *Journal of Computational Physics*, 181(2):729–752, 2002.

[43] R.J. Plemmons. M-matrix characterizations.I – nonsingular M-matrices. *Linear Algebra and its Applications*, 18(2):175 – 188, 1977.

[44] E. H. Quenjel. Enhanced positive vertex-centered finite volume scheme for anisotropic convection-diffusion equations. *ESAIM, Math. Model. Numer. Anal.*, 54(2):591–618, 2020.

[45] L. G. Rebholz and M. Xiao. The Effect of Anderson Acceleration on Superlinear and Sublinear Convergence. *J. Sci. Comput.*, 96, 2023.

[46] Z. Sheng and G. Yuan. A new nonlinear finite volume scheme preserving positivity for diffusion equations. *J. Comput. Phys.*, 315:182–193, 2016.

[47] Z. Sun, J. Carrillo, and C.-W. Shu. A discontinuous Galerkin method for nonlinear parabolic equations and gradient flow problems with interaction potentials. *Journal of Computational Physics*, 352:76–104, 2018.

[48] R. S. Varga. *Matrix iterative analysis*, volume 1. Prentice Hall, 1962.

[49] J. Wang, Z. Sheng, and G. Yuan. A finite volume scheme preserving maximum principle with cell-centered and vertex unknowns for diffusion equations on distorted meshes. *Appl. Math. Comput.*, 398(1):1–21, 2021.

[50] S. Wang and G. Yuan. Discrete strong extremum principles for finite element solutions of diffusion problems with nonlinear corrections. *Appl. Numer. Math.*, 174:1–16, 2022.

[51] J. Wu and Z. Gao. Interpolation-based second-order monotone finite volume schemes for anisotropic diffusion equations on general grids. *J. Comput. Phys.*, 275:569–588, 2014.

[52] H. Yang, B. Yu, Y. Li, and G. Yuan. Monotonicity correction for second order element finite volume methods of anisotropic diffusion problems. *J. Comput. Phys.*, 449:110759, 2022.

[53] G. Yuan and Z. Sheng. Monotone finite volume schemes for diffusion equations on polygonal meshes. *J. Comput. Phys.*, 227(12):6288–6312, 2008.

[54] R. Zangeneh and C. Ollivier-Gooch. Stability analysis and improvement of the solution reconstruction for cell-centered finite volume methods on unstructured meshes. *Journal of Computational Physics*, 393:375–405, 2019.

[55] X. Zhang and C.-W. Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. *J. Comput. Phys.*, 229(9):3091 – 3120, 2010.

[56] X. Zhang, S. Su, and J. Wu. A vertex-centered and positivity-preserving scheme for anisotropic diffusion problems on arbitrary polygonal grids. *J. Comput. Phys.*, 344:419–436, 2017.

[57] F. Zhao, Z. Sheng, and G. Yuan. A monotone combination scheme of diffusion equations on polygonal meshes. *Z. Angew. Math. Mech.*, 100(5):1–25, 2020.