OPT-*i*
An International Conference on
Engineering and Applied Sciences Optimization
M. Papadrakakis, M.G. Karlaftis, N.D. Lagaros (eds.)
Kos Island, Greece, 4-6, June 2014

ALLEVIATING ADJOINT SOLVER ROBUSTNESS ISSUES VIA MIMETIC CFD DISCRETIZATION SCHEMES

Mattia Oriani[1,2], **Guillaume Pierrot**[2]

[1]Queen Mary, University of London
327 Mile End Road, E1 4NS London, UK
e-mail: mattia.oriani@esi-group.com

[2] ESI Group
Parc d'Affaires SILIC, 99 Rue des Solets, BP 80112, 94513 Rungis cedex, France
e-mail: guillaume.pierrot@esi-group.com

Keywords: Adjoint, Gradient-Based Optimization, Mimetic Finite Differences, Mixed Finite Volumes, Virtual Element Methods, Anisotropic Diffusion, Convection-Diffusion-Reaction

Abstract. *Adjoint methods allow to compute the gradient of a cost function at an expense that is independent of the number of design variables, thus representing an appealing tool to those who deal with large-scale gradient-based optimization processes. However, researchers in the field are currently facing difficulties related to poor convergence and high storage memory requirements.*

Both continuous and discrete adjoints require a fully converged solution to the primal problem. It is argued that, specifically in Computational Fluid Dynamics, the presence of numerical adjustments (e.g. non-orthogonal correctors) and segregated solution algorithms yields a solution that, while acceptable as a mere aerodynamics study, is not accurate enough to produce a robust adjoint system.

In the past decade some new PDE discretization schemes have emerged aiming to remove some constraints imposed by classical finite volumes. The main goal is to allow for more freedom in both the physical model (e.g. strong anisotropy in a material property) and its corresponding numerical model (e.g. the possibility to use polyhedral meshes with strongly non-orthogonal and/or non-convex cells), while improving solution accuracy and convergence properties at the same time.

One of such new schemes is known as Mimetic Finite Differences; the present work introduces a specific implementation of it and extends the scheme to cater for convection-diffusion-reaction problems. Preliminary numerical results are also included.

1 INTRODUCTION

In the past decade some new PDE discretization schemes have emerged aiming to remove some constraints imposed by classical Finite Volumes (FV). The main goal is to allow for more freedom in both the physical model (e.g. strong anisotropy in a material property) and its corresponding numerical model, while improving solution accuracy and convergence properties at the same time.

One increasingly popular example of such new schemes is known as Mimetic Finite Differences (MFD), recently recast as (mixed) Virtual Element Method [8], and it aims at defining discrete operators that preserve the fundamental properties of the underlying physical and mathematical models. In its early developments the MFD method was aimed at discretizing the pure anisotropic diffusion equation [5, 6, 11, 14, 16, 17]; its potential in this case has been largely validated on a number of 2D polygonal and 3D polyhedral meshes, with stress on the fact that the requirements on mesh regularity are minimal compared to traditional FV. Thus we can, for instance, solve on unstructured polyhedral meshes featuring non-orthogonal, strongly skewed, non-convex and non-conforming elements; furthermore we can solve problems presenting discontinuities in material properties. Such freedom makes MFD appealing in a number of applications, most notably the modeling of geological layers in reservoirs, magnetostatic fields [19] or flow through porous media.

Recent attempts have also been made to extend MFD to convection-diffusion-type problems [13, 21, 22] and, subsequently, to the Navier-Stokes equations [12], thus making it a promising alternative CFD tool; in this context the method is often referred to as Mixed Finite Volumes (MFV), as it incorporates some elements from traditional FV.

In the context of optimization, remarks have been made on how MFD is a promising technology in e.g. control [2] or shape optimization [3] problems, where the presence of mesh-distorting or mesh-adapting algorithms easily produces grids that cannot be handled by classical FV, thus requiring time-consuming re-meshing processes that cannot always be automated.

We claim here that MFD can help solve some of the issues affecting adjoint-based CFD optimization, as well: a better-converged and more consistent solution to the primal problem may lead to a more robust expression of the adjoint equation (either discrete or continuous); it would be equally interesting to attempt an MFV discretization of the continuous adjoint equation. In the present work we go over the main concepts of MFD and present our own extension of the scheme to cater for convection-diffusion-reaction problems; we then show and comment on a few numerical results.

2 THE PURE ANISOTROPIC DIFFUSION CASE

As we said, early developments of MFD aimed at defining a novel discretization for the *pure anisotropic diffusion* equation. We provide in this section a brief overview of the main concepts. We start with the diffusion equation for a scalar p:

$$\nabla \cdot (-\mathbb{K}\nabla p) = f \tag{2.1}$$

with $\mathbb{K}$ being, in general, a full symmetric tensor describing the anisotropic diffusivity of the material. This can be written as a system of two first-order equations (*mixed formulation*):

$$\begin{aligned} \vec{V} &= -\mathbb{K}\nabla p \\ \nabla \cdot \vec{V} &= f \end{aligned} \tag{2.2}$$

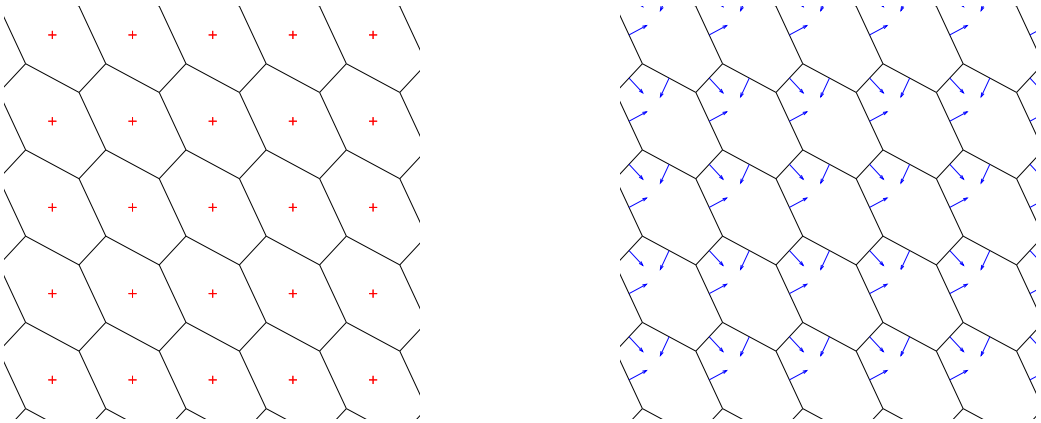
where the first is a constitutive law relating p to the velocity field $\vec{V}$ and the second describes mass conservation.

2.1 Finite spaces and scalar products

We proceed to define, over a solution domain Ω, two discrete solution spaces. If d is the number of geometric dimensions of Ω, then we have:

- d-forms: q_C = a cell-centered scalar value for each cell C
- $(d-1)$-forms: $W_{F \leftarrow C}$ = a face-centered, face oriented vector value for each face $F \in \partial C$

We shall name their corresponding finite dimensional spaces Q^h and X^h, respectively, borrowing notation from [6]; they are graphically represented in Fig.1. Thus Q^h scales with the no. of



(a) d-forms (b) $(d-1)$-forms

Figure 1: DOF location of discrete spaces Q^h and X^h for a 2D mesh

cells in Ω_h, and X^h with the no. of boundary faces + twice the no. of internal faces. However we explicitly state from now that $(d-1)$-forms must be conservative across each face:

$$W_{F \leftarrow C+} + W_{F \leftarrow C-} = 0$$

i.e.

$$s_{F,C+} W_F + s_{F,C-} W_F = 0 \tag{2.3}$$

where F is the common face between cells $C+$ and $C-$ and $s_{F,C}$ the sign, assumed fixed once and for all, defining the cell-face ordering between C and F; this allows to reduce the size of X^h to the no. of faces in Ω_h. Intuitively, the interpolation operator from $L^1(\Omega)$ onto Q^h is defined as:

$$q_C := \frac{1}{|C|} \int_C q \, dV \qquad \forall C \in \Omega_h \qquad \forall q \in L^1(\Omega) \tag{2.4}$$

where $|C|$ is the cell volume, while for X^h whe have:

$$W_{F \leftarrow C} := \int_F \vec{W} \cdot \vec{n}_F \, d\Sigma \qquad \forall F \in \Omega_h \qquad \forall \vec{W} \in H(\mathrm{div}, \Omega) \tag{2.5}$$

where $\vec{n}_F$ is the unit vector normal to F and outward w.r.t. C. The basic idea in mimetic schemes consists in equipping Q^h and X^h with scalar (inner) products that *mimic* certain key properties of their continuous counterparts, and then base our discrete operators on such scalar products.

Following [6], we start by defining the inner product for discrete scalar variables q_C, which is fairly straight-forward:

$$[p, q]_{C,Q^h} := p_C q_C |C| \qquad \forall p, q \in L^1(\Omega) \tag{2.6}$$

For the vector-valued unknowns in X^h we seek a scalar product in the form:

$$[\vec{V}, \vec{W}]_{C,X^h} := (\mathbb{B}_C(V_{F\leftarrow C})_{F\in\partial C}, (W_{F\leftarrow C})_{F\in\partial C}) \qquad \forall \vec{V}, \vec{W} \in H(\text{div}, \Omega) \tag{2.7}$$

Here, and in the sequel, $(\cdot)_{F\in\partial C}$ denotes a vector of size k_C (number of faces delimiting cell C) with its i^{th} entry being a value related to the i^{th} face of C (intended as a local index); $(\cdot, \cdot)$ denotes the standard dot product.

The matrix $\mathbb{B}_C$, evidently of size $k_C \times k_C$, must be defined for each cell in the mesh. The choice of $\mathbb{B}_C$ is the core of mimetic methods and arguably the most difficult task. Several approaches exist: some derive equations for the matrix coefficients based on mere algebraic properties of the scalar product (e.g. [6, 19]), others define local discrete operators first and then deduce the matrix from those (e.g. [11, 12]). However, the common point is that we want the inner product (2.7) to approximate the continuous one with sufficient accuracy. In particular, we ask for the Gauss-Green theorem:

$$\int_C \vec{W} \cdot \nabla q \, dV + \int_C q \nabla \cdot \vec{W} \, dV = \int_{\partial C} q \vec{W} \cdot \vec{n} \, d\Sigma \tag{2.8}$$

to be satisfied in the discrete spaces, i.e.:

$$[\vec{W}, -\mathcal{G}^h q]_{C,X^h} + [q, \mathcal{D}^h \vec{W}]_{C,Q^h} = \sum_{F\in\partial C} q_F W_{F\leftarrow C} \qquad \forall C \in \Omega_h \tag{2.9}$$

where $\mathcal{D}^h$ and $\mathcal{G}^h$ are the (not yet defined) discrete $(\nabla\cdot)$ and $(-\nabla)$ operators, and q_F is a discrete value of scalar q located at the centre of face F based on the interpolation:

$$q_F := \frac{1}{|F|} \int_F q \, d\Sigma \qquad \forall F \in \Omega_h \qquad \forall q \in L^1(\Omega) \tag{2.10}$$

with $|F|$ being the face area.

2.2 Definition of a divergence and flux operator

We choose here to follow an approach similar to that found in [17], where we exploit the inner products (2.6) and (2.7) on each cell in order to build operators. Indeed, let us first define a discrete divergence operator (quite simply a discrete version of the Gauss divergence theorem):

$$\mathcal{D}^h \vec{W} = \frac{1}{|C|} \sum_{F\in\partial C} W_{F\leftarrow C} \tag{2.11}$$

which is a mapping from X^h to Q^h. Then, replacing (2.11) in (2.9) and using definition (2.6) yields:

$$[\vec{W}, \mathcal{G}^h q]_{C,X^h} = ((q_C - q_F)_{F\in\partial C}, W_{F\leftarrow C}) \tag{2.12}$$

where we introduced the notation:

$$(q_C - q_F)_{F\in\partial C} = \begin{pmatrix} q_C - q_{F_1} \\ q_C - q_{F_2} \\ \dots \\ q_C - q_{F_{k_C}} \end{pmatrix} \tag{2.13}$$

Looking back at the original mixed formulation (2.2) for the anisotropic diffusion equation, it comes natural to place the unknown pressure p in the space Q^h and the unknown velocity $\vec{V}$ in X^h (where, according to (2.5), it takes the form of a set of discrete fluxes). Then the discretized constitutional equation in (2.2) in weak form reads:

$$[\mathbb{K}^{-1}\vec{V}, \vec{W}]_{C,X^h} = [\mathcal{G}^h p, \vec{W}]_{C,X^h} \qquad \forall \vec{W} \in H(\text{div}, \Omega) \tag{2.14}$$

According to property (2.12) the right-hand side of (2.14) can be expressed as $((p_C - p_F)_{F\in\partial C}, W_{F\leftarrow C})$. Hence, applying definition (2.7) to the left-hand side of (2.14) we get:

$$(\mathbb{M}_C(V_{F\leftarrow C})_{F\in\partial C}, (W_{F\leftarrow C})_{F\in\partial C}) = ((p_C - p_F)_{F\in\partial C}, (W_{F\leftarrow C})_{F\in\partial C}) \tag{2.15}$$

Where $\mathbb{M}_C$ is a $\mathbb{K}$-scalar product matrix, i.e. the previously discussed scalar product matrix $\mathbb{B}_C$ conveniently modified to incorporate the material diffusivity tensor $\mathbb{K}$ (more precisely, its inverse). In other words, we re-defined the scalar product in X^h to be material-dependent:

$$[\vec{V}, \vec{W}]_{C,X^h} := (\mathbb{M}_C(V_{F\leftarrow C})_{F\in\partial C}, (W_{F\leftarrow C})_{F\in\partial C}) \qquad \forall \vec{V}, \vec{W} \in H(\text{div}, \Omega) \tag{2.16}$$

In this sense, we can consistently refer to this operator as a ***flux*** operator. Since $\vec{W}$ is an arbitrary vector-valued function, (2.15) implies the following identity:

$$\mathbb{M}_C(V_{F\leftarrow C})_{F\in\partial C} = (p_C - p_F)_{F\in\partial C}$$

i.e.:

$$(V_{F\leftarrow C})_{F\in\partial C} = \mathbb{M}_C^{-1}(p_C - p_F)_{F\in\partial C} \tag{2.17}$$

which is interpreted as the discrete approximation to:

$$\int_F \vec{V}\cdot\vec{n}_F\, d\Sigma = \int_F -\mathbb{K}\nabla p\cdot\vec{n}_F\, d\Sigma \tag{2.18}$$

In the reminder of this paper will often make use of (2.17) as a starting point to derive a complete method as well as some additional features.

2.3 Construction of the scalar product matrix

The crucial point in mimetic methods is the construction of $\mathbb{M}_C$, the $\mathbb{K}$-scalar product matrix in X^h, for each cell. As stated in section 2.1, several approaches exist, but ultimately they all impose the same specific restraints on $\mathbb{M}_C$. We list them here as they are expressed in [6]:

- By definition of inner product, $\mathbb{M}_C$ must be symmetric positive definite $\forall C \in \Omega_h$.
- There exist two positive constants s_* and S^* such that $\forall C$:

$$s_* \sum_{F\in\partial C} |C| W_F^2 \le [\vec{W}, \vec{W}]_{C,X^h} \le S^* \sum_{F\in\partial C} |C| W_F^2 \tag{2.19}$$

- For every linear function q^I, the following must hold:

$$[\mathbb{K}\nabla q^I, \vec{W}]_{C,X^h} + \int_C q^I \mathcal{D}^h \vec{W}\, dV = \sum_{F\in\partial C} W_{F\leftarrow C} \frac{1}{|F|} \int_F q^I\, d\Sigma \tag{2.20}$$

We found in [9] a good interpretation of such restraints: (2.19) expresses a *stability condition*, imposing that $\mathbb{M}_C$ be spectrally equivalent to a scalar matrix and behaving as a mass matrix for cell C; (2.20) is a semi-discrete Gauss-Green formula, similar to (2.9) with the addition that it requires the scalar product to be exact where the interpolated function is linear (*local consistency*).

We suggest here a simple, although somewhat naive, process to derive a suitable $\mathbb{M}_C$. We start by introducing a *cell-average* operator for vector-valued quantities in X^h:

$$< \vec{W} >_C = \sum_{F\in\partial C} \frac{W_{F\leftarrow C}(\vec{x}_F - \vec{x}_C)}{|C|} \tag{2.21}$$

where $\vec{x}_F$ and $\vec{x}_C$ are respectively the centres of gravity of face F and cell C. Lemma (2.21) is derived from Stokes formula, proof of it can be found in e.g. the appendix of [11]. Then it would seem natural to define our $\mathbb{K}$-scalar product such that:

$$[\vec{V}, \vec{W}]^{avg}_{C,X^h} = |C|(\mathbb{K}_C^{-1} < \vec{V} >_C, < \vec{W} >_C) \tag{2.22}$$

where $\mathbb{K}_C$ is the cell-averaged diffusion tensor. Unfortunately, expression (2.22) alone is not sufficient to guarantee a stable scalar product. This is well shown in [14, 21], where the authors define directly a discrete gradient based on (2.21) and verify that this gradient vanishes for checkerboard modes (Fig.2). A trivial example in our case: over a cubic cell, if $W_{F\leftarrow C}$ happens to be the same on all six faces, then $< \vec{W} >_C = 0$ and the inner product (2.22) goes to 0 regardless of $\vec{V}$. Interestingly enough, authors who follow a completely different approach, such as [6, 9] who derive equations for the unknown $\mathbb{M}_C$ from local consistency (2.20), get to the same conclusion, i.e. they obtain a matrix which is only semi-definite.

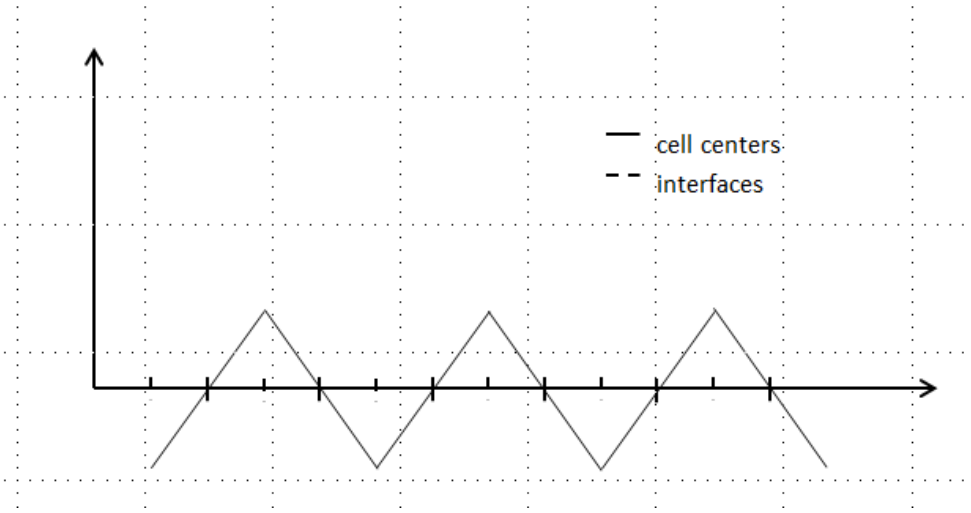


Figure 2: Checkerboard modes

Thus we are led to introduce a *stabilization term*, which we choose to be of the following form:

$$\mathcal{R}_C(\vec{V},\vec{W}) = \sum_{F\in\partial C} \lambda_{F,C}\left(V_{F\leftarrow C} - |F|(<\vec{V}>_C,\vec{n}_F)\right)\left(W_{F\leftarrow C} - |F|(<\vec{W}>_C,\vec{n}_F)\right) \tag{2.23}$$

with $\lambda_{F,C} = \mathcal{O}(h)$ (where h is some local mesh size) being a weighting factor whose precise expression will be discussed in section 2.6.

Hence the full expression for our $\mathbb{K}$-scalar product becomes:

$$[\vec{V},\vec{W}]_{C,X^h} = [\vec{V},\vec{W}]^{avg}_{C,X^h} + \alpha_C \mathcal{R}_C(\vec{V},\vec{W}) \tag{2.24}$$

with α_C being defined to scale with $\mathbb{K}_C^{-1}$ as suggested in [6, 7, 9, 18]:

$$\alpha_C = \frac{\text{trace}\left(\mathbb{K}_C^{-1}\right)}{d} \tag{2.25}$$

where d is the dimension of the geometrical space.

One can easily verify that the stabilization term is consistent as it vanishes for constant vector fields on the cell and that it makes the $\mathbb{K}$-scalar product positive definite. In fact one can show that the stability property (2.19) holds.

Another easy property to verify is that the inner product in the form (2.24) satisfies (2.20).

Proof Let us assume that $\vec{V} = \nabla q^I$; q^I is linear, hence $\vec{V}$ is constant. Then $<\vec{V}>_C = \vec{V}$, and $V_{F\leftarrow C} = \int_F \vec{V}\cdot\vec{n}_F\, d\Sigma = |F|(<\vec{V}>_C,\vec{n}_F)$, thus the stabilization term vanishes; we are left, replacing (2.21) in (2.22), with:

$$[\mathbb{K}\nabla q^I,\vec{W}]_{C,X^h} = |C|\left(\nabla q^I, \sum_{F\in\partial C}\frac{W_{F\leftarrow C}(\vec{x}_F - \vec{x}_C)}{|C|}\right) \tag{2.26}$$

Since q^I is linear, it follows that $(\nabla q^I, (\vec{x}_F - \vec{x}_C)) = q^I_F - q^I_C$ and we write, by using basic dot product properties:

$$[\mathbb{K}\nabla q^I,\vec{W}]_{C,X^h} = \sum_{F\in\partial C} q^I_F W_{F\leftarrow C} - q^I_C \sum_{F\in\partial C} W_{F\leftarrow C} \tag{2.27}$$

which, considering definitions (2.4), (2.10) and (2.11), is identical to our local consistency condition (2.20).

It is worth mentioning that the choice of the stabilization term is not unique, implying that, as [6] points out, *there is not a unique admissible* $\mathbb{M}_C$.

The reason for this is well explained in [7] where it is argued that, under some assumptions on the magnitude of the scaling factor in the stabilization term, if we defined a certain *lifting* (reconstruction) for quantities V_F and W_F and integrated the scalar product between such liftings over cell C, the result would be equivalent to that obtained via an expression of type (2.24), with the stabilization term depending on the specific reconstruction. In other words, we managed to define a scalar product (or better, a *family* of admissible scalar products, as [6] clarifies) between two functions lifted from the discrete space X^h, but without having to explicitly compute any shape functions. This consideration justifies the recasting of this family of approaches under the name Virtual Element Method (VEM).

2.4 Inversion of the local inner product matrix

It will be shown in the next section that most solution strategies require knowledge of the inverse of $\mathbb{M}_C$ for each cell (in fact, some require explicit knowledge of $\mathbb{M}_C^{-1}$ *only*). Some authors, e.g. [6, 18], have devised ways of computing $\mathbb{M}_C^{-1}$ directly, while others, like [11], choose to invert $\mathbb{M}_C$ via direct methods, which is computationally feasible since the matrix size scales with k_C, which is presumably small enough. In our case we worked out a way of computing $\mathbb{M}_C^{-1}$ which only requires direct inversion of a $d \times d$ matrix for each cell, hence making the computational cost of the inversion fully independent of mesh complexity. More precisely we have that:

$$\begin{aligned} \delta_{\mathbb{K}}^C p &= \mathbb{M}_C^{-1}(p_C - p_F)_{F\in\partial C} \\ &= -\left(|F|(\mathbb{K}_C \nabla_C^{\mathscr{G}} p, \vec{n}_F) + \alpha_C^{-1}\ \lambda_{F,C}^{-1}\ \left\{p_F - p_C - \left(\nabla_C^{\mathscr{L}} p, (\vec{x}_F - \vec{x}_C)\right)\right\}\right)_{F\in\partial C} \end{aligned} \tag{2.28}$$

where $\nabla_C^{\mathscr{G}}$ and $\nabla_C^{\mathscr{L}}$ are two linearly consistent approximate gradients, respectively based on Green-Gauss formula and least-square approach:

$$\nabla_C^{\mathscr{G}} p = \frac{1}{|C|} \sum_{F\in\partial C} p_F \vec{n}_F |F| \tag{2.29}$$

$$\nabla_C^{\mathscr{L}} p = \underset{\mathscr{A}\in\mathbb{R}^d}{\operatorname{argmin}} \sum_{F\in\partial C} \lambda_{F,C}^{-1}\ \{p_F - p_C - (\mathscr{A}, (\vec{x}_F - \vec{x}_C))\}^2 \tag{2.30}$$

Proof Let $\vec{V_C} = (V_F)_{F\in\partial C}$. by denoting $\tilde{v}_C = \vec{V_C} - |F|\left(< \vec{V} >_C, \vec{n}_F\right)_{F\in\partial C}$, we have the following decomposition:

$$\vec{V_C} = |F|\left(< \vec{V} >_C, \vec{n}_F\right)_{F\in\partial C} + \tilde{v}_c \tag{2.31}$$

and subsequently:

$$[\vec{V}, \vec{W}]_{C,X^h} = |C|\ \left\langle \mathbb{K}_C^{-1} < \vec{V} >, < \vec{W} > \right\rangle_{\mathbb{R}^d}^1 + \alpha_C\ \langle \tilde{v}, \tilde{w} \rangle_{\mathbb{R}^{|\partial C|}}^{\lambda} \tag{2.32}$$

where $\langle \cdot, \cdot \rangle_{\mathbb{R}^n}^{\mu}$ denotes the μ-weighted canonical inner product of $\mathbb{R}^n$ and for sake of readability we have skipped the C indices. Hence, by using (2.17) we get:

$$\begin{aligned} [\delta_{\mathbb{K}} p, \vec{W}]_{C,X^h} &= |C|\ \left\langle \mathbb{K}_C^{-1} < \delta_{\mathbb{K}} p >, < \vec{W} > \right\rangle_{\mathbb{R}^d}^1 + \alpha_C\ \left\langle \tilde{\delta_{\mathbb{K}} p}, \tilde{w} \right\rangle_{\mathbb{R}^{|\partial C|}}^{\lambda} \\ &= \left\langle \left\{\lambda_{F,C}^{-1}\,(p_C - p_F)\right\}_{F\in\partial C}, \vec{W} \right\rangle_{\mathbb{R}^{|\partial C|}}^{\lambda} \end{aligned} \tag{2.33}$$

At first let's consider any vector $\vec{W}_C \in \mathbb{R}^{|\partial C|}$ satisfying:

$$\vec{W}_C = |F|\left(< \vec{W} >_C, \vec{n}_F\right)_{F\in\partial C} \tag{2.34}$$

From (2.33) it comes immediately that:

$$|C|\ \left\langle \mathbb{K}_C^{-1} < \delta_{\mathbb{K}} p >, < \vec{W} > \right\rangle_{\mathbb{R}^d}^1 = \left\langle \left\{\lambda_{F,C}^{-1}\,(p_C - p_F)\right\}_{F\in\partial C}, \vec{W} \right\rangle_{\mathbb{R}^{|\partial C|}}^{\lambda} \tag{2.35}$$

Now as the subspace of $\mathbb{R}^{|\partial C|}$ corresponding to (2.34) is obviously equal to span $\{e_\alpha\}_{\alpha=1\cdots d}$ where $\{e_\alpha\}_{\alpha=1\cdots d}$ is the canonical basis of $\mathbb{R}^d$ and using the exactness of the average (2.21) on constant vector fields, we deduce that:

$$\begin{aligned} |C| \ \left\langle \mathbb{K}_C^{-1} < \delta_{\mathbb{K}}\, p >, e_\alpha \right\rangle_{\mathbb{R}^d}^1 &= \sum_{F\in\partial C} (p_C - p_F)\, n_F^\alpha |F| \\ &= -\sum_{F\in\partial C} p_F\, n_F^\alpha |F| \end{aligned} \tag{2.36}$$

Hence:

$$< \delta_{\mathbb{K}}\, p >_C = -\mathbb{K}_C\, \nabla_C^{\mathscr{G}}\, p \tag{2.37}$$

Now, lets's consider any vector $\vec{W}_C \in \mathbb{R}^{|\partial C|}$ satisfying:

$$\vec{W}_C = \tilde{v}_C \tag{2.38}$$

From (2.33) it comes that:

$$\begin{aligned} \left\langle \alpha_C\, \tilde{\delta_{\mathbb{K}}\, p}, \tilde{w} \right\rangle_{\mathbb{R}^{|\partial C|}}^\lambda &= \left\langle \left\{ \lambda_{F,C}^{-1} (p_C - p_F) \right\}_{F\in\partial C}, \vec{W} \right\rangle_{\mathbb{R}^{|\partial C|}}^\lambda \\ &= \left\langle \left\{ \lambda_{F,C}^{-1} (p_C - p_F) \right\}_{F\in\partial C}, \tilde{w} \right\rangle_{\mathbb{R}^{|\partial C|}}^\lambda \end{aligned} \tag{2.39}$$

Hence:

$$\tilde{\delta_{\mathbb{K}}\, p} = \alpha_C^{-1}\ \mathscr{P}_{\mathcal{F}}^{\perp,\lambda} \left(\left\{ \lambda_{F,C}^{-1} (p_C - p_F) \right\}_{F\in\partial C} \right) \tag{2.40}$$

where $\mathscr{P}_{\mathcal{F}}^{\perp,\lambda}$ denotes the orthogonal projection on $\mathcal{F}$ according to the λ-weighted inner product of $\mathbb{R}^{|\partial C|}$ and $\mathcal{F}$ is the space of all vectors satisfying (2.38). But using the definition of the cell average (2.21) we have that:

$$\mathcal{F} = \left(\mathcal{G}_\lambda = \text{span} \left\{ \left(\left(\lambda_{F,C}^{-1} (x_F^\alpha - x_C^\alpha) \right)_{F\in\partial C} \right)_{\alpha=1\cdots d} \right\} \right)^{\perp,\lambda} \tag{2.41}$$

Hence:

$$\tilde{\delta_{\mathbb{K}}\, p} \ = \ \alpha_C^{-1}\ \left\{ \left\{ \lambda_{F,C}^{-1} (p_C - p_F) \right\}_{F\in\partial C} - \mathscr{P}_{\mathcal{G}_\lambda}^{\perp,\lambda} \left(\left\{ \lambda_{F,C}^{-1} (p_C - p_F) \right\}_{F\in\partial C} \right) \right\} \tag{2.42}$$

With some obvious rescaling we deduce further that:

$$\begin{aligned} \left(\tilde{\delta_{\mathbb{K}}\, p} \right)_F &= \alpha_C^{-1}\, \lambda_{F,C}^{-1}\ \left\{ (p_C - p_F) - \left(\mathscr{P}_{\mathcal{G}_1}^{\perp,\lambda^{-1}} \left(\{ (p_C - p_G) \}_{G\in\partial C} \right) \right)_F \right\} \\ &= \alpha_C^{-1}\, \lambda_{F,C}^{-1}\ \left\{ (p_C - p_F) - \nabla_C^{\mathscr{L}}\, p \cdot (\vec{x}_C - \vec{x}_F) \right\} \end{aligned} \tag{2.43}$$

and, as anticipated, we notice that $\nabla_C^{\mathscr{L}}\, p$ may be computed by inverting a $d \times d$ system only, using:

$$\nabla_C^{\mathscr{L}}\, p = \mathbb{X}_C^{-1} \left(\left\langle \{p_F - p_C\}_{F\in\partial C}, \{x_F^\alpha - x_C^\alpha\}_{F\in\partial C} \right\rangle_{\mathbb{R}^{|\partial C|}}^{\lambda^{-1}} \right)_{1\le\alpha\le d} \tag{2.44}$$

with:

$$\mathbb{X}_C = \left(\left\langle x_C^\alpha - x_F^\alpha, x_C^\beta - x_F^\beta \right\rangle_{\mathbb{R}^{|\partial C|}}^{\lambda^{-1}} \right)_{1\le\alpha\le d}^{1\le\beta\le d} \tag{2.45}$$

2.5 Solution strategy for pure diffusion

Once all necessary operators have been assembled we can easily construct the discrete linear system for the whole problem. From (2.17), let us consider one face F interfacing elements $C+$ and $C-$. The same face-centered pressure value then appears in two equations:

$$\begin{aligned} p_F &= p_{C+} - (\mathbb{M}_{C+}(V_{G\leftarrow C+})_{G\in\partial C+}, \delta_{FG}) \\ p_F &= p_{C-} - (\mathbb{M}_{C-}(V_{G\leftarrow C-})_{G\in\partial C-}, \delta_{FG}) \end{aligned} \tag{2.46}$$

where δ_{FG} is the Kronecker delta. Therefore we can write:

$$\begin{aligned} s_{F,C+}\left(p_{C+} - (\mathbb{M}_{C+}(s_{G,C+}V_G)_{G\in\partial C+}, \delta_{FG})\right) + \\ s_{F,C-}\left(p_{C-} - (\mathbb{M}_{C-}(s_{G,C-}V_G)_{G\in\partial C-}, \delta_{FG})\right) = 0 \\ \forall F \in \Omega_h \end{aligned} \tag{2.47}$$

where we also made use of the notion of flux conservation (2.3). Writing (2.47) for all faces provides the constitutive law in (2.2) discretized over Ω_h. Mass conservation is discretized on each cell via the divergence operator (2.11):

$$((s_{F,C}V_F)_{F\in\partial C}, \mathbb{1}) = f_C|C| \tag{2.48}$$

Hence the discretised anisotropic diffusion equation (minus boundary conditions) on Ω_h takes the form:

$$\begin{bmatrix} \mathbb{H} & \mathbb{G} \\ \mathbb{G}^T & 0 \end{bmatrix} \begin{pmatrix} (V_F)_{F\in\Omega_h} \\ (p_C)_{C\in\Omega_h} \end{pmatrix} = \begin{pmatrix} 0 \\ (f_C|C|)_{C\in\Omega_h} \end{pmatrix} \tag{2.49}$$

Matrix $\mathbb{H}$ can be formally referred to as a ***Hodge*** operator, as it constitutes a linear mapping from d-forms to $(d-1)$-forms. Explicitation of the coefficients of $\mathbb{H}$ shows that the operator is symmetric (and positive definite, as expected since it represents a discrete scalar product integrated over the whole mesh). Hence the linear system in (2.49) is evidently symmetric and saddle-point. Several techniques have been devised to solve it (see [4]); a popular strategy is to build an approximate Schur Complement (which in turns requires an approximation to the inverse of $\mathbb{H}$), which is then used to precondition the system and solve iteratively.

Here hovever we choose to solve via *hybridization*, similarly to what is typically done in FEM. Taking again the flux as expressed in (2.17), substitution in (2.48) gives:

$$\left({\mathbb{M}_C}^{-1}(p_C - p_F)_{F\in\partial C}, \mathbb{1}\right) = f_C|C| \tag{2.50}$$

from which we explicitate the cell-centered pressure:

$$p_C = \frac{f_C|C| + (\mathbb{M}_C^{-1}\mathbb{1}, (p_F)_{F\in\partial C})}{(\mathbb{M}_C^{-1}\mathbb{1}, \mathbb{1})} \tag{2.51}$$

Reinjecting this in (2.17) we obtain an expression for the flux $V_{F\leftarrow C}$ in terms of p_F only. Finally, by imposing flux conservation (2.3) across each face, we obtain a linear system in the form:

$$\tilde{\mathbb{H}}(p_F)_{F\in\Omega_h} = (R\tilde{H}S_F)_{F\in\Omega_h} \tag{2.52}$$

It it easy to verify that in order to compute coefficients of $\tilde{\mathbb{H}}$ one only needs knowledge of the inverse of matrix $\mathbb{M}_C$ for each cell, which we compute as explained in section 2.4; we also observe that $\tilde{\mathbb{H}}$ is symmetric positive definite. The unknowns in (2.52) are the face-centered values of pressure, from which p_C and V_F can be retrieved in that order through (2.51) and (2.17) respectively.

2.6 Relashionship with classical Finite Volumes

In this paragraph we will consider the case of isotropic scalar diffusion, meaning $\mathbb{K}_C = k_C \mathbb{I}$. In this case the expression (2.28) simplifies to:

$$\delta_k^C p \;=\; -k_C \left(\nabla_C^{\mathscr{G}} p \cdot \vec{F} + \lambda_{F,C}^{-1} \left\{ p_F - p_C - \nabla_C^{\mathscr{L}} p \cdot (\vec{x}_F - \vec{x}_C) \right\} \right)_{F \in \partial C} \tag{2.53}$$

Following [15] in classical FV method one has to distinguish between two distinct situations, whether we are considering an orthogonal mesh (meaning vectors $\mathbf{d}$ and $\mathbf{S}$ as defined in Fig.3 are parallel) or on a general mesh.

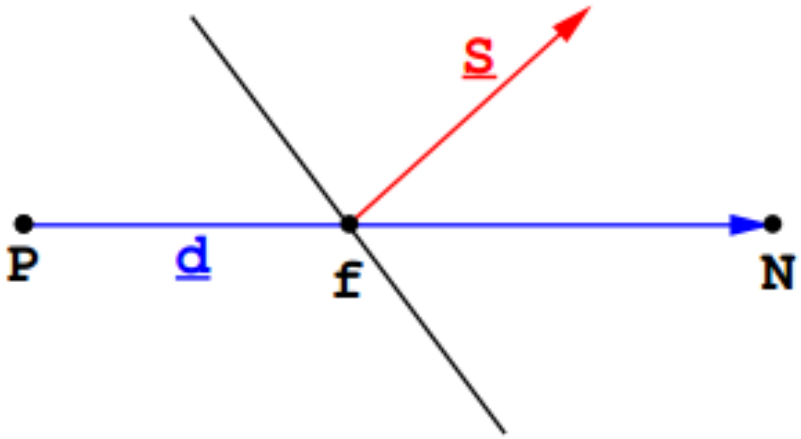


Figure 3: Vectors $\mathbf{d}$ and $\mathbf{S}$ on non-orthogonal mesh

When the mesh is orthogonal, the face gradient at an internal face may be expressed by the 2-points scheme:

$$\left(\delta_k^C \, p\right)^{\mathscr{FV}} = \left\{ < k_C, k_{C'} > \; s_{F,C} \, |F| \, \frac{p_{C'} - p_C}{\|x_{C'} - x_C\|} \right\}_{F \in \partial C} \tag{2.54}$$

where C' denotes the only other cell in the mesh sharing the face F with C and $< k_C, k_{C'} >$ is some 2-points face averaging of the diffusion coefficient. On can use for example the arithmetic average:

$$< k_C, k_{C'} >= \frac{k_C + k_{C'}}{2} \tag{2.55}$$

When the mesh is non-orthogonal, on the other hand, which as stated in [15] is more of a rule than an exception, one has to introduce a splitting of the kind:

$$\vec{F} = \alpha_F \, s_{F,C} \, (\vec{x}_{C'} - \vec{x}_C) + \vec{\tau}_F \tag{2.56}$$

where α_F is (hopefully) some positive coefficient. The face gradient is subsequently computed through:

$$\left(\delta_k^C \, p\right)^{\mathscr{FV}} = \left\{ < k_C, k_{C'} > \; \left(\underbrace{\alpha_F \, s_{F,C} \, (p_{C'} - p_C)}_{\text{orthogonal term}} + \underbrace{\text{"}\nabla_F \, p \cdot \vec{\tau}_F\text{"}}_{\text{NOC}} \right) \right\}_{F \in \partial C} \tag{2.57}$$

where the Non-Orthogonal Corrector (NOC) may be computed using interpolated node values, like in diamond schemes [10], or using cell gradients [15].

It is worth mentioning that the solution algorithm for such FV schemes is usually *semi-implicit*, using the deferred correction method [20] with the orthogonal term being treated implicitly while the NOC is added to the right-hand side, making the convergence behaviour of the method dependent on the orthogonality of the mesh. Even more problematic is the fact that, as the mesh non-orthogonality increases, the iterative correction can become unbounded [20], hence requiring some limiting procedure of the kind (with typical values for μ ranging from 0.333 to 0.5):

$$|"\nabla_F p \cdot \vec{\tau}_F"| \geq \mu \, |\alpha_F \, (p_{C'} - p_C)| \tag{2.58}$$

that surely degrades the accuracy of the scheme. Another important remark is that the face gradient is no longer approximated using a 2-points scheme but is now discretized with a Multipoint Flux Approximation (MPFA) [1]. Furthermore the discretized system is usually non-symmetric, thus violating a fundamental property of the continuous problem.

[15] gives 3 examples of face vector splitting of the form (2.56):

- *minimal correction*:
$$\alpha_F = s_{F,C} \, \frac{(\vec{x}_{C'} - \vec{x}_C) \cdot \vec{F}}{\|\vec{x}_{C'} - \vec{x}_C\|^2}$$
- *orthogonal correction*:
$$\alpha_F = \frac{|F|}{\|\vec{x}_{C'} - \vec{x}_C\|}$$
- *over-relaxed correction*:
$$\alpha_F = s_{F,C} \, \frac{|F|^2}{(\vec{x}_{C'} - \vec{x}_C) \cdot \vec{F}}$$

It is worth mentioning that neither the minimal correction nor the over-relaxed correction approach guarantee the positivity of α_F when $s_{F,C} \, (\vec{x}_{C'} - \vec{x}_C) \cdot \vec{F}$ becomes negative (which may happen only for non-convex cells), with the over-relaxed approach even presenting some singularity at the change of sign. On a large variety of meshes however the positivity holds and [15] demonstrates numerically the superiority of the over-relaxed approach when the degree of non-orthogonality increases, with fairly robust behaviour for meshes with angles of non-orthogonality up to 65 degrees.

Going back to the Mixed Virtual Element approach, we will now consider symmetric face coefficients of the form:

$$\lambda_F = \lambda_{F,C} = \lambda_{F,C'} = \mu_F \, \frac{\|\vec{x}_{C'} - \vec{x}_C\|}{2\,|F|} \tag{2.59}$$

with μ_F being some positive weight. By making use of the flux conservation (2.3) and one-sided flux expression (2.53) one gets:

$$\left(\delta_k^C \, p\right)_F = \frac{2\,k_C\,k_{C'}}{k_C + k_{C'}} \left\{ s_{F,C} \, \mu_F \, |F| \, \frac{p_{C'} - p_C}{\|\vec{x}_{C'} - \vec{x}_C\|} + \mathscr{T}_F^{\text{NOC}} \right\} \tag{2.60}$$

with:

$$\mathscr{T}_F^{\text{NOC}} = \frac{\left(\nabla_C^{\mathscr{G}} \, p + \nabla_{C'}^{\mathscr{G}} \, p\right)}{2} \cdot \vec{F} - s_{F,C} \, \mu_F \, \left(\nabla_C^{\mathscr{L}} \, p \cdot (\vec{x}_F - \vec{x}_C) + \nabla_{C'}^{\mathscr{L}} \, p \cdot (\vec{x}_{C'} - \vec{x}_F)\right) \tag{2.61}$$

in which we can observe obvious similarities with (2.57) by identifying

$$\alpha_F = \mu_F \, \frac{|F|}{\|\vec{x}_{C'} - \vec{x}_C\|}$$

and

$$< k_C, k_{C'} >= \frac{2\, k_C \, k_{C'}}{k_C + k_{C'}}$$

Playing on the value of μ_F, one can then retrieve Mixed Virtual Element analogous of the minimal correction, the orthogonal correction and the over-relaxed correction schemes. The difference is that this time the solution method is fully implicit, making the efficiency of the scheme independent of mesh orthogonality, and the fact that the symmetry of the continuous problem is preserved.

Notice that we may also choose to define non-symmetric face weights, i.e. different depending on which side of F we are considering. It is sufficient to replace the distance between cell centres with the distance between face and cell centre:

- *orthogonal non-symm*:
$$\lambda_{F,C} = \frac{\|\vec{x}_F - \vec{x}_C\|}{|F|}$$
- *over-relaxed non-symm*:
$$\lambda_{F,C} = \frac{(\vec{x}_F - \vec{x}_C) \cdot \vec{n}_F}{|F|}$$

One last remark is in order: in case we are running on a fully Cartesian mesh, additional symmetry properties provide that:

$$\nabla_C^{\mathscr{L}} \, p = \nabla_C^{\mathscr{G}} \, p$$

Hence we see, thanks to (2.60), that we retrieve the classical FV 2-points scheme for any of the aforementioned choices of μ_F.

3 CONVECTION-DIFFUSION-REACTION

Based on the framework for pure anisotropic diffusion described in the previous section, we now proceed to add the terms required to discretise the convection-diffusion-reaction equation:

$$\nabla \cdot (-\mathbb{K} \nabla p + \vec{U} p) + a p = f \tag{3.1}$$

where $\vec{U}$ is the convective field and a the reaction coefficient. Again, we split the problem in two first-order equations:

$$\begin{aligned} \vec{V} &= -\mathbb{K} \nabla p + \vec{U} p \\ \nabla \cdot \vec{V} + a p &= f \end{aligned} \tag{3.2}$$

3.1 Discretization of the convective term

The diffusive term is discretized via the flux operator built in section 2.2. Addition of a discretized convective term leads to:

$$V_{F \leftarrow C} = \left({\mathbb{M}_C}^{-1} (p_C - p_G)_{G \in \partial C}, \delta_{FG} \right) + A p_F + B p_C \tag{3.3}$$

where coefficients *A* and *B* depend on the specific strategy adopted to evaluate the convective flux on each face. Following the ideas expressed in [22] we formulate a unified approach to handle this, limiting our choice for now to four strategies: *hybrid centered*, *mixed centered*, *hybrid 1st-order upwind* and *hybrid θ-scheme*.

These are essentially equivalent to their classical FV homonyms with the difference that, in our framework, scalar fields evaluated at cell faces exist naturally as degrees of freedom of the problem; as suggested by [13], we take advantage of this by using these variables directly rather than resorting to averaging procedures which may affect the consistency of the method. Hence, hybrid centering simply takes the face-centered pressure p_F as the convected quantity; hybrid 1st-order upwinding takes p_C if cell C is upwind w.r.t. F, p_F otherwise; the θ-scheme is an intermediate choice between these two; and mixed centering is the equivalent to a FV weighted 2-points scheme. This leads to the following unified formulation:

$$V_{F\leftarrow C} = (\mathbb{N}_C(p_C - p_G)_{G\in\partial C}, \delta_{FG}) + \varphi_{F,C}p_C \tag{3.4}$$

where:

$$\varphi_{F,C} = (U_{F\leftarrow C}) \tag{3.5}$$

(i.e. the convective flux through face *F* as seen from cell *C*, projected onto X^h via (2.5)) and:

$$\mathbb{N}_C = {\mathbb{M}_C}^{-1} - \text{diag}(\Lambda_{G,C})_{G\in\partial C} \tag{3.6}$$

with:

$$\Lambda_{G,C} = \begin{cases} \varphi_{G,C} & \text{for hybrid centering} \\ 0 & \text{for mixed centering} \\ \min(0, \varphi_{G,C}) & \text{for hybrid 1st-order upwinding} \\ \min(\theta\varphi_{G,C}, \varphi_{G,C}) & \text{for hybrid } \theta\text{-scheme} \end{cases} \tag{3.7}$$

3.2 Solution strategy for convection-diffusion-reaction

The unified framework makes it possible to assemble the convection-diffusion-reaction operator for the whole domain with minimal modifications to what we did in section 2.5 for the pure diffusion case. In particular, the whole system (before applying boundary conditions) takes the form:

$$\begin{bmatrix} \mathbb{H}_N & \mathbb{T} \\ \mathbb{G}^T & \mathbb{A} \end{bmatrix} \begin{pmatrix} (V_F)_{F\in\Omega_h} \\ (p_C)_{C\in\Omega_h} \end{pmatrix} = \begin{pmatrix} 0 \\ (f_C|C|)_{C\in\Omega_h} \end{pmatrix} \tag{3.8}$$

Here the *Hodge matrix $\mathbb{H}_N$ is formally equivalent to $\mathbb{H}$ in (2.49), with the difference that its coefficients are based on $\mathbb{N}_C$ rather than $\mathbb{M}_C$; $\mathbb{T}$ is equivalent to $\mathbb{G}$ with the addition of convective coefficients as expressed in (3.5); $\mathbb{A}$ is simply a diagonal matrix holding the reaction coefficient integrated over each cell: $a_C|C|$.

System (3.8) is no longer saddle-point nor symmetric. However it is still possible to hybridize as above, i.e. by explicitating the cell-centered pressure:

$$p_C = \frac{f_C|C| + (\mathbb{N}_C\mathbb{1}, (p_F)_{F\in\partial C})}{(\mathbb{N}_C\mathbb{1}, \mathbb{1}) + ((\varphi_{F,C})_{F\in\partial C}, \mathbb{1}) + a_C|C|} \tag{3.9}$$

then re-injecting in (3.4) and imposing flux conservation (2.3), which again yields a system where the unknowns are the face-centered pressures:

$$\tilde{\mathbb{H}}_N(p_F)_{F\in\Omega_h} = (R\tilde{H}S_{NF})_{F\in\Omega_h} \tag{3.10}$$

Solution to (3.10) is then used to retrieve p_C and V_F, in that order, via (3.9) and (3.4).

3.3 Stability estimate

In this paragraph we will consider the case of pure convection-diffusion problem and make the assumption that $\nabla \cdot \vec{U} \geq 0$ and

$$U_{F \leftarrow C} = \int_F \vec{U} \cdot \vec{n}_F \, d\Sigma$$

With the help of previously defined notations one may write the following estimate:

$$\sum_C \left\langle \mathbb{N}_C \left(p_F - p_C\right)_{F \in \partial C}, \left(p_F - p_C\right)_{F \in \partial C} \right\rangle_{\mathbb{R}^{|\partial C|}}$$
$$+\frac{1}{2} \sum_C \sum_{F \in \partial C} \varphi_{F,C} \left(p_F - p_C\right)^2 + \frac{1}{2} \sum_C (\int_C \nabla \cdot \vec{U} \, dx) \, p_C^2 = \sum_C |C| f_C \, p_C \tag{3.11}$$

Proof By virtue of continuity of the convective flux at the interfaces and (3.2) we get that:

$$\begin{aligned} \sum_C \sum_{F \in \partial C} V_{F \leftarrow C} \left(p_C - p_F\right) &= \sum_C \left(\sum_{F \in \partial C} V_{F \leftarrow C} \right) p_C \\ &= \sum_C (\int_C \nabla \cdot \vec{V} \, dx) p_C \\ &= \sum_C |C| \, f_C p_C \end{aligned} \tag{3.12}$$

On the other hand, we have thanks to the unified framework notations (3.4):

$$\begin{aligned} \sum_C \sum_{F \in \partial C} V_{F \leftarrow C} \left(p_C - p_F\right) &= \sum_C \left\langle \mathbb{N}_C \left(p_F - p_C\right)_{F \in \partial C}, \left(p_F - p_C\right)_{F \in \partial C} \right\rangle_{\mathbb{R}^{|\partial C|}} \\ &+ \sum_C \sum_{F \in \partial C} \varphi_{F,C} \, p_C \left(p_C - p_F\right) \end{aligned} \tag{3.13}$$

The second term on the right-hand side can be further decomposed:

$$\begin{aligned} \sum_C \sum_{F \in \partial C} \varphi_{F,C} \, p_C \left(p_C - p_F\right) &= \sum_C \sum_{F \in \partial C} \varphi_{F,C} \left(p_C - p_F\right)^2 \\ &+ \sum_C \sum_{F \in \partial C} \varphi_{F,C} \, p_F \left(p_C - p_F\right) \end{aligned} \tag{3.14}$$

By making use again of flux continuity it comes then:

$$\begin{aligned} \sum_C \sum_{F \in \partial C} \varphi_{F,C} \, p_C \left(p_C - p_F\right) &= \sum_C \sum_{F \in \partial C} \varphi_{F,C} \left(p_C - p_F\right)^2 \\ &+ \sum_C \sum_{F \in \partial C} \varphi_{F,C} \, p_F \, p_C \\ &= \sum_C \sum_{F \in \partial C} \varphi_{F,C} \left(p_C - p_F\right)^2 \\ &- \sum_C \sum_{F \in \partial C} \varphi_{F,C} \, p_C \left(p_C - p_F\right) \\ &+ \sum_C (\int_C \nabla \cdot \vec{U} \, dx) \, p_C^2 \end{aligned} \tag{3.15}$$

From which we deduce that:

$$\sum_C \sum_{F\in\partial C} \varphi_{F,C}\, p_C\,(p_C - p_F) \;=\; \frac{1}{2}\sum_C \sum_{F\in\partial C} \varphi_{F,C}\,(p_C - p_F)^2 + \frac{1}{2}\sum_C (\int_C \nabla\cdot\vec{U}\; dx)\; p_C^2$$

which concludes the proof.

Applying the stability estimate (3.11) in case of the hybrid centered scheme gives:

$$\sum_C \left\langle \mathbb{M}_C^{-1}\,(p_F - p_C)_{F\in\partial C}\,, (p_F - p_C)_{F\in\partial C}\right\rangle_{\mathbb{R}^{|\partial C|}}$$
$$-\frac{1}{2}\sum_C \sum_{F\in\partial C} \varphi_{F,C}\,(p_F - p_C)^2 + \frac{1}{2}\sum_C (\int_C \nabla\cdot\vec{U}\; dx)\; p_C^2 = \sum_C |C| f_C\, p_C \tag{3.16}$$

the second term

$$-\frac{1}{2}\sum_C \sum_{F\in\partial C} \varphi_{F,C}\,(p_F - p_C)^2$$

in the equation is the one that is problematic because $\varphi_{F,C}$ has no definite sign. It can be either positive or negative. Of course, as $\mathbb{M}_C^{-1} = \mathcal{O}(h)$ and $\varphi_{F,C} = \mathcal{O}(h^2)$, the positive diffusive term will ultimately dominate, but when the Peclet number is big, the tipping point may be delayed to very fine meshes that cannot be afforded in practice.

For the mixed centered scheme, the estimate is similar:

$$\sum_C \left\langle \mathbb{M}_C^{-1}\,(p_F - p_C)_{F\in\partial C}\,, (p_F - p_C)_{F\in\partial C}\right\rangle_{\mathbb{R}^{|\partial C|}}$$
$$+\frac{1}{2}\sum_C \sum_{F\in\partial C} \varphi_{F,C}\,(p_F - p_C)^2 + \frac{1}{2}\sum_C (\int_C \nabla\cdot\vec{U}\; dx)\; p_C^2 = \sum_C |C| f_C\, p_C \tag{3.17}$$

leading to the same problematic.

For the hybrid upwind scheme however the estimate is more favourable. Indeed it gives:

$$\sum_C \left\langle \mathbb{M}_C^{-1}\,(p_F - p_C)_{F\in\partial C}\,, (p_F - p_C)_{F\in\partial C}\right\rangle_{\mathbb{R}^{|\partial C|}}$$
$$+\frac{1}{2}\sum_C \sum_{F\in\partial C/\varphi_{F,C}\geq 0} \varphi_{F,C}\,(p_F - p_C)^2 - \frac{1}{2}\sum_C \sum_{F\in\partial C/\varphi_{F,C}\leq 0} \varphi_{F,C}\,(p_F - p_C)^2$$
$$+\frac{1}{2}\sum_C (\int_C \nabla\cdot\vec{U}\; dx)\; p_C^2 = \sum_C |C| f_C\, p_C \tag{3.18}$$

This time the scheme is clearly stable in an appropriate norm involving variations of the scalar variable. In fact the estimate shows that we even have some room for reduced dissipation. Indeed, if instead of taking p_C as the face value when the flux is upwind we choose $\theta\, p_F + (1-\theta)\, p_C$, we get the following estimate

$$\sum_C \left\langle \mathbb{M}_C^{-1}\,(p_F - p_C)_{F\in\partial C}\,, (p_F - p_C)_{F\in\partial C}\right\rangle_{\mathbb{R}^{|\partial C|}}$$
$$+\frac{1-2\,\theta}{2}\sum_C \sum_{F\in\partial C/\varphi_{F,C}\geq 0} \varphi_{F,C}\,(p_F - p_C)^2 - \frac{1}{2}\sum_C \sum_{F\in\partial C/\varphi_{F,C}\leq 0} \varphi_{F,C}\,(p_F - p_C)^2$$
$$+\frac{1}{2}\sum_C (\int_C \nabla\cdot\vec{U}\; dx)\; p_C^2 = \sum_C |C| f_C\, p_C \tag{3.19}$$

Thus we can see that as long that θ stays bounded below 0.5 the scheme remains stable.

4 NUMERICAL RESULTS

The development of a code fully capable of solving according to the methodologies outlined in the previous sections is still very much in progress. However we find appropriate to include here and comment on a few preliminary results.

4.1 Pure anisotropic diffusion test case

A first validation of the method is done by solving for a test case of 2D pure anisotropic diffusion. We choose the benchmark case proposed by [6, 16], where we solve (2.1) in the unit square $\Omega =]0,1[\times]0,1[$. The diffusion tensor is taken as:

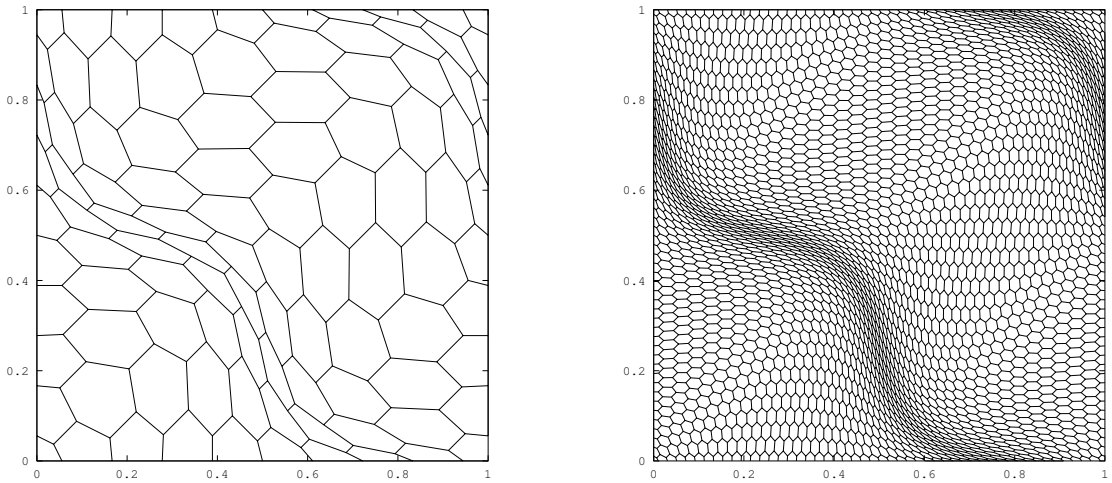
$$\mathbb{K} = \begin{pmatrix} (x+1)^2 + y^2 & -xy \\ -xy & (x+1)^2 \end{pmatrix}$$

and the source term is calculated such that the exact solution is:

$$p_{ex}(x,y) = x^3y^2 + x\sin(2\pi xy)\sin(2\pi y)$$

We use a mix of Dirichlet, Neumann and Robin boundary conditions on the four sides of Ω.

Literature suggests testing on a polygonal unstructured mesh based on the dual to a Voronoi tessellation, generated via the algorithm described in [22] and shown in Fig.4 for two different values of refinement h; such mesh features strongly skewed and non-orthogonal cells, which makes it suitable to test the capabilities of mimetic methods. Since the analytical solution is



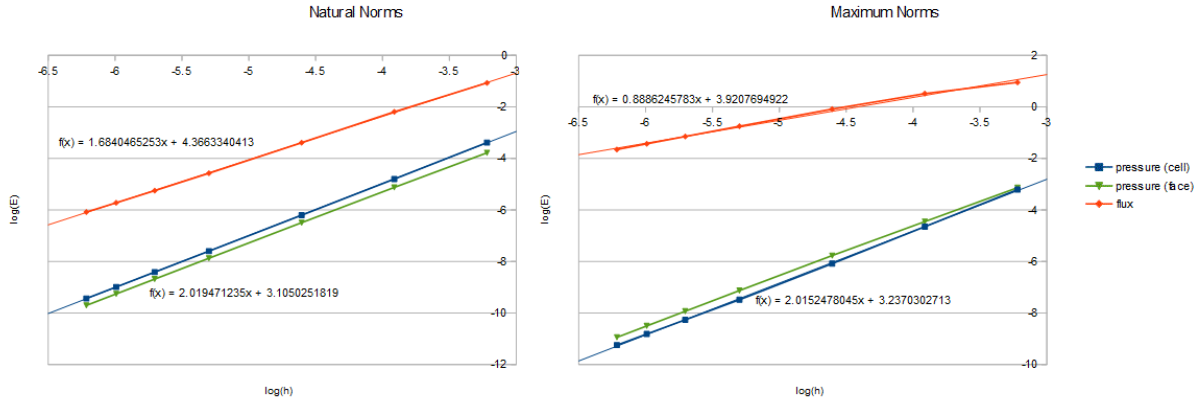
(a) h=0.1 (b) h=0.02

Figure 4: Polygonal mesh for different h

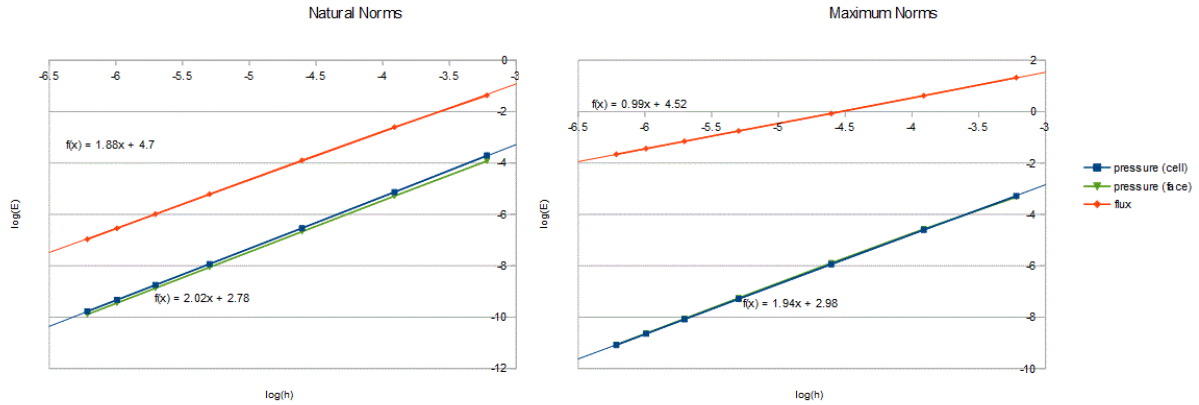
available, we can perform an h-convergence test by measuring, for different degrees of mesh refinement, the error on p and $\vec{V}$ in terms of both maximum norms in the Euclidean space and *natural* norms, based on discrete scalar products (2.6) and (2.16):

$$\begin{aligned} \|p_{ex} - p\| &= [p_{ex}(x_C, y_C) - p_C, p_{ex}(x_C, y_C) - p_C]^{1/2}_{C,Q^h} \\ \|\vec{V}_{ex} - \vec{V}\| &= [(\vec{V}_{ex}(x_F, y_F), \vec{n}_F) - V_{F\leftarrow C}, (\vec{V}_{ex}(x_F, y_F), \vec{n}_F) - V_{F\leftarrow C}]^{1/2}_{C,X^h} \end{aligned}$$

For the sake of comparison we performed the same computation on progressively refined Cartesian meshes as well. In Fig.5 we plot the error norms against h in logarithmic scale and analyze their linear regression to see convergence rates; we also include for p_C and V_F the equation obtained via linear regression. In all cases we observe second order convergence for p (both



(a) Polygonal mesh



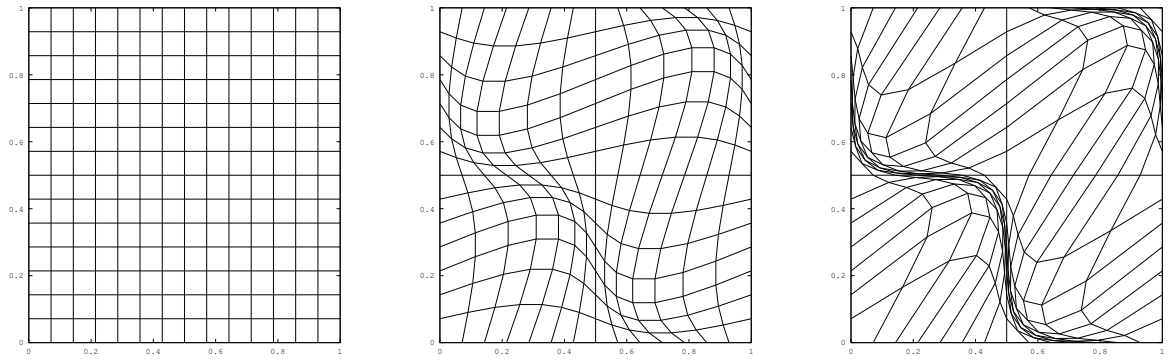
(b) Cartesian mesh

Figure 5: Pure diffusion: h-convergence for different meshes

cell-centered and face-centered); as for the flux V, we observe linear or super-linear convergence, which deteriorates as the mesh quality worsens. These results are in perfect agreement with the theoretical findings expressed by [5].

4.2 Comparison between different choices of λ_F

We discussed in section 2.6 how the choice of weight $\lambda_{F,C}$ in (2.23) is not unique, and various expressions of it can reduce the MFD formulation to various classical FV formulations for orthogonal meshes. We now want to compare how different weight types perform on progressively non-orthogonal meshes (such progressive distortion is represented in Fig.6, on a coarse mesh initially Cartesian). When naming the weight types, the absence of suffix "_sym" means that we are using the non-symmetric weight expressions as explained in section 2.6. Again we carry out our tests on two types of 2D mesh, one polygonal unstructured and the other quadrilateral, and we solve for the same exact solution as in the previous section, but with isotropic



(a) (b) (c)

Figure 6: Progressive distortion of a structured mesh

(scalar) diffusivity:

$$\mathbb{K}(x, y) = (1 + xy)\mathbb{I}$$

We take as an indicator of non-orthogonality the minimum value of $\cos\theta_F$ found in the mesh, with θ_F being the angle between the face-normal $\vec{n}_F$ and the distance between the centres of the two cells adjacent to F: smaller values indicate higher non-orthogonality.

$\cos\theta_F$	orthogonal	over-relaxed	orthogonal_sym	over-relaxed_sym	minimal_sym
0.83	5.63E-004	4.50E-004	5.51E-004	4.45E-004	6.94E-004
0.64	6.28E-004	5.20E-004	6.19E-004	5.19E-004	7.90E-004
0.20	1.34E-003	1.02E-003	1.35E-003	1.02E-003	2.32E-003
6.74E-2	4.16E-003	3.32E-003	4.22E-003	3.31E-003	8.02E-003
2.41E-2	1.42E-002	1.14E-002	1.42E-002	1.15E-002	3.98E-002

Table 1: Polygonal mesh: effect of mesh non-orthogonality for different expressions of λ_F

$\cos\theta_F$	orthogonal	over-relaxed	orthogonal_sym	over-relaxed_sym	minimal_sym
1.00	5.78E-004	5.78E-004	5.78E-004	5.78E-004	5.78E-004
0.81	6.78E-004	6.52E-004	6.78E-004	6.52E-004	7.06E-004
0.32	1.64E-003	1.15E-003	1.64E-004	1.15E-003	7.72E-003
1.00E-1	9.21E-003	3.26E-003	9.14E-003	3.24E-003	6.53E-002
3.65E-2	1.52E-002	8.34E-003	1.47E-002	8.21E-003	2.01E-002

Table 2: Quadrilateral mesh: effect of mesh non-orthogonality for different expressions of λ_F

Results are reported in table 1 and 2 for polygonal and quadrilateral meshes, respectively, where we output the error on p_C in natural norm. We observe first of all that all weights produce similar results in terms of order of magnitude, with a slight superiority of the over-relaxed correctors and a deterioration of the minimal symmetric one for higher values of non-orthogonality. This is in agreement with what expressed by [15], with the difference that thanks to the VEM framework the error remains acceptable on any mesh as we claimed in section 2.6.

At a first glance it would appear that the error does increase with mesh non-orthogonality, which would directly contradict the claims we previously made regarding independency of MFD methods from mesh regularity. We believe however that this effect is not linked to orthogonality at all. It is evident from Figure 6 that the algorithm we use to distort our mesh doesn't just affect mesh orthogonality, but also local coarseness. Such algorithm however does not affect the boundary faces, and since we (naively) take their area h as indicator of refinement we get a series of meshes that, although nominally all equivalent in terms of coarseness, feature in certain zones some cell volumes and face areas so large with respect to the original Cartesian grid that they have significant impact on the error, hence the comparison is somewhat invalidated.

Such hypothesis is partially confirmed by the fact that all our h-convergence tests produced positive results regardless of the degree of non-orthogonality (in fact, despite the mesh refinement process negatively affecting mesh orthogonality); however further verifications are due, on meshes that are actually equivalent coarseness-wise but different orthogonality-wise.

4.3 Convection-diffusion test case

We now run a simple numerical experiment for a 2D convection-diffusion test case; we report here the results obtained on the same type of unstructured mesh (Fig.4). We choose a linear exact solution:

$$p_{ex}(x, y) = 2x + 3y$$

and a material diffusivity ν which is isotropic and constant across Ω. As for the convective field $\vec{U}$, for the first test case we set it to also be constant. This constitutes a *patch test*: since we claimed that the scalar product (2.16) is exact for linear functions, and the interpolator (2.5) is also exact for a constant convective flux, we expect the results to match the analytical solution down to machine precision. We choose to solve via a hybrid strategy and we test on all four convective schemes described in section 3.1 (for the θ-scheme we set $\theta = 0.49$ to ensure stability).

We test for different values of the Peclet number $Pe = |\vec{U}|/\nu$, making the value of ν increasingly smaller so that the problem becomes more and more convection-dominated; we report in table 3 the measured errors on p_C in natural norm.

Pe	hyb. centered	mix. centered	hyb. 1st upwinding	hyb. θ-scheme (0.49)
1.41E+000	5.73E-015	3.72E-005	1.95E-004	9.98E-005
1.41E+001	3.91E-014	3.56E-004	1.68E-003	8.88E-004
1.41E+002	1.44E-015	2.52E-003	5.84E-003	3.71E-003
1.41E+003	2.09E-012	5.77E-003	9.88E-003	8.87E-003
1.41E+004	3.57E-010	6.50E+000	1.10E-002	1.10E-002
1.41E+005	6.27E-009	9.98E-001	1.11E-002	1.13E-002
1.41E+006	2.53E-007	6.22E+000	1.11E-002	1.13E-002
1.41E+007	1.21E-006	6.59E+001	1.11E-002	1.13E-002
1.41E+008	5.99E-005	6.63E+002	1.11E-002	1.13E-002

Table 3: Convection-diffusion with linear p_{ex}, measure of error on p_C

We observe that, at least for "reasonable" values of Pe, we indeed get the exact solution when convective terms are discretized via the hybrid centered strategy, while other schemes produce

a less precise solution. This is expected: hybrid centering always takes the value of p_F as the convected quantity (which is indeed the exact value, since both the face interpolator (2.10) and the scalar product (2.16) are exact for linear functions); mixed-centering and upwinding, on the other hand, are sources of discretization error and therefore yield less precise results. It is also observed that, for higher Pe, hybrid centering ceases to produce the exact solution and the error seems to become unbound: we argue that this effect is due to the fact that too high values of Pe affect the condition number of the matrix in system (3.10) to a point where a standard linear solver is no longer capable of handling it.

We then run a second set of tests, this time seeking a nonlinear scalar field:

$$p_{ex}(x, y) = 2x^2 + \cos(2\pi x y^2)$$

with a linear conservative, but not constant, convective field:

$$\vec{U}(x, y) = \begin{pmatrix} 10x + 2 \\ 3x - 10y \end{pmatrix}$$

Pe	hyb. centered	mix. centered	hyb. 1st upwinding	hyb. θ-scheme (0.49)
8.53E+000	1.87E-004	4.34E-004	2.58E-003	1.33E-003
8.53E+001	1.60E-004	2.76E-003	9.43E-003	5.51E-003
8.53E+002	1.89E-004	3.73E+002	1.72E-002	1.37E-002
8.53E+003	4.22E+000	3.25E+002	2.05E-002	2.00E-002
8.53E+004	1.81E+000	3.34E+000	2.11E-002	2.14E-002
8.53E+005	5.16E+000	4.61E+000	2.11E-002	2.16E-002
8.53E+006	1.33E+001	1.29E+001	2.11E-002	2.16E-002
8.53E+007	5.36E+001	1.44E+002	2.11E-002	2.16E-002
8.53E+008	4.81E+002	1.47E+003	2.11E-002	2.16E-002

Table 4: Convection-diffusion with nonlinear p_{ex}, measure of error on p_C

Results are reported in table 4. As expected, both centering strategies yield reliable results for low values of Pe but, as the problem becomes more convection-dominated, the solution becomes unreliable. The error on upwinding and θ-scheme, on the other hand, ceases to grow after a certain Pe and becomes bound; in fact, the same behaviour was already evident in the patch test, table 3. We also notice that in both cases, when applying the θ-scheme, results are very similar to those obtained via upwinding with a slight improvement at low-Pe and a slight deterioration at high-Pe; that makes sense since this scheme is effectively a trade-off between centering and upwinding, hence there will be a range of Pe for which the chosen value of θ is optimal. Therefore it would be convenient to have a θ that varies according to the local diffusivity; this what the Scharfetter-Gummel scheme does, see [13, 22].

5 CONCLUSIONS

Mimetic discretizations surely represent a promising new technology for computational engineering in general, including CFD, thanks to their superior consistency with respect to traditional methods and better convergence properties. As we said, they are particularly appealing in the context of optimization thanks to the extra freedom on element shapes, which is an enormous advantage in e.g. shape optimization processes featuring mesh-adaptation algorithms;

furthermore, both continuous and discrete CFD adjoint systems for gradient computation may benefit greatly (in terms of robustness) from a better converged and more consistent primal flow field.

In the present paper we outlined the MFD philosophy and presented our own implementation. Our preliminary numerical results on a pure anisotropic diffusion test case are in perfect agreement with previous literature and confirm the superiority of the method in terms of convergence properties; our results for 2D convection-diffusion cases are also quite encouraging, showing how the method can be extended with relative ease to cater for this type of problem as well.

In the near future we plan to a) explore more sophisticated ways of discretizing convection terms (some already dealt with in [22]), including 2nd order schemes; b) implement an incompressible steady-state Navier-Stokes solver using a SIMPLE-like preconditioner (which is feasible since we already have all the necessary tools to solve both convection-diffusion and Poisson equations) c) investigate alternative solution strategies for the Navier-Stokes and how they translate into the MFD framework.

Once the basic mimetic CFD solver will be fully validated and proven to be robust enough, it will then be possible to tackle more "advanced" issues (e.g. the inclusion of turbulence models); finally, we will move on to attempt a mimetic discretization of an adjoint system and assess the benefits that MFD brings about in that context.

6 ACKNOWLEDGEMENTS

This work was funded by the EU through the FP7-PEOPLE-2012-ITN "AboutFlow" Grant agreement number 317006.

REFERENCES

[1] I. Aavatsmark, An introduction to multipoint flux approximations for quadrilateral grids. *Computational Geosciences*, **6**, 405-432, 2002.

[2] P.F. Antonietti, L.B. da Veiga, N. Bigoni, M. Verani, Mimetic finite differences for nonlinear and control problems. *Mathematical Models and Methods in Applied Sciences*, accepted for publication, doi: 10.1142/S0218202514400016, 2014.

[3] P.F. Antonietti, N. Bigoni, M. Verani, Mimetic finite difference method for shape optimization problems. *Numerical Mathematics and Advanced Applications (ENUMATH 2013), Proceedings of the 10th European Conference on Numerical Mathematics and Advanced Applications*, Springer Verlag Italia, 2014.

[4] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems. *Acta Numerica*, **14**, 1–137, 2005.

[5] F. Brezzi, K. Lipnikov, M. Shashkov, Convergence of mimetic finite difference method for diffusion problems on polyhedral meshes. *SIAM Journal of Numerical Analysis*, **43** (5), 1872–1896, 2005.

[6] F. Brezzi, K. Lipnikov, V. Simoncini, A family of mimetic finite difference methods on polygonal and polyhedral meshes. *Mathematical Models and Methods in Applied Sciences*, **15** (10), 1533–1551, 2005.

[7] F. Brezzi, Cochain approximations of differential forms. *Foundations of Computational Mechanics*, Hong Kong, June 16-26, 2008.

[8] F. Brezzi, R.S. Falk, L.D. Marini, Basic principles of mixed virtual element method. *Mathematical Modeling and Numerical Analysis*, accepted for publication, 2014.

[9] A. Cangiani, G. Manzini, Flux reconstruction and solution post-processing in mimetic finite difference methods. *Computational Methods in Applied Mechanics and Engineering*, **197**, 933–945, 2008.

[10] Y. Coudiere, J.P. Vila, P. Villedieu, Convergence of a Finite-Volume schemes for a two-dimensional convection-diffusion problem. *Mathematical Modeling and Numerical Analysis*, **33** (3), 493–516, 1999.

[11] J. Droniou, R. Eymard, A mixed finite volume scheme for anisotropic diffusion problems on any grid. *Numerische Mathematik*, **105** (1), 35–71, 2006.

[12] J. Droniou, R. Eymard, Study of the mixed finite volume method for Stokes and Navier-Stokes equations. *Numerical Methods for Partial Differential Equations*, **25** (1), 137–171, 2008.

[13] J. Droniou, Remarks on discretizations of convection terms in hybrid mimetic mixed methods. *Networks and Heterogeneous Media*, **5** (3), 545–563, 2010.

[14] R. Eymard, T. Gallouet, R. Herbin, Discretization of heterogeneous and anisotropic diffusion problems on general nonconforming meshes SUSHI: a scheme using stabilization and hybrid interfaces. *IMA Journal of Numerical Analysis*, **30** (4), 1009–1043, 2010.

[15] H. Jasak, Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows. PhD thesis, Imperial College, 1996.

[16] Y. Kuznetsov, K. Lipnikov, M. Shashkov, The mimetic finite difference method on polygonal meshes for diffusion-type problems. *Computational Geosciences*, **8** (4), 301–324, 2004.

[17] K. Lipnikov, M. Shashkov, D. Svyatskiy, The mimetic finite difference discretization of diffusion problem on unstructured polyhedral meshes. *Journal of Computational Physics*, **211**, 473–491, 2005.

[18] K. Lipnikov, Mimetic finite difference method for solving PDEs on polygonal and polyhedral meshes. *International Workshop Non-Standard Numerical Methods for PDEs*, University of Pavia, Italy, June 29-July 2, 2010.

[19] K. Lipnikov, G. Manzini, F. Brezzi, A. Buffa, The mimetic finite difference method for the 3D magnetostatic field problems on polyhedral meshes. *Journal of Computational Physics*, **230** (2), 305–328, 2011.

[20] A. de Oliveira, S. Moraes, P. Laranjeira da Cunha Lage, G. Coelho Cunha, L.F. Lopes Rodrigues da Silva, Analysis of the non-orthogonality correction of Finite Volume Discretization on unstructured meshes. *22nd International Congress of Mechanical Engineering (COBEM 2013)*, November 3–7, 2013.

[21] L. Piar, F. Babik, R. Herbin, J.C. Latch, A formally second order cell centered scheme for convection-diffusion equations on unstructured non-conforming grids. *International Journal for Numerical Methods in Fluids*, **71**, 873–890, 2013.

[22] L.B. da Veiga, J. Droniou, G. Manzini, A unified approach to handle convection terms in Finite Volumes and Mimetic Discretization Methods for elliptic problems. *IMA Journal of Numerical Analysis*, **31** (4), 1357–1401, 2011.