THE HITCHHIKER'S GUIDE TO THE VIRTUAL ELEMENT METHOD

L. BEIRÃO DA VEIGA[1], F. BREZZI[2], L.D. MARINI[3], AND A. RUSSO[4]

ABSTRACT. We present the essential ingredients in the Virtual Element Method for a simple linear elliptic second-order problem. We emphasize its computer implementation, which will enable interested readers to readily implement the method.

"Don't Panic."
– Douglas Adams, The Hitchhiker's Guide to the Galaxy

1. INTRODUCTION

The aim of this paper is to present in detail the practical aspects of the implementation of the Virtual Element Method (VEM) [1, 5, 6, 8, 13, 17].

On the one hand, VEM can be viewed as an extension of Finite Element Methods to general polygonal and polyhedral elements [11, 21, 29–31, 33–39]. On the other hand, many of its aspects are closely connected with Finite Volume Methods (see for instance [19, 20, 28, 32]) and in particular with Mimetic Finite Differences (MFD) [2–4, 7, 10, 12, 14–16, 18, 22, 23, 25–27]. Indeed, for many problems – for instance in two dimensions and for C^0 elements – the final system matrices are identical for MFD and VEM. On particular meshes, this correspondence exists between traditional Finite Elements and Finite Differences.

The strongest aspects in favor of the VEM are its firm mathematical foundations, simplicity in implementation, and efficiency and accuracy in computations. In particular, for diffusion or reaction-diffusion problems in two and three dimensions, the VEM permits the analysis to be performed without using any numerical quadrature formulas.

We begin by considering the model problem of the Poisson equation in two dimensions:

$$\begin{cases} -\Delta u = f & \text{in } \Omega \\ \quad u = 0 & \text{on } \partial\Omega \end{cases} \tag{1.1}$$

where Ω is a polygonal domain in $\mathbb{R}^2$. We choose the simple problem posed in (1.1) so that the essential features in the implementation of the VEM can be easily explained. Unlike FEM that only allow elements that are either triangles or quadrilaterals, we herein admit the decomposition of Ω into nonoverlapping elements that can be of very general shape (convex or nonconvex polygons).

In Section 2, we introduce the local virtual element space within each element of the decomposition. This space contains a certain number of polynomials (sufficient to guarantee *accuracy*), plus a few other *additional functions* (typically the solution of a PDE within the element) that are however never required to be computed. In Section 3, we show how to construct a suitable local stiffness matrix (ensuring the necessary consistency and stability) without knowing these *additional functions* but by using only their degrees of freedom. A crucial ingredient in this operation is the construction of the operator Π^∇ (a projection operator, related to the bilinear form of the problem). We present the details on how to compute Π^∇ using only the degrees of freedom of the space. Once the operator Π^∇ is known, the local element stiffness matrices for problems such as (1.1) are readily constructed. In Section 4 we present two

[1] Dipartimento di Matematica, Università di Milano Statale, Via Saldini 50, I-20133 Milano (Italy); lourenco.beirao@unimi.it.

[2] IUSS and IMATI-CNR 27100 Pavia, Italy, and KAU, Jeddah, Saudi Arabia; brezzi@imati.cnr.it.

[4] Dipartimento di Matematica, Università di Pavia, and IMATI-CNR, 27100 Pavia (Italy); marini@imati.cnr.it.

[4] Dipartimento di Matematica e Applicazioni, Università di Milano–Bicocca, Via Cozzi 53, I-20153, Milano (Italy); alessandro.russo@unimib.it.

worked examples. We explicitly write the basic ingredients needed to compute the projection operator Π^∇ in the case of a square and of a pentagon.

In the second part of this paper, we shed light on more complex problems. In Section 5, we introduce the second, and possibly even more fundamental operator Π^0 – the L^2-projection on the space of polynomials of a certain degree. We show that by just changing *the definition* and *the perspective* for the local spaces $V_k(E)$ (but not the degrees of freedom and the construction and the use of the operator Π^∇), the construction of the new operator Π^0 is realized. The fundamental nature of the operator Π^0 stems from the fact that the knowledge of the L^2-projection of the *additional functions* is an extremely valuable instrument that is needed to solve more complex problems. We illustrate in Section 6 the use of Π^0 in the construction of load terms and mass matrix, with the latter arising due to the presence of a reaction term or for time-dependent problems. In Section 7 we discuss the use of the operator Π^0 on every face of a polyhedron, thereby paving the way for an efficient implementation of three-dimensional VEM. Finally, in Section 8, we provide a few additional remarks on the implementation of VEM for more general weak forms.

The general level of the presentation is intended for people not necessarily having a strong mathematical background, but with some experience in implementing the Finite Element Method. A few, more sophisticated remarks are indicated with a "$*$" and are not strictly necessary for the practical aspects of the VEM.

The authors wish to thank Professor Sukumar for his valuable comments and suggestions.

1.1. **Notation.**

If $\mathcal{D}$ is a subset of $\mathbb{R}^n$, we will denote by $\boldsymbol{x}_\mathcal{D}$, $h_\mathcal{D}$ and $|\mathcal{D}|$ the centroid, the diameter, and the measure of $\mathcal{D}$, respectively. The L^2 inner product between two functions u and v defined on $\mathcal{D}$ will be written $(u,v)_{0,\mathcal{D}}$. Let $\mathcal{P}_k(\mathcal{D})$ be the space of polynomials of degree less than or equal to k in $\mathcal{D}$. If $\mathcal{D} \subset \mathbb{R}^2$, we define

$$n_k := \dim \mathcal{P}_k(\mathcal{D}) = \frac{(k+1)(k+2)}{2},$$

while for $\mathcal{D} \subset \mathbb{R}^3$, we set

$$\nu_k := \dim \mathcal{P}_k(\mathcal{D}) = \frac{(k+1)(k+2)(k+3)}{6}. \tag{1.2}$$

In two dimensions a boldface Greek letter will indicate a multiindex: $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$ with the usual notation $|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2$. If $\boldsymbol{x} = (x_1, x_2)$, then $\boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} x_2^{\alpha_2}$. We will denote by $m_{\boldsymbol{\alpha}}$ the scaled monomial of degree equal to $|\boldsymbol{\alpha}|$ defined by

$$m_{\boldsymbol{\alpha}} := \left(\frac{\boldsymbol{x} - \boldsymbol{x}_\mathcal{D}}{h_\mathcal{D}} \right)^{\boldsymbol{\alpha}}. \tag{1.3}$$

We denote by $\mathcal{M}_k(\mathcal{D})$ the set of scaled monomials of degree less then or equal to k:

$$\mathcal{M}_k(\mathcal{D}) := \{ \, m_{\boldsymbol{\alpha}} \; : \; 0 \le |\boldsymbol{\alpha}| \le k \, \}.$$

It is clear that the set $\mathcal{M}_k(\mathcal{D})$ is a basis for $\mathcal{P}_k(\mathcal{D})$. A (non bold) greek letter α will indicate a one-dimensional index starting from 1 with the natural correspondence

$$1 \leftrightarrow (0,0), \quad 2 \leftrightarrow (1,0), \quad 3 \leftrightarrow (0,1), \quad 4 \leftrightarrow (2,0), \quad 5 \leftrightarrow (1,1), \quad \ldots \tag{1.4}$$

We will also write m_α instead of $m_{\boldsymbol{\alpha}}$. Note that with this convention $m_{(0,0)} = m_1 \equiv 1$. From now on, the statement "degree k" will always mean "degree less than or equal to k" unless explicitly indicated.

In three dimensions, the obvious changes of notation will apply. For $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$ we have $|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2 + \alpha_3$, and if $\boldsymbol{x} = (x_1, x_2, x_3)$ then $\boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} x_2^{\alpha_2} x_3^{\alpha_3}$. Always in three dimensions we will denote by $\mu_{\boldsymbol{\alpha}}$ the scaled monomial of degree equal to $|\boldsymbol{\alpha}|$ defined again by

$$\mu_{\boldsymbol{\alpha}}(\boldsymbol{x}) := \left(\frac{\boldsymbol{x} - \boldsymbol{x}_\mathcal{D}}{h_\mathcal{D}} \right)^{\boldsymbol{\alpha}}, \tag{1.5}$$

and we denote again by $\mathcal{M}_k(\mathcal{D})$ the set of scaled monomials of degree k. Finally, a (non bold) Greek letter α will still indicate a one-dimensional index starting from 1, and the natural correspondence this time goes as

$$1 \leftrightarrow (0,0,0), \quad 2 \leftrightarrow (1,0,0), \quad 3 \leftrightarrow (0,1,0), \quad 4 \leftrightarrow (0,0,1), \quad 5 \leftrightarrow (2,0,0), \quad \ldots$$

Remark 1.1. *On a domain $\widehat{\mathcal{D}}$, and for a given $\boldsymbol{\alpha}$, consider the function*

$$\hat{\psi}(\hat{\boldsymbol{x}}) := \left(\frac{\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}_{\widehat{\mathcal{D}}}}{h_{\widehat{\mathcal{D}}}}\right)^{\boldsymbol{\alpha}}.$$

Consider now the change of variable $\boldsymbol{x} = h\hat{\boldsymbol{x}}$ (for some given $h \neq 1$) from $\widehat{\mathcal{D}}$ to $\mathcal{D}$, mapping the function $\hat{\psi}$ into the function ψ given (as usual) by $\psi(\boldsymbol{x}) = \hat{\psi}(\hat{\boldsymbol{x}}) = \hat{\psi}(\boldsymbol{x}/h)$. Then it is not difficult to check that $h_{\mathcal{D}} = h\, h_{\widehat{\mathcal{D}}}$ and hence

$$\psi(\boldsymbol{x}) = \left(\frac{\boldsymbol{x} - \boldsymbol{x}_{\mathcal{D}}}{h_{\mathcal{D}}}\right)^{\boldsymbol{\alpha}}.$$

This explains why we called the elements of $\mathcal{M}_k(\mathcal{D})$ ***scaled monomials****.*

2. The local Virtual Element Space $V_k(E)$ in $2D$

We assume that $\Omega \subset \mathbb{R}^2$ has been partitioned into a collection $\mathcal{P}_h$ of non-overlapping polygons E, not necessarily convex: $\Omega = \cup_{E \in \mathcal{P}_h} E$.

For each polygon E, we will denote by V_i ($i = 1, \dots, N^V$) its vertices counterclockwise ordered, and by e_i the edge connecting V_i to V_{i+1}, as shown in Fig. 1; the dependence on E will be always omitted when no confusion can arise.

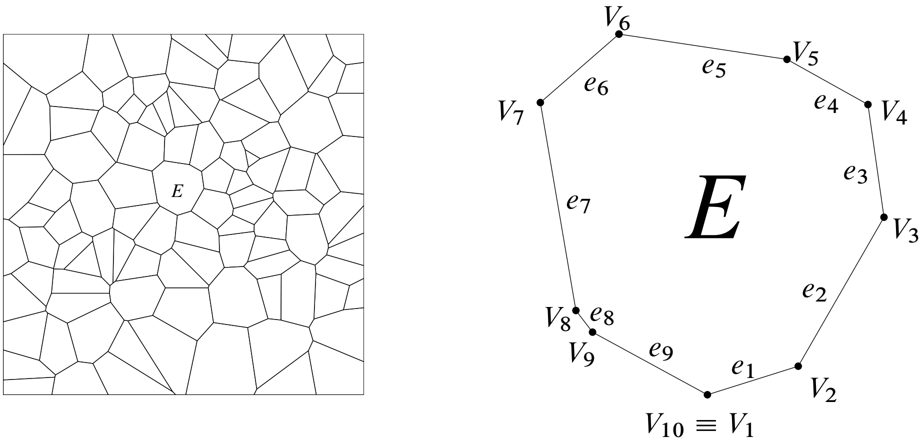

FIGURE 1. A random polygonal mesh $\mathcal{P}_h$ for the unit square and a typical polygon E

Remark 2.1. *We point out that two consecutive edges are allowed to form a straight angle (180 degrees). For example, a square with an extra point on each edge is simply regarded as an octagon, as shown in Fig. 2. In other words,* hanging nodes are allowed geometrically *in the mesh, but in practice they are not really hanging.*

For each polygon E we define a local finite element space $V_k(E)$. Roughly speaking, $V_k(E)$ contains all polynomials of degree k (which is essential for convergence) plus other functions whose restriction on an edge is still a polynomial of degree k. Following [5], a function $v_h \in V_k(E)$ is defined by the following properties:

$$\begin{cases} i)\ v_h \text{ is a polynomial of degree } k \text{ on each edge } e \text{ of } E, \text{ i.e., } v_{h|e} \in \mathcal{P}_k(e); \\ ii)\ v_h \text{ on } \partial E \text{ is globally continuous, i.e., } v_{h|\partial E} \in C^0(\partial E); \\ iii)\ \Delta v_h \text{ is a polynomial of degree } k-2 \text{ in } E, \text{ i.e., } \Delta v_h \in \mathcal{P}_{k-2}(E). \end{cases}$$

Remark 2.2. *Note that a polynomial of degree k satisfies conditions i), ii), and iii) so that $\mathcal{P}_k(E)$ is a subspace of $V_k(E)$.*

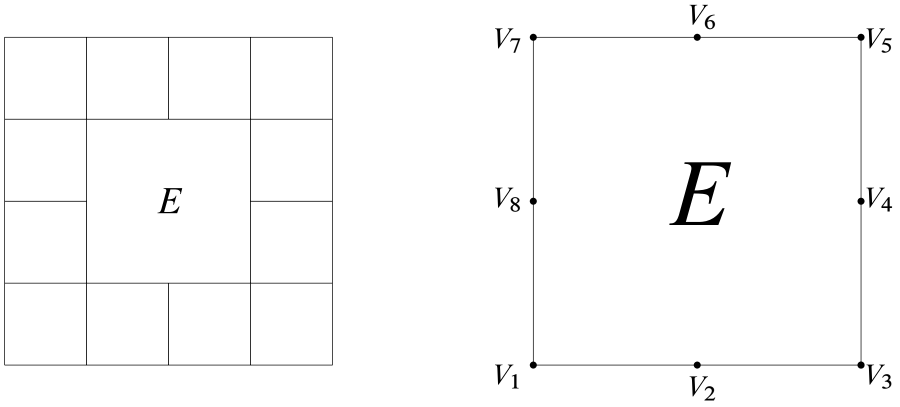

FIGURE 2. Hanging nodes are allowed: the central square is treated as an octagon

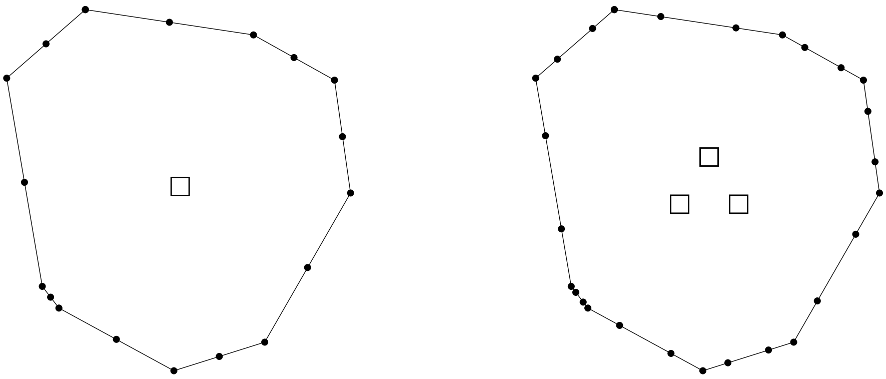

FIGURE 3. Boundary (•) and internal (□) degrees of freedom for $k = 2$ (left) and $k = 3$ (right)

In [5] it is shown that we can take the following degrees of freedom in $V_k(E)$:

$$\left\{\begin{array}{l} \bullet \text{ the value of } v_h \text{ at the vertices of } E \\ \bullet \text{ on each edge } e\text{, the value of } v_h \text{ at the } k-1 \text{ internal points of} \\ \quad \text{the } (k+1)\text{-point Gauss-Lobatto quadrature rule on } e; \\ \bullet \text{ the moments up to order } k-2 \text{ of } v_h \text{ in } E: \\ \qquad \dfrac{1}{|E|}\displaystyle\int_E v_h\, m_\alpha, \quad \alpha = 1, \dots, n_{k-2} \\ \quad \text{where the scaled monomials } m_\alpha \text{ are defined in (1.3) and (1.4)} \\ \quad \text{and } n_{k-2} = \dim \mathcal{P}_{k-2}(E). \end{array}\right. \tag{2.1}$$

As a consequence, the dimension of $V_k(E)$ is

$$\dim V_k(E) = N^V + N^V(k-1) + n_{k-2} = N^V k + \frac{(k-1)k}{2},$$

where we recall that N^V is the number of vertices (and edges) of the polygon E. We will refer to the first and to the second set of degrees of freedom as *boundary degrees of freedom*, and to the third set as *internal degrees of freedom*, see Fig. 3.

Remark 2.3. *There are no internal degrees of freedom in the case $k = 1$. In this case, the space $V_1(E)$ corresponds to the well-known polygonal finite element method with harmonic barycentric coordinates (see [33]). However, as we shall see, the construction of the stiffness matrix for the VEM is different.*

Remark 2.4. *The choice of the internal Gauss-Lobatto quadrature points on each edge is not mandatory; in fact, we could have used the uniformly spaced points or even the (scaled) moments up to degree $k-2$, or more generally, any set of parameters that, together with the vertex values, identify a unique* *polynomial of degree k on each edge. Our option has the obvious advantage that we can compute the integral of a polynomial of degree $2k-1$ on each edge directly from its $k+1$ degrees of freedom on that edge, and this feature will greatly simplify the exposition and the actual implementation of the method.*

Remark 2.5. *In the framework of Remark 1.1, assume again that we have a domain $\widehat{\mathcal{D}}$ and the change of variable $\boldsymbol{x} = h\hat{\boldsymbol{x}}$ (for some given $h \neq 1$) that maps the domain $\widehat{\mathcal{D}}$ into the domain $\mathcal{D}$. Assume now that for a given $\hat{\psi}$ and a given $\boldsymbol{\alpha}$ we have*

$$\frac{1}{|\widehat{\mathcal{D}}|}\int_{\widehat{\mathcal{D}}} \hat{\psi}(\hat{\boldsymbol{x}}) \left(\frac{\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}_{\widehat{\mathcal{D}}}}{h_{\widehat{\mathcal{D}}}}\right)^{\boldsymbol{\alpha}} \mathrm{d}\hat{\boldsymbol{x}} = 1.$$

Then it is not difficult to check that, setting again $\psi(\boldsymbol{x}) = \hat{\psi}(\hat{\boldsymbol{x}}) \equiv \hat{\psi}(\boldsymbol{x}/h)$ for all $\boldsymbol{x} \in \mathcal{D}$, we also have

$$\frac{1}{|\mathcal{D}|}\int_{\mathcal{D}} \psi(\boldsymbol{x}) \left(\frac{\boldsymbol{x} - \boldsymbol{x}_{\mathcal{D}}}{h_{\mathcal{D}}}\right)^{\boldsymbol{\alpha}} \mathrm{d}\boldsymbol{x} = 1.$$

We express this property by saying that the degrees of freedom

$$\psi \;\mapsto\; \frac{1}{|\mathcal{D}|}\int_{\mathcal{D}} \psi(\boldsymbol{x}) \left(\frac{\boldsymbol{x} - \boldsymbol{x}_{\mathcal{D}}}{h_{\mathcal{D}}}\right)^{\boldsymbol{\alpha}} \mathrm{d}\boldsymbol{x}$$

scale as *1. We notice that all the degrees of freedom in* (2.1) ***scale as*** *1.*

We number the degrees of freedom from 1 to $N^{\text{dof}} := \dim V_k(E)$ and we define the operator dof_i from $V_k(E)$ to $\mathbb{R}$ as

$$\text{dof}_i(v_h) := i\text{-th degree of freedom of } v_h, \quad i = 1, \dots, N^{\text{dof}}.$$

The *basis functions* $\varphi_i \in V_k(E)$ are defined as usual as the canonical basis functions:

$$\text{dof}_i(\varphi_j) = \delta_{ij}, \quad i, j = 1, \dots, N^{\text{dof}}, \tag{2.2}$$

so that we have a Lagrange-type interpolation identity:

$$v_h = \sum_{i=1}^{N^{\text{dof}}} \text{dof}_i(v_h)\,\varphi_i \quad \text{for all } v_h \in V_k(E). \tag{2.3}$$

The boundary degrees of freedom identify on each edge a unique polynomial of degree k. Hence, we can define the global finite element space $V_h \subset H_0^1(\Omega)$ as

$$V_h := \{v_h \in H_0^1(\Omega) \quad : \quad v_{h|E} \in V_k(E) \quad \text{for all } E \in \mathcal{P}_h\}$$

with the following global degrees of freedom for v_h:

$$\begin{cases} \bullet \text{ the value of } v_h \text{ at the internal vertices of the decomposition;} \\ \bullet \text{ on each internal edge } e, \text{ the value of } v_h \text{ at the } k-1 \text{ internal points of} \\ \quad \text{the } (k+1)\text{-point Gauss-Lobatto quadrature rule on } e; \\ \bullet \text{ for each polygon } E, \text{ the moments up to order } k-2 \text{ of } v_h \text{ in } E: \\ \qquad \dfrac{1}{|E|}\displaystyle\int_E v_h\, m_\alpha, \quad \alpha = 1, \dots, n_{k-2}. \end{cases} \tag{2.4}$$

3. Computation of the local stiffness matrix

We want to compute the local stiffness matrix $\mathbf{K}_E$ of the Laplace operator in the polygon E, i.e.,

$$\left(\mathbf{K}_E\right)_{ij} = \left(\nabla\varphi_i, \nabla\varphi_j\right)_{0,E}, \quad i, j = 1, \dots, N^{\text{dof}}, \tag{3.1}$$

where $\varphi_i \in V_k(E)$ is defined by (2.2). An obvious but very expensive way would be to choose a quadrature formula on E, and then use some approximation for the values of the gradient of the basis functions φ_i at the integration nodes in order to compute an approximation of $(\mathbf{K}_E)_{ij}$. The VEM approach is completely different: it does not require neither the use of quadrature formulas nor an approximate expression of the basis functions. Nevertheless, the final output will be a matrix $\mathbf{K}_E^h$ whose associated bilinear form is *exact* (up to machine precision) whenever one of the two entries is a polynomial of degree k. This will allow to retain the optimal approximation properties of the space $V_k(E)$ (see [5]).

The explicit computation of the basis functions φ_i is actually not needed and this is the reason of the word "Virtual" in VEM.

3.1. **The projection operator Π^∇.** We begin by defining a *projection operator*

$$\Pi^\nabla_{E,k} : V_k(E) \longrightarrow \mathcal{P}_k(E)$$

which will play a fundamental role in the rest of the paper. For the sake of simplicity, the subscripts E and/or k will be omitted when no confusion can arise.

The operator Π^∇ is defined for every $v_h \in V_k(E)$ by the following orthogonality condition:

$$\left(\nabla p_k, \nabla\left(\Pi^\nabla v_h - v_h\right)\right)_{0,E} = 0 \quad \text{for all } p_k \in \mathcal{P}_k(E). \tag{3.2}$$

As it can be easily seen, condition (3.2) defines $\Pi^\nabla v_h$ only up to a constant; this is fixed by prescribing a projection operator onto constants $\mathrm{P}_0 : V_k(E) \longrightarrow \mathcal{P}_0(E)$ and requiring

$$\mathrm{P}_0\big(\Pi^\nabla v_h - v_h\big) = 0. \tag{3.3}$$

Many options are possible for P_0. Here we choose

$$\mathrm{P}_0 v_h := \frac{1}{N^V}\sum_{i=1}^{N^V} v_h(V_i) \quad \text{for } k = 1 \tag{3.4a}$$

$$\mathrm{P}_0 v_h := \frac{1}{|E|}\int_E v_h \quad \text{for } k \geq 2. \tag{3.4b}$$

The reasons for this choice will be explained later on. We will now show how to actually compute $\Pi^\nabla v_h$ for a given $v_h \in V_k(E)$ using only the degrees of freedom of v_h. Since $\mathcal{M}_k(E)$ is a basis for $\mathcal{P}_k(E)$, in equation (3.2) we can let p_k vary only in $\mathcal{M}_k(E)$:

$$\left(\nabla m_\alpha, \nabla\left(\Pi^\nabla v_h - v_h\right)\right)_{0,E} = 0, \quad \alpha = 1, \dots n_k, \tag{3.5}$$

and since $\Pi^\nabla v_h$ is an element of $\mathcal{P}_k(E)$ we can represent it in the basis $\mathcal{M}_k(E)$:

$$\Pi^\nabla v_h = \sum_{\beta=1}^{n_k} s^\beta m_\beta. \tag{3.6}$$

* **Remark 3.1.** *The use of the scaled monomials $\mathcal{M}_k(E)$ is essential in the definition of the degrees of freedom* (2.1)*. On the contrary, in* (3.6) *and in the sequel we could use any basis for $\mathcal{P}_k(E)$. Nevertheless, for the sake of simplicity we will always keep using the basis $\mathcal{M}_k(E)$.*

Equation (3.5) becomes:

$$\sum_{\beta=1}^{n_k} s^\beta \left(\nabla m_\alpha, \nabla m_\beta\right)_{0,E} = \left(\nabla m_\alpha, \nabla v_h\right)_{0,E}, \quad \alpha = 1, \dots n_k \tag{3.7}$$

which is a linear system of n_k equations in the n_k unknowns $s^\beta = s^\beta(v_h)$. The indeterminacy of (3.2) is reflected here in the fact that for $\alpha = 1$, which corresponds to $m_\alpha \equiv 1$, equation (3.7) is the identity $0 = 0$. Condition (3.3) adds one linear equation that eliminates this indeterminacy:

$$\sum_{\beta=1}^{n_k} s^\beta\, \mathrm{P}_0 m_\beta = \mathrm{P}_0 v_h. \tag{3.8}$$

The linear system arising from (3.7) and (3.8) can then be written as follows:

$$\begin{bmatrix} \mathrm{P}_0 m_1 & \mathrm{P}_0 m_2 & \dots & \mathrm{P}_0 m_{n_k} \\ 0 & (\nabla m_2, \nabla m_2)_{0,E} & \dots & (\nabla m_2, \nabla m_{n_k})_{0,E} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & (\nabla m_{n_k}, \nabla m_2)_{0,E} & \dots & (\nabla m_{n_k}, \nabla m_{n_k})_{0,E} \end{bmatrix} \begin{bmatrix} s^1 \\ s^2 \\ \vdots \\ s^{n_k} \end{bmatrix} = \begin{bmatrix} \mathrm{P}_0 v_h \\ (\nabla m_2, \nabla v_h)_{0,E} \\ \vdots \\ (\nabla m_{n_k}, \nabla v_h)_{0,E} \end{bmatrix},$$

or, in compact way,

$$\mathbf{G}\,\underline{s} = \underline{b} \tag{3.9}$$

where

$$\mathbf{G} := \begin{bmatrix} \mathrm{P}_0 m_1 & \mathrm{P}_0 m_2 & \dots & \mathrm{P}_0 m_{n_k} \\ 0 & (\nabla m_2, \nabla m_2)_{0,E} & \dots & (\nabla m_2, \nabla m_{n_k})_{0,E} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & (\nabla m_{n_k}, \nabla m_2)_{0,E} & \dots & (\nabla m_{n_k}, \nabla m_{n_k})_{0,E} \end{bmatrix} \tag{3.10}$$

and

$$\underline{b} := \begin{bmatrix} \mathrm{P}_0 v_h \\ (\nabla m_2, \nabla v_h)_{0,E} \\ \vdots \\ (\nabla m_{n_k}, \nabla v_h)_{0,E} \end{bmatrix}. \tag{3.11}$$

The matrix $\mathbf{G}$ in (3.10) is computable because we assume to be able to compute integrals of polynomials on E. Hence, to compute $\Pi^\nabla v_h$, we only need to show that the right-hand side of (3.9) (that is, the vector $\underline{b}$ in (3.11)) is computable from of the degrees of freedom of v_h.

By the definitions (3.4a) and (3.4b) for the projector P_0, it is clear that in both cases, $k = 1$ and $k \geq 2$, we can compute $\mathrm{P}_0 v_h$ (the first component of the right-hand side (3.11)) from the degrees of freedom of v_h. Concerning the other components of (3.11), we have, integrating by parts:

$$(\nabla m_\alpha, \nabla v_h)_{0,E} = -\int_E \Delta m_\alpha v_h + \int_{\partial E} \frac{\partial m_\alpha}{\partial n} v_h. \tag{3.12}$$

We examine separately the two terms.

Since $\Delta m_\alpha \in \mathcal{P}_{k-2}(E)$, the first term can be computed from the internal degrees of freedom of v_h. In fact, writing Δm_α as a linear combination of the m_β,

$$\Delta m_\alpha = \sum_{\beta=1}^{n_{k-2}} d_\alpha^\beta m_\beta,$$

we have

$$-\int_E \Delta m_\alpha v_h = -\sum_{\beta=1}^{n_{k-2}} d_\alpha^\beta \int_E m_\beta v_h = -|E| \sum_{\beta=1}^{n_{k-2}} d_\alpha^\beta \operatorname{dof}_{(kN^V+\beta)}(v_h).$$

Concerning the second term, we observe that the integrand is a polynomial of degree $(k-1)+k = 2k-1$ on each edge e so it can be integrated exactly by evaluating it at the Gauss-Lobatto quadrature points (including the extrema of the edge). This can be done easily because the values of v_h at these points are precisely the boundary degrees of freedom of v_h.

Remark 3.2. *We note that in the calculation of the two terms in* (3.12)*, and thus in the construction of the above matrices and vectors, we never used the requirement that functions in $V_k(E)$ have the Laplacian in $\mathcal{P}_{k-2}(E)$. The reason for this assumption is mainly to fix the dimension of the space $V_k(E)$.*

3.2. **Computation of $\Pi^\nabla \varphi_i$.** For each basis function φ_i, we define s_i^α as the coefficients of $\Pi^\nabla \varphi_i$ in the basis m_α:

$$\Pi^\nabla \varphi_i = \sum_{\alpha=1}^{n_k} s_i^\alpha m_\alpha, \quad i = 1, \dots N^{\mathrm{dof}}. \tag{3.13}$$

The coefficients s_i^α are solutions of the system (3.9) with φ_i in place of v_h in the right-hand side:

$$\begin{bmatrix} \mathrm{P}_0 m_1 & \mathrm{P}_0 m_2 & \dots & \mathrm{P}_0 m_{n_k} \\ 0 & (\nabla m_2, \nabla m_2)_{0,E} & \dots & (\nabla m_2, \nabla m_{n_k})_{0,E} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & (\nabla m_{n_k}, \nabla m_2)_{0,E} & \dots & (\nabla m_{n_k}, \nabla m_{n_k})_{0,E} \end{bmatrix} \begin{bmatrix} s_i^1 \\ s_i^2 \\ \vdots \\ s_i^{n_k} \end{bmatrix} = \begin{bmatrix} \mathrm{P}_0 \varphi_i \\ (\nabla m_2, \nabla \varphi_i)_{0,E} \\ \vdots \\ (\nabla m_{n_k}, \nabla \varphi_i)_{0,E} \end{bmatrix},$$

or, in compact form,

$$\underline{s}^{(i)} = \mathbf{G}^{-1} \underline{b}^{(i)}.$$

Denoting by $\mathbf{B}$ the $n_k \times N^{\mathrm{dof}}$ matrix given by

$$\mathbf{B} := \left[\underline{b}^{(1)}\ \underline{b}^{(2)}\ \ldots\ \underline{b}^{(N^{\mathrm{dof}})}\right] = \begin{bmatrix} \mathrm{P}_0\varphi_1 & \ldots & \mathrm{P}_0\varphi_{N^{\mathrm{dof}}} \\ \left(\nabla m_2, \nabla\varphi_1\right)_{0,E} & \ldots & \left(\nabla m_2, \nabla\varphi_{N^{\mathrm{dof}}}\right)_{0,E} \\ \vdots & \ddots & \vdots \\ \left(\nabla m_{n_k}, \nabla\varphi_1\right)_{0,E} & \ldots & \left(\nabla m_{n_k}, \nabla\varphi_{N^{\mathrm{dof}}}\right)_{0,E} \end{bmatrix}, \tag{3.14}$$

the matrix representation $\mathbf{\Pi}^{\nabla}_{*}$ of the operator Π^{∇} acting from $V_k(E)$ to $\mathcal{P}_k(E)$ in the basis $\mathcal{M}_k(E)$ is given by $(\mathbf{\Pi}^{\nabla}_{*})_{\alpha i} = s_i^{\alpha}$, that is,

$$\mathbf{\Pi}^{\nabla}_{*} = \mathbf{G}^{-1}\mathbf{B}. \tag{3.15}$$

We will also need the matrix representation, in the canonical basis (2.2), of the same operator Π^{∇}, this time thought as an operator $V_k(E) \longrightarrow V_k(E)$. Hence, let

$$\Pi^{\nabla}\varphi_i = \sum_{j=1}^{N^{\mathrm{dof}}} \pi_i^j \varphi_j, \quad i = 1, \ldots N^{\mathrm{dof}},$$

with

$$\pi_i^j = \mathrm{dof}_j\left(\Pi^{\nabla}\varphi_i\right).$$

From (3.13) and (2.3) we have

$$\Pi^{\nabla}\varphi_i = \sum_{\alpha=1}^{n_k} s_i^{\alpha} m_{\alpha} = \sum_{\alpha=1}^{n_k} s_i^{\alpha} \sum_{j=1}^{N^{\mathrm{dof}}} \mathrm{dof}_j(m_{\alpha})\,\varphi_j$$

so that

$$\pi_i^j = \sum_{\alpha=1}^{n_k} s_i^{\alpha}\,\mathrm{dof}_j(m_{\alpha}). \tag{3.16}$$

In order to express (3.16) in matrix form, we define the $N^{\mathrm{dof}} \times n_k$ matrix $\mathbf{D}$ by:

$$\mathbf{D}_{i\alpha} := \mathrm{dof}_i(m_{\alpha}), \quad i = 1, \ldots, N^{\mathrm{dof}}, \quad \alpha = 1, \ldots, n_k,$$

that is,

$$\mathbf{D} = \begin{bmatrix} \mathrm{dof}_1(m_1) & \mathrm{dof}_1(m_2) & \ldots & \mathrm{dof}_1(m_{n_k}) \\ \mathrm{dof}_2(m_1) & \mathrm{dof}_2(m_2) & \ldots & \mathrm{dof}_2(m_{n_k}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{dof}_{N^{\mathrm{dof}}}(m_1) & \mathrm{dof}_{N^{\mathrm{dof}}}(m_2) & \ldots & \mathrm{dof}_{N^{\mathrm{dof}}}(m_{n_k}) \end{bmatrix}. \tag{3.17}$$

Equation (3.16) becomes:

$$\pi_i^j = \sum_{\alpha=1}^{n_k} (\mathbf{G}^{-1}\mathbf{B})_{\alpha i}\mathbf{D}_{j\alpha} = (\mathbf{D}\mathbf{G}^{-1}\mathbf{B})_{ji}.$$

Hence, the *matrix representation* $\mathbf{\Pi}^{\nabla}$ of the operator $\Pi^{\nabla} : V_k(E) \longrightarrow V_k(E)$ in the canonical basis (2.2), is given by

$$\mathbf{\Pi}^{\nabla} = \mathbf{D}\mathbf{G}^{-1}\mathbf{B} = \mathbf{D}\mathbf{\Pi}^{\nabla}_{*}. \tag{3.18}$$

Remark 3.3. *We point out that* $\mathbf{G}$ *can be expressed in terms of* $\mathbf{D}$ *and* $\mathbf{B}$ *as*

$$\mathbf{G} = \mathbf{B}\mathbf{D}. \tag{3.19}$$

In fact:

- *for* $\alpha = 1$*:*

$\sum_{i=1}^{N^{\mathrm{dof}}} \mathbf{B}_{1i}\mathbf{D}_{i\beta} = \sum_{i=1}^{N^{\mathrm{dof}}} P_0\varphi_i\,\mathrm{dof}_i(m_\beta) = P_0\big(\sum_{i=1}^{N^{\mathrm{dof}}} \mathrm{dof}_i(m_\beta)\varphi_i\big) = P_0(m_\beta) = \mathbf{G}_{1\beta};$

- *for* $\alpha \geq 2$*:*

$$\sum_{i=1}^{N^{\mathrm{dof}}} \mathbf{B}_{\alpha i}\mathbf{D}_{i\beta} = \sum_{i=1}^{N^{\mathrm{dof}}} (\nabla m_\alpha, \nabla\varphi_i)_{0,E}\,\mathrm{dof}_i(m_\beta) = \\ \big(\nabla m_\alpha, \nabla\big(\sum_{i=1}^{N^{\mathrm{dof}}} \mathrm{dof}_i(m_\beta)\varphi_i\big)\big)_{0,E} = \big(\nabla m_\alpha, \nabla m_\beta\big)_{0,E} = \mathbf{G}_{\alpha\beta},$$

and we can conclude that $\mathbf{G} = \mathbf{BD}$. *In the implementation of the VEM it might be convenient to compute directly the matrix* $\mathbf{G}$ *by* (3.10) *and then use identity* (3.19) *to check the correctness of the code.*

3.3. **Construction of the local stiffness matrix.** At this point we can write the VEM local stiffness matrix $\mathbf{K}_E^h$ for the polygon E. Using the projector Π^∇, we write φ_i as

$$\varphi_i = \Pi^\nabla \varphi_i + (\mathrm{I} - \Pi^\nabla)\varphi_i$$

and we plug it into (3.1) obtaining, after expansion,

$$(\mathbf{K}_E)_{ij} = \left(\nabla \Pi^\nabla \varphi_i, \nabla \Pi^\nabla \varphi_j\right)_{0,E} + \left(\nabla\left(\mathrm{I} - \Pi^\nabla\right)\varphi_i, \nabla\left(\mathrm{I} - \Pi^\nabla\right)\varphi_j\right)_{0,E} + \\ \left(\nabla \Pi^\nabla \varphi_i, \nabla\left(\mathrm{I} - \Pi^\nabla\right)\varphi_j\right)_{0,E} + \left(\nabla\left(\mathrm{I} - \Pi^\nabla\right)\varphi_i, \nabla \Pi^\nabla \varphi_j\right)_{0,E}.$$

Since the last two terms are zero by the definition of Π^∇, we obtain the following expression for $\mathbf{K}_E$:

$$(\mathbf{K}_E)_{ij} = (\nabla \Pi^\nabla \varphi_i, \nabla \Pi^\nabla \varphi_j)_{0,E} + \left(\nabla\left(\mathrm{I} - \Pi^\nabla\right)\varphi_i, \nabla\left(\mathrm{I} - \Pi^\nabla\right)\varphi_j\right)_{0,E}. \tag{3.20}$$

The first term ensures *consistency* and must be computed exactly, while the second term ensures *stability* and can be approximated. In [5] it is shown that we can take for the second term the following rough approximation

$$\left(\nabla\left(\mathrm{I} - \Pi^\nabla\right)\varphi_i, \nabla\left(\mathrm{I} - \Pi^\nabla\right)\varphi_j\right)_{0,E} \approx \sum_{r=1}^{N^{\mathrm{dof}}} \mathrm{dof}_r\left(\left(\mathrm{I} - \Pi^\nabla\right)\varphi_i\right) \mathrm{dof}_r\left(\left(\mathrm{I} - \Pi^\nabla\right)\varphi_j\right), \tag{3.21}$$

still retaining the optimal approximation properties of the space $V_k(E)$.

* **Remark 3.4.** *We recall that, as pointed out in Remark 2.5, the degrees of freedom* (2.1) *and hence the corresponding basis functions scale as 1. Moreover, it is easy to check that in two dimensions the integral of* $\nabla\varphi_i \cdot \nabla\varphi_j$ *on a domain of size* $\sim h^2$ *also scales as 1. This is the reason why in the approximation* (3.21) *we do not need any multiplication by suitable powers of h. This will not be the case in other important situations such as fourth order problems in 2D or second order problems in 3D; see also Remark 6.2.*

We are now ready to write the explicit expression of the local stiffness matrix of the Virtual Element Method:

$$(\mathbf{K}_E^h)_{ij} := (\nabla \Pi^\nabla \varphi_i, \nabla \Pi^\nabla \varphi_j)_{0,E} + \sum_{r=1}^{N^{\mathrm{dof}}} \mathrm{dof}_r\left(\left(\mathrm{I} - \Pi^\nabla\right)\varphi_i\right) \mathrm{dof}_r\left(\left(\mathrm{I} - \Pi^\nabla\right)\varphi_j\right). \tag{3.22}$$

From equation (3.13) we have

$$(\nabla \Pi^\nabla \varphi_i, \nabla \Pi^\nabla \varphi_j)_{0,E} = \sum_{\alpha=1}^{n_k}\sum_{\beta=1}^{n_k} s_i^\alpha s_j^\beta \,(\nabla m_\alpha, \nabla m_\beta)_{0,E} = \\ \sum_{\alpha=1}^{n_k}\sum_{\beta=1}^{n_k} (\mathbf{\Pi}_*^\nabla)_{\alpha i}(\mathbf{\Pi}_*^\nabla)_{\beta j}\widetilde{\mathbf{G}}_{\alpha\beta} = [(\mathbf{\Pi}_*^\nabla)^{\mathrm{T}}\widetilde{\mathbf{G}}(\mathbf{\Pi}_*^\nabla)]_{ij} \tag{3.23}$$

where $\widetilde{\mathbf{G}}$ is the matrix that coincides with $\mathbf{G}$ (see (3.10)) except for the first row which is set to zero, and $\mathbf{\Pi}_*^\nabla$ is defined in (3.15). From the computation above we have

$$\mathrm{dof}_r\left(\left(\mathrm{I} - \Pi^\nabla\right)\varphi_i\right) = \left[(\mathbf{I} - \mathbf{\Pi}^\nabla)^{\mathrm{T}}\right]_{ir}$$

and finally

$$\sum_{r=1}^{N^{\mathrm{dof}}} \mathrm{dof}_r\left(\left(\mathrm{I} - \Pi^\nabla\right)\varphi_i\right) \mathrm{dof}_r\left(\left(\mathrm{I} - \Pi^\nabla\right)\varphi_j\right) = \left[(\mathbf{I} - \mathbf{\Pi}^\nabla)^{\mathrm{T}}(\mathbf{I} - \mathbf{\Pi}^\nabla)\right]_{ij}. \tag{3.24}$$

We end up with the following matrix expression for the VEM local stiffness matrix:

$$\boxed{\mathbf{K}^h_E = (\mathbf{\Pi}^\nabla_*)^{\mathrm{T}}\widetilde{\mathbf{G}}(\mathbf{\Pi}^\nabla_*) + (\mathbf{I} - \mathbf{\Pi}^\nabla)^{\mathrm{T}}(\mathbf{I} - \mathbf{\Pi}^\nabla)}. \tag{3.25}$$

Remark 3.5. *The matrices* $\mathbf{B}, \mathbf{D}$ *and* $\mathbf{G} = \mathbf{BD}$ *depend only on the shape of the polygon* E *and not on its size. Therefore, any transformation that preserves the shape of* E *will have no effect on the matrices* $\mathbf{B}, \mathbf{D}, \mathbf{G}$ *and thus also on the stiffness matrix* $\mathbf{K}^h_E$*. Note that this same observation does not apply, for instance, in three dimensions or for higher order problems.*

* **Remark 3.6.** *As shown in [5], we can multiply the stabilization term* (3.24) *by a factor* σ_E *and still achieve the optimal convergence results if*

$$0 < \sigma_* \le \sigma_E \le \sigma^*$$

where the two constants σ_* *and* σ^* *are independent of* h.

* **Remark 3.7.** *The VEM stiffness matrix* $\mathbf{K}^h_E$ *in general is* not *close to the exact stiffness matrix* $\mathbf{K}_E$*; actually* $(\mathbf{K}^h_E)_{ij}$ *is* not *meant to be an approximation of* $(\mathbf{K}_E)_{ij}$ *in the standard sense. This notwithstanding, thanks to the first term in* (3.20)*, the VEM solution retains the optimal accuracy of the space* V_h.

3.4. **The case** $k = 1$**.** In the case $k = 1$ we can easily give an “intrinsic” formula for $\Pi^\nabla v_h$. Since the gradient of a polynomial of degree 1 is a constant vector, equation (3.2) becomes

$$|E|\,\nabla p_1 \cdot \nabla(\Pi^\nabla v_h) = \nabla p_1 \cdot \int_E \nabla v_h, \tag{3.26}$$

and by taking $p_1 = x_1$, $p_1 = x_2$ we easily see that (3.26) is equivalent to

$$\mathbf{g}(v_h) := \nabla(\Pi^\nabla v_h) = \frac{1}{|E|}\int_E \nabla v_h. \tag{3.27}$$

Hence,

$$\Pi^\nabla v_h = \mathbf{x}\cdot\mathbf{g}(v_h) + c, \tag{3.28}$$

where c is a constant function that depends on v_h. From this expression we can readily compute the *consistency* term of the VEM stiffness matrix (3.22). In fact, by (3.27), equation (3.23) in this case becomes

$$\left(\nabla\Pi^\nabla\varphi_i, \nabla\Pi^\nabla\varphi_j\right)_{0,E} = |E|\,\mathbf{g}(\varphi_i)\cdot\mathbf{g}(\varphi_j).$$

Since φ_i is linear on each edge, it is easily seen that

$$|E|\,\mathbf{g}(\varphi_i) \equiv \int_E \nabla\varphi_i = \frac{1}{2}\left(|e_{i-1}|\,\boldsymbol{n}_{i-1} + |e_i|\,\boldsymbol{n}_i\right) = \frac{1}{2}\mathbf{d}_i^\perp \tag{3.29}$$

where $\mathbf{d}_i$ is the vector $V_{i+1} - V_{i-1}$ and the $\perp$ superscript means a counterclockwise rotation of 90 degrees. Hence,

$$\left(\nabla\Pi^\nabla\varphi_i, \nabla\Pi^\nabla\varphi_j\right)_{0,E} = \frac{1}{4|E|}\mathbf{d}_i^\perp\cdot\mathbf{d}_j^\perp = \frac{1}{4|E|}\mathbf{d}_i\cdot\mathbf{d}_j.$$

In order to obtain the stability term of the VEM stiffness matrix (3.22), we need to compute the degrees of freedom of $\Pi^\nabla\varphi_i$, which in the case $k = 1$ reduce to the values of $\Pi^\nabla\varphi_i$ at the vertices of the polygon E. For this, we need to know the constant c in (3.28). Applying (3.3) we obtain:

$$\mathrm{P}_0(\Pi^\nabla v_h) \equiv \mathrm{P}_0(\boldsymbol{x})\cdot\mathbf{g}(v_h) + \mathrm{P}_0 c = \mathrm{P}_0 v_h. \tag{3.30}$$

We obviously have $\mathrm{P}_0 c = c$; recalling (3.4a) and denoting by $\overline{V}$ and $\overline{v}_h$, respectively, the coordinates of the *vertex center* and the *mean nodal value of* v_h, given by

$$\overline{V} := \mathrm{P}_0(\boldsymbol{x}) = \frac{1}{N^V}\sum_{i=1}^{N^V} V_i \qquad \text{and} \qquad \overline{v}_h := \mathrm{P}_0 v_h = \frac{1}{N^V}\sum_{i=1}^{N^V} v_h(V_i)$$

we easily have from (3.30)

$$c = \mathrm{P}_0 v_h - \mathrm{P}_0(\boldsymbol{x})\cdot\mathbf{g}(v_h) = \overline{v}_h - \overline{V}\cdot\mathbf{g}(v_h),$$

giving

$$\Pi^\nabla v_h = (\mathbf{x} - \overline{V}) \cdot \mathbf{g}(v_h) + \overline{v}_h.$$

Applying this formula to $v_h = \varphi_i$, using (3.29), and since $\overline{\varphi}_i = 1/N^V$, we end up with the following formula:

$$\Pi^\nabla \varphi_i = \frac{1}{2|E|}\,(\mathbf{x} - \overline{V}) \cdot \mathbf{d}_i^\perp + \frac{1}{N^V}.$$

Hence,

$$\left(\Pi^\nabla\right)_{ri} = \mathrm{dof}_r(\Pi^\nabla \varphi_i) = \left(\Pi^\nabla \varphi_i\right)(V_r) = \frac{1}{2|E|}\,(V_r - \overline{V}) \cdot \mathbf{d}_i^\perp + \frac{1}{N^V} \tag{3.31}$$

so that

$$\left(\mathrm{I} - \Pi^\nabla\right)_{ri} = \mathrm{dof}_r\left(\left(\mathrm{I} - \Pi^\nabla\right)\varphi_i\right) = \left(\delta_{ir} - \frac{1}{N^V}\right) - \frac{1}{2|E|}\,(V_r - \overline{V}) \cdot \mathbf{d}_i^\perp, \tag{3.32}$$

from which we can easily obtain the *stability* term (3.24).

3.5. **Reviewing the whole procedure.** We summarize here the calculation of the local VEM stiffness matrix $\mathbf{K}_E^h$.

- Compute the matrices $\mathbf{B}$, $\mathbf{D}$, and $\mathbf{G} = \mathbf{BD}$ given in (3.14), (3.17), and (3.10), respectively.
- Compute the matrices corresponding to the projection operators:

$$\mathbf{\Pi}_*^\nabla = \mathbf{G}^{-1}\mathbf{B}\,, \qquad \mathbf{\Pi}^\nabla = \mathbf{D}\mathbf{\Pi}_*^\nabla.$$

- Finally compute the matrix

$$\mathbf{K}_E^h = (\mathbf{\Pi}_*^\nabla)^{\mathrm{T}}\,\widetilde{\mathbf{G}}\,(\mathbf{\Pi}_*^\nabla) + (\mathbf{I} - \mathbf{\Pi}^\nabla)^{\mathrm{T}}(\mathbf{I} - \mathbf{\Pi}^\nabla),$$

where we recall that $\widetilde{\mathbf{G}}$ is the matrix that coincides with $\mathbf{G}$ except for the first row which is set to zero.

4. Two worked examples

In this section we will present two worked examples in which we compute the matrices **B**, **D** and **G** for a given polygon in the cases $k = 1$ and $k = 2$. We consider the unit square $[0, 1] \times [0, 1]$ and the pentagon E depicted in Fig. 4. In Fig. 5 we show the degrees of freedom for the pentagon when $k = 1$ and $k = 2$. As noted in Remark 3.5, we would get the same results for any polygon obtained by rotation and dilation of our square and our pentagon.

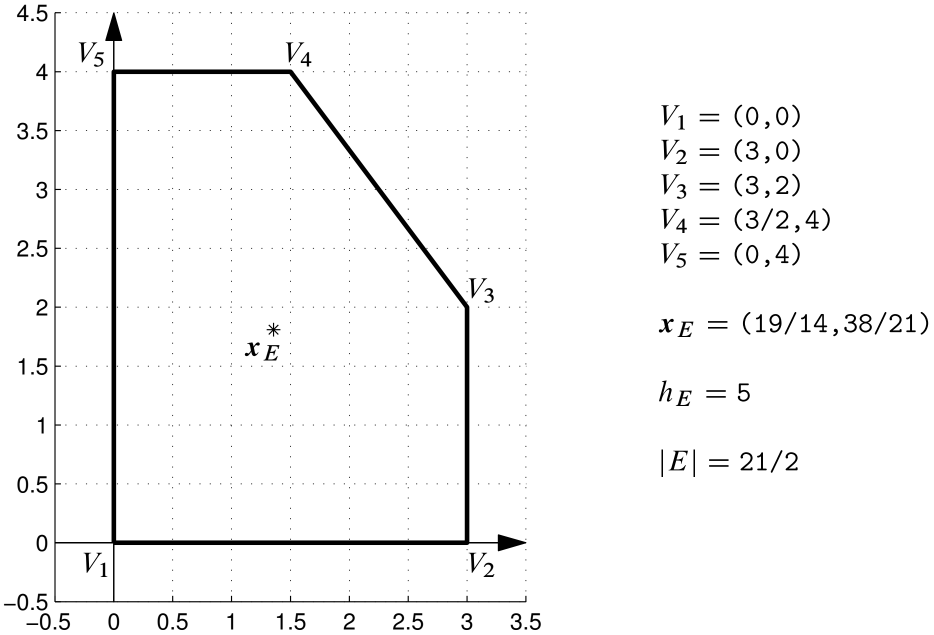

FIGURE 4. The pentagon E

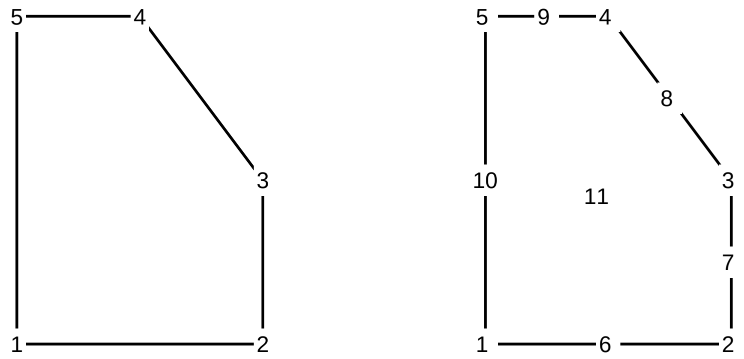

FIGURE 5. Local degrees of freedom for $k = 1$ (left) and $k = 2$ (right)

4.1. **VEM matrices for the unit square.** Set $\mathtt{d}$ equal to the ratio between the diagonal and the edge of the square, i.e. $\mathtt{d}=\sqrt{2}$.

- case $k = 1$

We show for $k = 1$ the matrices **B**, **D**, and **G** = **BD** defined respectively in (3.14), (3.17), and (3.10).

$$\mathtt{B}=\frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ -\mathtt{d} & \mathtt{d} & \mathtt{d} & -\mathtt{d} \\ -\mathtt{d} & -\mathtt{d} & \mathtt{d} & \mathtt{d} \end{bmatrix},\quad \mathtt{D}=\frac{1}{4}\begin{bmatrix} 4 & -\mathtt{d} & -\mathtt{d} \\ 4 & \mathtt{d} & -\mathtt{d} \\ 4 & \mathtt{d} & \mathtt{d} \\ 4 & -\mathtt{d} & \mathtt{d} \end{bmatrix},\quad \mathtt{G}=\frac{1}{2}\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Since on a square harmonic functions are bilinear polynomials, the exact local stiffness matrix $\mathbf{K}_{\square}$ defined in (3.1) (or in (3.20)) coincides with the stiffness matrix obtained from the classical bilinear finite elements. Instead, in agreement with Remark 3.7, the approximate VEM-matrix $\mathbf{K}^h_{\square}$ given in (3.22) is *not* close to $\mathbf{K}_{\square}$, as shown here below:

$$\mathbf{K}_{\square} = \frac{1}{12}\begin{bmatrix} 8 & -2 & -4 & -2 \\ -2 & 8 & -2 & -4 \\ -4 & -2 & 8 & -2 \\ -2 & -4 & -2 & 8 \end{bmatrix},\quad \mathbf{K}^h_{\square} = \frac{1}{12}\begin{bmatrix} 9 & -3 & -3 & -3 \\ -3 & 9 & -3 & -3 \\ -3 & -3 & 9 & -3 \\ -3 & -3 & -3 & 9 \end{bmatrix}.$$

- case $k = 2$

$$\mathtt{B}=\frac{1}{12}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12 \\ -\mathtt{d} & \mathtt{d} & \mathtt{d} & -\mathtt{d} & 0 & 4\mathtt{d} & 0 & -4\mathtt{d} & 0 \\ -\mathtt{d} & -\mathtt{d} & \mathtt{d} & \mathtt{d} & -4\mathtt{d} & 0 & 4\mathtt{d} & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 4 & 0 & 4 & -12 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 4 & 0 & 4 & 0 & -12 \end{bmatrix}$$

$$\mathtt{D}=\frac{1}{24}\begin{bmatrix} 24 & -6\mathtt{d} & -6\mathtt{d} & 3 & 3 & 3 \\ 24 & 6\mathtt{d} & -6\mathtt{d} & 3 & -3 & 3 \\ 24 & 6\mathtt{d} & 6\mathtt{d} & 3 & 3 & 3 \\ 24 & -6\mathtt{d} & 6\mathtt{d} & 3 & -3 & 3 \\ 24 & 0 & -6\mathtt{d} & 0 & 0 & 3 \\ 24 & 6\mathtt{d} & 0 & 3 & 0 & 0 \\ 24 & 0 & 6\mathtt{d} & 0 & 0 & 3 \\ 24 & -6\mathtt{d} & 0 & 3 & 0 & 0 \\ 24 & 0 & 0 & 1 & 0 & 1 \end{bmatrix},\quad \mathtt{G}=\frac{1}{24}\begin{bmatrix} 24 & 0 & 0 & 1 & 0 & 1 \\ 0 & 12 & 0 & 0 & 0 & 0 \\ 0 & 0 & 12 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

4.2. **VEM matrices for the pentagon.**

• case $k = 1$

$$\mathrm{B}=\frac{1}{20}\begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 8 & 4 & 8 & 4 & -8 \\ -6 & -6 & 3 & 6 & 3 \end{bmatrix}, \quad \mathrm{D}=\frac{1}{1470}\begin{bmatrix} 1470 & -399 & -532 \\ 1470 & 483 & -532 \\ 1470 & 483 & 56 \\ 1470 & 42 & 644 \\ 1470 & -399 & 644 \end{bmatrix}$$

$$\mathrm{G}=\frac{1}{1050}\begin{bmatrix} 1050 & 30 & 40 \\ 0 & 441 & 0 \\ 0 & 0 & 441 \end{bmatrix}$$

• case $k = 2$

$$\mathrm{B}=\frac{1}{88200}\times$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 88200 \\ -11760 & 5880 & 11760 & 5880 & -11760 & 0 & 23520 & 23520 & 0 & -47040 & 0 \\ -8820 & -8820 & 4410 & 8820 & 4410 & -35280 & 0 & 17640 & 17640 & 0 & 0 \\ 6384 & 3864 & 7728 & 336 & 6384 & 0 & 15456 & 8400 & 0 & 25536 & -74088 \\ 6650 & -5026 & 1897 & 2828 & -6349 & -1008 & -3808 & 8750 & -2142 & -1792 & 0 \\ 6384 & 6384 & 336 & 7728 & 3864 & 25536 & 0 & 8400 & 15456 & 0 & -74088 \end{bmatrix}$$

$$\mathrm{D}=\frac{1}{176400}\begin{bmatrix} 176400 & -47880 & -63840 & 12996 & 17328 & 23104 \\ 176400 & 57960 & -63840 & 19044 & -20976 & 23104 \\ 176400 & 57960 & 6720 & 19044 & 2208 & 256 \\ 176400 & 5040 & 77280 & 144 & 2208 & 33856 \\ 176400 & -47880 & 77280 & 12996 & -20976 & 33856 \\ 176400 & 5040 & -63840 & 144 & -1824 & 23104 \\ 176400 & 57960 & -28560 & 19044 & -9384 & 4624 \\ 176400 & 31500 & 42000 & 5625 & 7500 & 10000 \\ 176400 & -21420 & 77280 & 2601 & -9384 & 33856 \\ 176400 & -47880 & 6720 & 12996 & -1824 & 256 \\ 176400 & 0 & 0 & 4770 & -1452 & 8480 \end{bmatrix}$$

$$\mathrm{G}=\frac{1}{4410000}\begin{bmatrix} 4410000 & 0 & 0 & 119250 & -36300 & 212000 \\ 0 & 1852200 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1852200 & 0 & 0 & 0 \\ 0 & 0 & 0 & 200340 & -30492 & 0 \\ 0 & 0 & 0 & -30492 & 139125 & -30492 \\ 0 & 0 & 0 & 0 & -30492 & 356160 \end{bmatrix}$$

5. THE L^2-PROJECTOR

Suppose that we are given the degrees of freedom of a function $v_h \in V_k(E)$. In the previous sections we have shown that we can recover the following information about the function v_h:

$$\begin{cases} \bullet \text{ the pointwise value of } v_h \text{ on the edges of the polygon } E; \\ \bullet \text{ the } L^2\text{-projection of } v_h \text{ on the space of polynomials of degree } k-2; \\ \bullet \text{ the projection } \Pi^{\nabla} v_h \text{ on the space of polynomials of degree } k. \end{cases} \tag{5.1}$$

It has been shown in [1, 5, 6] how to compute from (5.1) an approximation $\boldsymbol{b}_E^h$ of the local load term $(\boldsymbol{b}_E)_i := \int_E f\varphi_i$ which ensures that the resulting VEM solution u_h satisfies the optimal error estimates

$$\|u - u_h\|_{0,\Omega} = O(h^{k+1}), \quad \|\nabla u - \nabla u_h\|_{0,\Omega} = O(h^k).$$

However, the information in (5.1) is not sufficient in slightly more complicated situations, like for instance the presence of a time-dependent term or a zero-order term in the equation. In fact, in this case we would need to compute a good approximation of the local mass matrix

$$(\mathbf{M}_E)_{ij} := \int_E \varphi_i \, \varphi_j \, ,$$

and the information in (5.1) is not enough, both for accuracy and stability reasons.

In this Section we present an enhanced version of the Virtual Element Method that will allow us to compute, using only the degrees of freedom, the full L^2 projection on $\mathcal{P}_k(E)$ of any function $v_h \in V_k(E)$. This possibility greatly increases the number of situations in which the VEM can be used, including a simple treatment of the load term, the presence of other terms in the differential equation, and the extension to three-dimensional problems.

5.1. **Definition and properties of the L^2-projector.** The idea here is to compute the L^2-projection $\Pi^0_{E,k} v_h$ onto the space of polynomials $\mathcal{P}_k(E)$ of every function $v_h \in V_k(E)$ using (again) only the degrees of freedom of v_h. Also in this case we will simply write Π^0, Π^0_E or Π^0_k when no confusion can arise.

To this aim, we recall that for every function $v_h \in V_k(E)$ (and, actually, for every function in $L^2(E)$) the polynomial $\Pi^0 v_h$ is defined by the orthogonality conditions

$$\left(p_k, \Pi^0 v_h - v_h\right)_{0,E} = 0, \quad p_k \in \mathcal{P}_k(E). \tag{5.2}$$

Proceeding as before, we define the matrix

$$\mathbf{H}_{\alpha\beta} := \left(m_\alpha, m_\beta\right)_{0,E} \quad \alpha, \beta = 1, \ldots, n_k \tag{5.3}$$

and denote by $t^\alpha = t^\alpha(v_h)$ the coefficients of $\Pi^0 v_h$ in the basis m_α:

$$\Pi^0 v_h = \sum_{\alpha=1}^{n_k} t^\alpha m_\alpha. \tag{5.4}$$

Then, mimicking what we did for the operator Π^∇, we define the n_k-vectors $\underline{t}$ and $\underline{c}$ with components, respectively, t^α and

$$c^\alpha := \left(m_\alpha, v_h\right)_{0,E}. \tag{5.5}$$

Using this notation (namely (5.4), (5.3), and (5.5)) we can re-write (5.2) as

$$\mathbf{H}\underline{t} = \underline{c} \qquad \text{that is} \qquad \underline{t} = \mathbf{H}^{-1}\underline{c}, \tag{5.6}$$

and we have *just* to compute $\underline{c}$ in (5.5) using only the degrees of freedom of v_h. This looks clearly impossible. In particular, in (2.1), for $k = 1$ we have no internal degrees of freedom at all for v_h, for $k = 2$ we have only $\left(m_1, v_h\right)_{0,E} \equiv \left(1, v_h\right)_{0,E}$, and for a general k we have the moments (to be used in (5.5)) only for the monomials m_α in $\mathcal{P}_{k-2}(E)$. No way...

The escape from this (apparent) *cul-de-sac* is given, once more, by the operator Π^∇. Indeed, we recall first that we have been able to compute $\Pi^\nabla v_h$, for every $v_h \in V_k(E)$, using only the degrees of freedom of v_h. Then we point out that, for every $v_h \in V_k(E)$, both $\Pi^\nabla v_h$ and $\Pi^0 v_h$ are good approximations of v_h, and coincide with v_h whenever v_h is a polynomial in $\mathcal{P}_k(E)$. Hence, they must be close to each other. So the idea could be to replace (5.5), *for the monomials m_α of degree k and $k-1$*, by:

$$c^\alpha := \left(m_\alpha, \Pi^\nabla v_h\right)_{0,E} \tag{5.7}$$

which is computable from the degrees of freedom of v_h. In so doing, we are apparently introducing some small mistakes in the computation. Instead, we will show now that this is not the case. We do that by imagining new spaces $W_k(E)$ that share all the previous good properties of $V_k(E)$, i.e.,

- the elements of $W_k(E)$ are polynomials of degree k on each edge of E;
- $W_k(E)$ contains $\mathcal{P}_k(E)$;
- in $W_k(E)$ we can use the same degrees of freedom used in $V_k(E)$,

plus the additional (spectacular) property that

$$\int_E w_h\, m_{\boldsymbol{\alpha}} = \int_E \Pi^\nabla w_h\, m_{\boldsymbol{\alpha}}, \quad |\boldsymbol{\alpha}| = k-1, k, \quad w_h \in W_k(E). \tag{5.8}$$

If these spaces existed, then all the previous computations would hold unchanged, and for them (5.7) would *coincide* with (5.5). Indeed, the proof of existence of these spaces and the analysis of their properties has been presented in [1].

It is easy to check that whenever two functions $v_h \in V_k(E)$ and $w_h \in W_k(E)$ share the same degrees of freedom (2.1), they also share the same Π^∇-projection, namely $\Pi^\nabla v_h = \Pi^\nabla w_h$. See also Remark 3.2.

* **Remark 5.1.** *The precise definition of the space $W_k(E)$ given in [1] consists in substituting property iii) in the definition of $V_k(E)$ with the following two:*

iii)' Δw_h *is a polynomial of degree* k *in* E*;*

iii)" $\displaystyle\int_E w_h\, m_{\boldsymbol{\alpha}} = \int_E \Pi^\nabla w_h\, m_{\boldsymbol{\alpha}}, \quad |\boldsymbol{\alpha}| = k-1, k.$

From the practical point of view, using the degrees of freedom (2.1) in the computer, one never knows whether they refer to a function $v_h \in V_k(E)$ or to a function $w_h \in W_k(E)$. Hence, what has to be done, *in practice*, for a given set of degrees of freedom $\mathrm{dof}(v_h) = \mathrm{dof}(w_h)$ is the following:

- Compute the matrix $\mathbf{H}$ from (5.3). Remember that we consider that we can always compute the integral of a polynomial over E.
- Compute $\Pi^\nabla v_h \equiv \Pi^\nabla w_h$ as in Section 3.1. Remember that the projection depends only on the degrees of freedom.
- Compute the right-hand side $\underline{c}$ using (5.7) when m_α has a degree bigger than $k-2$, or (5.5) otherwise.
- Compute $\underline{t}$ solving the system (5.6) with $\mathbf{H}$ given by (5.3).
- Recalling (2.3) and (3.17) compute

$$\Pi^0 w_h = \sum_{\alpha=1}^{n_k} t^\alpha m_\alpha = \sum_{\alpha=1}^{n_k} t^\alpha \sum_{i=1}^{N^{\mathrm{dof}}} \mathbf{D}_{\alpha i} \varphi_i$$

according to whether you need to represent $\Pi^0 w_h$ in the monomial basis m_α (of $\mathcal{P}_k(E)$) or in the VEM basis φ_i of $W_k(E)$.

Remark 5.2. *As the final result depends only on the degrees of freedom, we might also think that the above function is, somehow, "the $L^2(E)$ projection of the degrees of freedom".*

5.2. The cases $k = 1$ and $k = 2$.

If $k = 1$ or $k = 2$, we can easily see that $\Pi^\nabla = \Pi^0$.

• $k = 1$:

$\Pi^0 v_h$ is defined by equation (5.2):

$$\left(p_1, \Pi^0 v_h\right)_{0,E} = \left(p_1, v_h\right)_{0,E}, \quad p_1 \in \mathcal{P}_1(E),$$

and condition (5.8) in this case is equivalent to

$$\left(p_1, \Pi^\nabla v_h\right)_{0,E} = \left(p_1, v_h\right)_{0,E}, \quad p_1 \in \mathcal{P}_1(E).$$

Hence $\Pi^\nabla = \Pi^0$.

• $k = 2$:

Condition (3.3) and definition (3.4b) imply

$$\left(1, \Pi^\nabla v_h\right)_{0,E} = \left(1, v_h\right)_{0,E}$$

which, together with (5.8), shows that

$$\left(p_2, \Pi^\nabla v_h\right)_{0,E} = \left(p_2, v_h\right)_{0,E}, \quad p_2 \in \mathcal{P}_2(E).$$

Hence, by (5.2) we conclude that $\Pi^\nabla = \Pi^0$.

Remark 5.3. *At this point it is clear why we made the choice (3.4b) instead of (3.4a) for $k \geq 2$.*

5.3. **Construction of the L^2-projection of the basis functions.** In the code, you will need to compute the $L^2(E)$-projection of each basis function φ_i in $W_k(E)$, and the corresponding matrices $\mathbf{\Pi}^0_*$ and $\mathbf{\Pi}^0$ that give the projection in terms of the monomial basis or in terms of the VEM basis, respectively.

• The former is

$$\left(\mathbf{\Pi}^0_*\right)_{\alpha i} := t^\alpha(\varphi_i) = \sum_{\beta=1}^{n_k} \left(\mathbf{H}^{-1}\right)_{\alpha\beta} c_i^\beta = \left(\mathbf{H}^{-1}\mathbf{C}\right)_{\alpha i} \tag{5.9}$$

where $\mathbf{H}$ is given in (5.3), and $\mathbf{C}$ is the $n_k \times N^{\text{dof}}$ matrix

$$\mathbf{C}_{\alpha i} := \left(m_\alpha, \varphi_i\right)_{0,E} = \begin{cases} \left(m_\alpha, \varphi_i\right)_{0,E} & \text{if} \qquad 1 \leq \alpha \leq n_{k-2} \\ \left(m_\alpha, \Pi^\nabla \varphi_i\right)_{0,E} & \text{if} \quad n_{k-2}+1 \leq \alpha \leq n_k \end{cases} \tag{5.10}$$

for $i = 1, \ldots, N^{\text{dof}}$ and $\alpha = 1, \ldots, n_k$. The definition of $\mathbf{C}$ in (5.10) can also be expressed by saying that the coefficients $\left(m_\alpha, \varphi_i\right)_{0,E}$ have been written using (5.7) when m_α has a degree bigger than $k-2$, and (5.5) otherwise. Note that this means that

$$\mathbf{C}_{\alpha i} = (\mathbf{H}\mathbf{G}^{-1}\mathbf{B})_{\alpha i} \text{ when } m_\alpha \text{ has degree } k-1 \text{ and } k.$$

• The latter instead is

$$\left(\mathbf{\Pi}^0\right)_{ji} := \sum_{\alpha=1}^{n_k} \mathbf{D}_{j\alpha}\left(\mathbf{\Pi}^0_*\right)_{\alpha i} = \left(\mathbf{D}\mathbf{H}^{-1}\mathbf{C}\right)_{ji} \tag{5.11}$$

where $\mathbf{D}$ is always defined in (3.17). Note that, in particular, from (5.4) and (5.9) we also have

$$\Pi^0 \varphi_i = \sum_{\alpha=1}^{n_k} t^\alpha(\varphi_i) m_\alpha = \sum_{\alpha=1}^{n_k} \left(\mathbf{\Pi}^0_*\right)_{\alpha i} m_\alpha. \tag{5.12}$$

Remark 5.4. *The structure of the matrix* $\mathbf{C}$ *can be easily described as follows:*

- *if α is between 1 and n_{k-2}, then $\mathbf{C}_{\alpha i} = 0$ except for $i = kN^V + \alpha$ (corresponding to the α-th internal degree of freedom) for which $\mathbf{C}_{\alpha i} = |E|$;*
- *if α is between $n_{k-2}+1$ and n_k (corresponding to scaled monomials of degree $k-1$ and k) then*

$$\mathbf{C}_{\alpha i} = (\mathbf{H}\mathbf{\Pi}^\nabla_*)_{\alpha i} = (\mathbf{H}\mathbf{G}^{-1}\mathbf{B})_{\alpha i}.$$

6. The load term and the mass matrix

With this new weapon, we are ready to face the approximation of the load term and of the mass matrix.

6.1. **Approximation of the load term $\boldsymbol{b}_E$.** We simply define:

$$\left(\boldsymbol{b}_E^h\right)_i := \int_E f \ \Pi^0_k \varphi_i. \tag{6.1}$$

With this choice, in [1] we have proven that we have the optimal error estimate in H^1 and L^2.

* **Remark 6.1.** *In [1] we have shown that we can still achieve the optimal error estimate even if we do not employ the full L^2-projection onto $\mathcal{P}_k(E)$. In particular, we have shown that it is enough to take the following approximation for the right-hand side:*

$$\begin{aligned} \left(\boldsymbol{b}_E^h\right)_i &:= \int_E f \ \Pi^0_{k-1}\varphi_i \quad \textit{for } k = 1,2 \\ \left(\boldsymbol{b}_E^h\right)_i &:= \int_E f \ \Pi^0_{k-2}\varphi_i \quad \textit{for } k \geq 3. \end{aligned} \tag{6.2}$$

We remark that using (6.2) *instead of* (6.1) *produces a larger approximation error, notwithstanding the convergence curves retain the same (optimal) slope. We finally observe that a different approximation of the load term was proposed initially in [5]; such an approximation coincides with that given in* (6.2) *for $k \geq 3$ but not for $k = 1, 2$. The present choice is better as it guarantees an optimal rate of convergence* *in the L^2 norm also for $k = 2$. A different, but in the end equivalent, presentation of* (6.1) *for $k = 2$ was proposed in [6].*

6.2. **Approximation of the local mass matrix $\mathbf{M}_E$.** In this section we show how to compute the VEM approximation $\mathbf{M}_E^h$ of the local mass matrix

$$(\mathbf{M}_E)_{ij} = \int_E \varphi_i\, \varphi_j .$$

Mimicking what we did in Subsection 3.3 for the local stiffness matrix, we use the identity

$$\varphi_i = \Pi^0 \varphi_i + (\mathrm{I} - \Pi^0)\varphi_i$$

to obtain

$$(\mathbf{M}_E)_{ij} = \int_E \varphi_i\, \varphi_j = \int_E \Pi^0\varphi_i\, \Pi^0\varphi_j + \int_E (\mathrm{I} - \Pi^0)\varphi_i\, (\mathrm{I} - \Pi^0)\varphi_j + \int_E \Pi^0\varphi_i\, (\mathrm{I} - \Pi^0)\varphi_j + \int_E (\mathrm{I} - \Pi^0)\varphi_i\, \Pi^0\varphi_j. \quad (6.3)$$

Since, from (5.2), we obviously have

$$\int_E (\mathrm{I} - \Pi^0)u\, \Pi^0 v = 0 \qquad u, v \in L^2(E),$$

the last two terms in (6.3) are zero and we end up with the identity

$$(\mathbf{M}_E)_{ij} = \int_E \Pi^0\varphi_i\, \Pi^0\varphi_j + \int_E (\mathrm{I} - \Pi^0)\varphi_i\, (\mathrm{I} - \Pi^0)\varphi_j .$$

As before, the first term ensures consistency and must be computed exactly, while the second one guarantees stability and, as shown in [1], can be approximated by

$$\int_E (\mathrm{I} - \Pi^0)\varphi_i\, (\mathrm{I} - \Pi^0)\varphi_j \approx |E| \sum_{r=1}^{N^{\mathrm{dof}}} \mathrm{dof}_r\left(\left(\mathrm{I} - \Pi^0\right)\varphi_i\right)\, \mathrm{dof}_r\left(\left(\mathrm{I} - \Pi^0\right)\varphi_j\right),$$

where $|E|$ is, as usual, the area of the polygon E. Then, using (5.12), we have

$$\int_E \Pi^0\varphi_i\, \Pi^0\varphi_j = \sum_{\alpha=1}^{n_k}\sum_{\beta=1}^{n_k} (\mathbf{\Pi}^0_*)_{\alpha i} (\mathbf{\Pi}^0_*)_{\beta j} \int_E m_\alpha\, m_\beta = ((\mathbf{\Pi}^0_*)^{\mathrm{T}} \mathbf{H} \mathbf{\Pi}^0_*)_{ij} = (\mathbf{C}^{\mathrm{T}}\mathbf{H}^{-1}\mathbf{C})_{ij}$$

and

$$\int_E (\mathrm{I} - \Pi^0)\varphi_i\, (\mathrm{I} - \Pi^0)\varphi_j \approx |E|\, [(\mathbf{I} - \mathbf{\Pi}^0)^{\mathrm{T}}(\mathbf{I} - \mathbf{\Pi}^0)]_{ij}. \quad (6.4)$$

The final formula for the local VEM mass matrix is

$$\boxed{\mathbf{M}_E^h = \mathbf{C}^{\mathrm{T}}\mathbf{H}^{-1}\mathbf{C} + |E|(\mathbf{I} - \mathbf{\Pi}^0)^{\mathrm{T}}(\mathbf{I} - \mathbf{\Pi}^0)}, \quad (6.5)$$

where the matrices $\mathbf{C}, \mathbf{H}$ and $\mathbf{\Pi}^0$ were introduced in equations (5.10), (5.3) and (5.11), respectively.

* **Remark 6.2.** *Note that, as already discussed in Remark 3.4, the integral of $\varphi_i \varphi_j$ on an element E scales like the area of E. This justifies the choice* (6.4) *for the stabilizing term.*

* **Remark 6.3.** *We point out that the stabilization term* (6.4) *for the local mass matrix is only needed when the equation is reaction-dominated.*

* **Remark 6.4.** *Also in this case we can multiply the stabilization term* (6.4) *by a factor which stays bounded with h. See Remark 3.6.*

7. VEM IN THREE DIMENSIONS

Here we will summarize the basic features of the Virtual Element Method in three dimensions. Before doing that, however, we want to stress an important feature of the construction of the L^2-projection operator of Subsection 5.1. Indeed, we saw that: assuming that our shape functions, inside each polygon E, are in the space $W_k(E)$ (and not $V_k(E)$), we can start from the local degrees of freedom (2.1) (the same ones that we would have in $V_k(E)$), and construct, for each shape function $w_h \in W_k(E)$, its *exact* $L^2(E)$ projection $\Pi^0 w_h$ on the space of polynomials $\mathcal{P}_k(E)$. Hence, using $W_k(E)$ we can say that for each $k \geq 1$ the degrees of freedom (2.1) of a given shape function w_h allow to compute in an *exact* way all the moments up to order k:

$$\int_E w_h\, p_k, \qquad p_k \in \mathcal{P}_k(E). \tag{7.1}$$

In our three-dimensional method this observation will be applied using spaces $W_k(f)$ on every *face* f of a given polyhedron.

7.1. **The model 3D problem and the decomposition.** We go back to the model problem (1.1) where this time Ω is (for simplicity) a convex polyhedron. We suppose that we are given a decomposition $\mathcal{P}_h$ of Ω in rather general polyhedra P. Almost every polyhedron will be acceptable, apart from the ones that come from the (usually shining) mind of a mathematician willing to construct counterexamples. But to fix the ideas, following [1], we can assume that there exists a constant $C_{\mathcal{P}_h} > 0$ such that: each polyhedron $P \in \mathcal{P}_h$ is starshaped with respect to every point of a ball of radius $\geq C_{\mathcal{P}_h} h_P$; for every face $f \in \partial P$ we have $h_f \geq C_{\mathcal{P}_h} h_P$ and f is starshaped with respect to every point of a disk of radius $\geq C_{\mathcal{P}_h} h_f$; for every edge $e \in \partial f$, we have $|e| \geq C_{\mathcal{P}_h} h_P$.

7.2. **The local and global degrees of freedom.** For every integer $k > 0$, mimicking the 2D case we consider the space $V_k(P)$ of functions v_h such that

$$\begin{cases} \bullet\ v_h \text{ is a polynomial of degree } k \text{ on each edge } e \text{ of } P, \text{ i.e. } v_{h|e} \in \mathcal{P}_k(e); \\ \bullet\ \text{for every face } f \text{ in } \partial P \text{ the restriction } v_{h|f} \text{ belongs to } W_k(f); \\ \bullet\ v_{h|\partial P} \in C^0(\partial P); \\ \bullet\ \Delta v_h \text{ is a polynomial of degree } k-2 \text{ in } P. \end{cases}$$

Remark 7.1. *Note that even here a polynomial of degree k satisfies the conditions above so that $\mathcal{P}_k(P)$ is a subspace of $V_k(P)$.*

In [1] it is shown that we can take the following degrees of freedom in $V_k(P)$:

$$\begin{cases} \textit{i)}\ \text{the value of } v_h \text{ at the vertices of } P; \\ \textit{ii)}\ \text{on each edge } e, \text{ the value of } v_h \text{ at the } k-1 \text{ internal points of the} \\ \quad (k+1)\text{-points Gauss-Lobatto quadrature rule on } e; \\ \textit{iii)}\ \text{for each face } f \text{ the moments up to order } k-2 \text{ of } v_h \text{ in } f: \\ \qquad \displaystyle\int_f v_h\, m_\alpha, \quad \alpha = 1, \dots, n_{k-2} \\ \quad \text{where the scaled monomials } m_\alpha \in \mathcal{M}_{k-2}(f) \text{ are defined in} \\ \quad (1.3); \\ \textit{iv)}\ \text{the moments up to order } k-2 \text{ of } v_h \text{ in } P: \\ \qquad \displaystyle\int_P v_h\, \mu_\alpha, \quad \alpha = 1, \dots, \nu_{k-2} \\ \quad \text{where the scaled monomials } \mu_\alpha \in \mathcal{M}_{k-2}(P) \text{ are defined in} \\ \quad (1.5), \text{ and } \nu_{k-2} \text{ in (1.2).} \end{cases} \tag{7.2}$$

As a consequence, the dimension of $V_k(P)$ is given by

$$\dim V_k(P) = N_P^V + N_P^e\,(k-1) + N_P^f\, n_{k-2} + \nu_{k-2}$$

where N_P^V, N_P^e, and N_P^f represent, obviously, the number of vertices, edges, and faces of P, respectively.

We will refer to the first three sets of degrees of freedom as *boundary degrees of freedom* and to the fourth set as *internal degrees of freedom.*

Remark 7.2. *As for polygons, there are no internal degrees of freedom in the case $k = 1$. However in three dimensions we do not recover exactly the harmonic barycentric coordinates (see [9]), as our functions are* **not** *harmonic, in general, on the faces (see iii)' in Remark 5.1). This however will allow us to satisfy, also in 3D, the patch-test exactly.*

As we did before, we number the degrees of freedom from 1 to $N_P^{\mathrm{dof}} := \dim V_k(P)$ and we define the operator dof_i from $V_k(P)$ to $\mathbb{R}$ as

$$\mathrm{dof}_i(v_h) := i\text{-th degree of freedom of } v_h, \quad i = 1, \dots, N_P^{\mathrm{dof}}.$$

The *basis functions* $\varphi_i \in V_k(P)$ are defined as usual as the canonical basis functions:

$$\mathrm{dof}_i(\varphi_j) = \delta_{ij}, \quad i, j = 1, \dots, N_P^{\mathrm{dof}},$$

so that we have a Lagrange-type interpolation identity:

$$v_h = \sum_{i=1}^{N_P^{\mathrm{dof}}} \mathrm{dof}_i(v_h)\, \varphi_i \quad \text{for all } v_h \in V_k(P).$$

Now we can define the global finite element space V_h as

$$V_h := \{v_h \in H_0^1(\Omega) \quad : \quad v_{h|E} \in V_k(P) \quad \text{for all } P \in \mathcal{P}_h\}$$

with the following global degrees of freedom for v_h:

$$\begin{cases} \textit{i)} \ \text{the value of } v_h \text{ at the internal vertices of the decomposition;} \\ \textit{ii)} \ \text{on each internal edge } e, \text{ the value of } v_h \text{ at the } k-1 \text{ internal points} \\ \quad \text{of the } (k+1)\text{-points Gauss-Lobatto quadrature rule on } e; \\ \textit{iii)} \ \text{for each internal face } f, \text{ the moments up to order } k-2 \text{ of } v_h \text{ in } f: \\ \qquad \displaystyle\int_f v_h\, m_\alpha, \quad \alpha = 1, \dots, n_{k-2}; \\ \textit{iv)} \ \text{for each polygon } P, \text{ the moments up to order } k-2 \text{ of } v_h \text{ in } P: \\ \qquad \displaystyle\int_P v_h\, \mu_\alpha, \quad \alpha = 1, \dots, \nu_{k-2}. \end{cases} \tag{7.3}$$

7.3. **Guidelines for constructing the stiffness matrix.** First of all, we point out that, on each internal face f of the decomposition $\mathcal{P}_h$, the degrees of freedom *i) – iii)* in (7.3) identify, in a unique way, a function $v_{h|f} \in W_k(f)$. Then one can, proceeding as in Section 5.1, construct the projection operator Π_f^∇ applied to v_h (and then, if necessary, its $L^2(f)$-projection).

Now, for each polyhedron $P \in \mathcal{P}_h$, and for each v_h in $V_k(P)$ we can proceed as in Subsection 3.1 and construct $\Pi_P^\nabla v_h$ defined by

$$\begin{cases} \left(\nabla p_k, \nabla\left(\Pi_P^\nabla v_h - v_h\right)\right)_{0,P} = 0 \quad \text{for all } p_k \in \mathcal{P}_k(P), \\ \mathrm{P}_0\big(\Pi_P^\nabla v_h - v_h\big) = 0, \end{cases}$$

where, here too, P_0 is a projection operator onto constant functions, that we choose (following the 2D case) as

$$\mathrm{P}_0 v_h := \frac{1}{N_P^V} \sum_{i=1}^{N_P^V} v_h(V_i) \quad \text{for } k = 1 \tag{7.4a}$$

$$\mathrm{P}_0 v_h := \frac{1}{|P|} \int_P v_h \quad \text{for } k \geq 2. \tag{7.4b}$$

Note that, integrating by parts as in (3.12), we have again, for every $p_k \in \mathcal{P}_k(P)$,

$$(\nabla p_k, \nabla v_h)_{0,P} = -\int_P \Delta p_k\, v_h + \int_{\partial P} \frac{\partial p_k}{\partial n}\, v_h. \tag{7.5}$$

Since $\Delta p_k \in \mathcal{P}_{k-2}(P)$, the first term can again (for $k > 1$) be computed using the degrees of freedom *iv)* of (7.2). The second term, instead, cannot be computed directly from the degrees of freedom *iii)* of

(7.2), since on each face f of ∂P, $\dfrac{\partial p_k}{\partial n}$ is in $\mathcal{P}_{k-1}(f)$, but the choice of using $v_{h|f} \in W_k(f)$ allows us to compute the moments of order $k-1$ and k as well, as in (7.1).

Once we know how to compute $\Pi^\nabla_P v_h$ for $v_h \in V_k(P)$, we can follow step by step the path of the two-dimensional case.

7.4. **Reviewing the whole procedure.**

We summarize here the main steps of the whole procedure.

For each face f:

- we compute the matrices: $\mathbf{G}_f$ as in (3.10), $\mathbf{B}_f$ as in (3.14), and $\mathbf{D}_f$ as in (3.17). Note that $\mathbf{G}_f$ can be obtained as $\mathbf{B}_f\mathbf{D}_f$ as done in (3.19). However an independent computation from (3.10) could be a very valuable check of the code;
- from the matrices $\mathbf{G}_f$, $\mathbf{B}_f$, and $\mathbf{D}_f$, we can compute the matrix $\mathbf{\Pi}^\nabla_f$ as in (3.18);
- given $\mathbf{\Pi}^\nabla_f$, we know the moments up to order $k-1$ of each basis function φ_i: using the original degrees of freedom *iii)* of (7.2) for the moments of order up to $k-2$, and those of $\Pi^\nabla_f \varphi_i$ for the moments of order $k-1$.

With this information we can use (7.5) to compute $\left(\nabla\varphi_i, \nabla\mu_\alpha\right)_{0,P}$ for all basis functions $\varphi_i \in V_k(P)$ and for all $\mu_\alpha \in \mathcal{M}_k(P)$. Next, following the 2D track:

- We compute the matrices
$$\mathbf{G}_P := \begin{bmatrix} \mathrm{P}_0\mu_1 & \mathrm{P}_0\mu_2 & \dots & \mathrm{P}_0\mu_{\nu_k} \\ 0 & \left(\nabla\mu_2, \nabla\mu_2\right)_{0,P} & \dots & \left(\nabla\mu_2, \nabla\mu_{\nu_k}\right)_{0,P} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \left(\nabla\mu_{\nu_k} v, \nabla\mu_2\right)_{0,P} & \dots & \left(\nabla\mu_{\nu_k}, \nabla\mu_{\nu_k}\right)_{0,P} \end{bmatrix}$$
$$\mathbf{B}_P = \begin{bmatrix} \mathrm{P}_0\varphi_1 & \dots & \mathrm{P}_0\varphi_{N_P^{\mathrm{dof}}} \\ \left(\nabla\mu_2, \nabla\varphi_1\right)_{0,P} & \dots & \left(\nabla\mu_2, \nabla\varphi_{N_P^{\mathrm{dof}}}\right)_{0,P} \\ \vdots & \ddots & \vdots \\ \left(\nabla\mu_{n_k}, \nabla\varphi_1\right)_{0,P} & \dots & \left(\nabla\mu_{n_k}, \nabla\varphi_{N_P^{\mathrm{dof}}}\right)_{0,P} \end{bmatrix},$$
with P_0 defined in (7.4) and N_P^{dof} being the number of degrees of freedom on the polyhedron P. Here too the matrix representation $\mathbf{\Pi}^\nabla_{*,P}$ of the operator Π^∇_P acting from $V_k(P)$ to $\mathcal{P}_k(P)$ in the basis $\mathcal{M}_k(P)$ is given by
$$\mathbf{\Pi}^\nabla_{*,P} := \mathbf{G}_P^{-1}\mathbf{B}_P.$$
- The same operator in the φ_i basis has the matrix representation
$$\mathbf{\Pi}^\nabla_P := \mathbf{D}_P\mathbf{G}_P{}^{-1}\mathbf{B}_P$$
where again
$$\mathbf{D}_P = \begin{bmatrix} \mathrm{dof}_1(\mu_1) & \mathrm{dof}_1(\mu_2) & \dots & \mathrm{dof}_1(\mu_{\nu_k}) \\ \mathrm{dof}_2(\mu_1) & \mathrm{dof}_2(\mu_2) & \dots & \mathrm{dof}_2(\mu_{\nu_k}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{dof}_{N_P^{\mathrm{dof}}}(\mu_1) & \mathrm{dof}_{N_P^{\mathrm{dof}}}(\mu_2) & \dots & \mathrm{dof}_{N_P^{\mathrm{dof}}}(\mu_{\nu_k}) \end{bmatrix}.$$
Obviously, also in this case we have the useful relationship (3.19), i.e. $\mathbf{G}_P = \mathbf{B}_P\mathbf{D}_P$.
- Finally, we compute the VEM local stiffness matrix as
$$\boxed{\mathbf{K}^h_P = (\mathbf{\Pi}^\nabla_{*,P})^{\mathrm{T}}\,\widetilde{\mathbf{G}}_P\,(\mathbf{\Pi}^\nabla_{*,P}) + h_P(\mathbf{I}-\mathbf{\Pi}^\nabla_P)^{\mathrm{T}}(\mathbf{I}-\mathbf{\Pi}^\nabla_P)} \tag{7.6}$$
where we recall that h_P represents the diameter of P, and $\widetilde{\mathbf{G}}_P$ is the matrix that coincides with $\mathbf{G}_P$ except for the first row which is set to zero.

Remark 7.3. *We point out that also in this case, following Section 5, we can define the "enhanced" space $W_k(P)$ and compute the L^2-projection onto $\mathcal{P}_k(P)$ to appropriately treat the load term and the zero order term.*

8. A HINT ON MORE GENERAL CASES

In the present section we give a brief hint on more general types of Virtual Element Methods and to the basic structure of their implementation. We consider a general symmetric and V-elliptic bilinear form

$$\mathcal{A}(\cdot,\cdot) : V \times V \longrightarrow \mathbb{R}$$

where V is a Hilbert space of (scalar or vector) functions defined on a domain Ω. For every given right-hand side $f \in V'$ we can consider the problem of finding $u \in V$ such that

$$\mathcal{A}(u,v) =< f, v > \qquad \forall\, v \in V,$$

which clearly has a unique solution. Given $\mathcal{A}^E$, the restriction of the bilinear form $\mathcal{A}$ to a generic element E, we need to build its VEM approximation

$$\mathcal{A}_h^E(\cdot,\cdot) : V_k(E) \times V_k(E) \longrightarrow \mathbb{R}$$

following the strategy shown above. Note that the space $V_k(E)$ in this section is more general with respect to that in Section 2 and may represent also vector valued functions. Although the global bilinear form $\mathcal{A}$ (defined on the whole space V) is supposed to be positive definite, its local counterpart $\mathcal{A}^E$ could easily be only positive semidefinite. This happens, for instance, in the case of the Laplace operator, both for VEM and for classical FEM. We therefore introduce $\ker(\mathcal{A}^E)$ as the kernel of the local bilinear form $\mathcal{A}^E$

$$\ker(\mathcal{A}^E) = \left\{ v \in V_{|E} \ : \ \mathcal{A}^E(v,v) = 0 \right\}.$$

We assume that we have a polynomial space $\mathcal{P}_k(E)$ that can provide, in E, a sufficiently good approximation of the solution u of our problem. We make the following two assumptions (with the obvious modifications for the vector valued case):

(C1) It holds

$$\ker(\mathcal{A}^E) \subseteq \mathcal{P}_k(E) \subseteq V_k(E) \subseteq V_{|E};$$

(C2) for all $v_h \in V_k(E)$ and $p_k \in \mathcal{P}_k(E)$ the term $\mathcal{A}^E(v_h, p_k)$ is explicitly computable on the basis of the available degrees of freedom.

We introduce a (scaled) polynomial basis

$$\mathcal{P}_k(E) = \text{span}\{\widetilde{m}_1, \widetilde{m}_2, ..., \widetilde{m}_{n_k}\}$$

is such a way that the first $n_{\ker}$ functions span the kernel of $\mathcal{A}^E$

$$\ker(\mathcal{A}^E) = \text{span}\{\widetilde{m}_1, \widetilde{m}_2, ..., \widetilde{m}_{n_{\ker}}\}.$$

Moreover let P_α, for $\alpha = 1, 2, .., n_{\ker}$, be some projection from $V_k(E)$ on the one dimensional space spanned by $\widetilde{m}_\alpha$ that satisfies the following assumption:

(C3) for any $\alpha = 1, 2, .., n_{\ker}$, the operator P_α acting on $V_k(E)$ is computable on the basis of the available degrees of freedom for $V_k(E)$.

Example 1 In the case of the Laplace operator (1.1) we have $V = H^1(\Omega)$ (possibly with boundary conditions) and $\mathcal{A}(v,w) = (\nabla v, \nabla w)_{0,\Omega}$. The local discrete space $V_k(E)$ is the one introduced in Section 2. Since in this case $\ker(\mathcal{A}) = \mathcal{P}_0(E)$, condition (C1) is immediately verified, while condition (C2) follows as explained in (3.12). The scaled polynomial basis is given by the monomials $\{m_1, m_2, ..., m_{n_k}\}$ introduced in (1.3), the dimension $n_{\ker}$ of the kernel is 1 and $\mathsf{P}_1 = \mathrm{P}_0$ is defined in (3.4).

Example 2 In the case of plate bending problems, say, $\Delta^2 u = f$ in Ω with $u = \partial u/\partial n = 0$ on $\partial\Omega$, the space V will be taken as $H_0^2(\Omega)$ and the bilinear form (for simplicity) $\mathcal{A}(v,w) := (\boldsymbol{\varepsilon}(\nabla v), \boldsymbol{\varepsilon}(\nabla w))_{0,\Omega}$. The simplest (lowest order) local discrete space $V_2(E)$ will be made of bi-harmonic functions that are polynomials of degree 3 on each edge, with normal derivative linear on each edge (see [17]) and the degrees of freedom will be the values of v_h and of its two (scaled) derivatives at each vertex. Clearly this space contains all polynomials in $\mathcal{P}_2(E)$. Since in this case $\ker(\mathcal{A}) = \mathcal{P}_1(E)$, condition (C1) is immediately verified, while condition (C2) follows, similarly to (3.12), integrating by parts. The scaled polynomial basis is given by the monomials $\{m_1, m_2, ..., m_{n_k}\}$ introduced in (1.3), the dimension n_{ker} of the kernel is 3, and the projection operators can be taken as

$$\mathsf{P}_1 v_h := \frac{1}{N^V}\sum_{i=1}^{N^V} v_h(V_i),$$

$$\mathsf{P}_2 v_h := \frac{1}{N^V}\sum_{i=1}^{N^V} \frac{\partial v_h}{\partial x_1}(V_i)\frac{(\boldsymbol{x}-\boldsymbol{x}_{\mathcal{P}})_1}{h_{\mathcal{P}}}, \quad \mathsf{P}_3 v_h := \frac{1}{N^V}\sum_{i=1}^{N^V} \frac{\partial v_h}{\partial x_2}(V_i)\frac{(\boldsymbol{x}-\boldsymbol{x}_{\mathcal{P}})_2}{h_{\mathcal{P}}}.$$

We have now all the tools to define the three fundamental matrices $\mathbf{B}, \mathbf{D}, \mathbf{G}$. Let as usual $\{\varphi_i\}_{i=1}^{N^{\text{dof}}}$ indicate the basis for $V_k(E)$ that is dual to the degrees of freedom. Then, in the spirit of (3.14), the "right-hand side" matrix $\mathbf{B}$ is defined by

$$\mathbf{B} = \begin{bmatrix} \mathsf{P}_1\varphi_1 & \dots & \mathsf{P}_1\varphi_{N^{\text{dof}}} \\ \vdots & \ddots & \vdots \\ \mathsf{P}_{n_{\text{ker}}}\varphi_1 & \dots & \mathsf{P}_{n_{\text{ker}}}\varphi_{N^{\text{dof}}} \\ \mathcal{A}^E(\widetilde{m}_{n_{\text{ker}}+1}, \varphi_1) & \dots & \mathcal{A}^E(\widetilde{m}_{n_{\text{ker}}+1}, \varphi_{N^{\text{dof}}}) \\ \vdots & \ddots & \vdots \\ \mathcal{A}^E(\widetilde{m}_{n_k}, \varphi_1) & \dots & \mathcal{A}^E(\widetilde{m}_{n_k}, \varphi_{N^{\text{dof}}}) \end{bmatrix}.$$

Note that matrix $\mathbf{B}$ is computable due to the assumptions (C2)-(C3).
The "change of basis" matrix $\mathbf{D}$, see (3.17), is given by

$$\mathbf{D} = \begin{bmatrix} \text{dof}_1(\widetilde{m}_1) & \text{dof}_1(\widetilde{m}_2) & \dots & \text{dof}_1(\widetilde{m}_{n_k}) \\ \text{dof}_2(\widetilde{m}_1) & \text{dof}_2(\widetilde{m}_2) & \dots & \text{dof}_2(\widetilde{m}_{n_k}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{dof}_{N^{\text{dof}}}(\widetilde{m}_1) & \text{dof}_{N^{\text{dof}}}(\widetilde{m}_2) & \dots & \text{dof}_{N^{\text{dof}}}(\widetilde{m}_{n_k}) \end{bmatrix}.$$

Finally, the matrix $\mathbf{G}$ is defined by

$$\mathbf{G} := \begin{bmatrix} \mathsf{P}_1\widetilde{m}_1 & \dots & \mathsf{P}_1\widetilde{m}_{n_{\text{ker}}} & \mathsf{P}_1\widetilde{m}_{n_{\text{ker}}+1} & \dots & \mathsf{P}_1\widetilde{m}_{n_k} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathsf{P}_{n_{\text{ker}}}\widetilde{m}_1 & \dots & \mathsf{P}_{n_{\text{ker}}}\widetilde{m}_{n_{\text{ker}}} & \mathsf{P}_{n_{\text{ker}}}\widetilde{m}_{n_{\text{ker}}+1} & \dots & \mathsf{P}_{n_{\text{ker}}}\widetilde{m}_{n_k} \\ 0 & \dots & 0 & \mathcal{A}^E(\widetilde{m}_{n_{\text{ker}}+1}, \widetilde{m}_2) & \dots & \mathcal{A}^E(\widetilde{m}_{n_{\text{ker}}+1}, \widetilde{m}_{n_k}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \mathcal{A}^E(\widetilde{m}_{n_k}, \widetilde{m}_2) & \dots & \mathcal{A}^E(\widetilde{m}_{n_k}, \widetilde{m}_{n_k}) \end{bmatrix}.$$

Note that, for the same reason described in Remark 3.3, once the matrices $\mathbf{B}, \mathbf{D}$ are computed, the matrix $\mathbf{G}$ can be calculated simply as $\mathbf{G} = \mathbf{BD}$.

We can now define the operator Π^∇, acting from $V_k(E)$ to $\mathcal{P}_k(E)$, that is, the projection with respect to the bilinear form $\mathcal{A}$ and operators $\{\mathsf{P}_\alpha\}_{\alpha=1}^{n_{\text{ker}}}$. More precisely, for all $v_h \in V_k(E)$ the operator Π^∇ is defined by

$$\begin{cases} \mathsf{P}_\alpha\big(\Pi^\nabla v_h - v_h\big) = 0 & \alpha = 1, \dots, n_{\text{ker}}, \\ \mathcal{A}^E(\widetilde{m}_\alpha, \Big(\Pi^\nabla v_h - v_h\Big)) = 0 & \alpha = n_{\text{ker}} + 1, \dots, n_k. \end{cases}$$

Following the same arguments of Section 3.1, the matrix representation of the operator Π^∇ in the $\{\widetilde{m}_\alpha\}_{\alpha=1}^{n_k}$ basis of $\mathcal{P}_k(E)$ is

$$\mathbf{\Pi}_*^\nabla := \mathbf{G}^{-1}\mathbf{B},$$

while in the $\{\varphi_i\}_{i=1}^{N^{\text{dof}}}$ basis of $V_k(E)$ is

$$\mathbf{\Pi}^\nabla := \mathbf{D}\mathbf{G}^{-1}\mathbf{B}.$$

To conclude, the local VEM stiffness matrix is computed, as in (3.25), by

$$\mathbf{K}_E^h = (\mathbf{\Pi}_*^\nabla)^{\mathrm{T}}\widetilde{\mathbf{G}}(\mathbf{\Pi}_*^\nabla) + (\mathbf{I} - \mathbf{\Pi}^\nabla)^{\mathrm{T}}(\mathbf{I} - \mathbf{\Pi}^\nabla), \tag{8.1}$$

where $\widetilde{\mathbf{G}}$ is the matrix that coincides with $\mathbf{G}$ except for the first n_{ker} rows which are set to zero.

REFERENCES

[1] B. Ahmed, A. Alsaedi, F. Brezzi, L.D. Marini, and A. Russo. Equivalent Projectors for Virtual Element Methods. *Comput. Math. Appl.*, 66(3):376–391, 2013.

[2] P.F. Antonietti, N. Bigoni, and M. Verani. Mimetic Discretizations of Elliptic Control Problems. *J. Sci. Comput.*, 56(1):14–27, 2013.

[3] L. Beirão da Veiga. A residual based error estimator for the mimetic finite difference method. *Numer. Math.*, 108(3):387–406, 2008.

[4] L. Beirão da Veiga. A mimetic finite difference method for linear elasticity. *M2AN: Math. Model. Numer. Anal.*, 44(2):231–250, 2010.

[5] L. Beirão da Veiga, F. Brezzi, A. Cangiani, L.D. Marini, G. Manzini, and A. Russo. The basic principles of Virtual Elements Methods. *Math. Models Methods Appl. Sci.*, 23(1):199–214, 2013,

[6] L. Beirão da Veiga, F. Brezzi, and L.D. Marini. Virtual Elements for linear elasticity problems. *SIAM J. Num. Anal.*, 5(2):794–812, 2013.

[7] L. Beirão da Veiga, K. Lipnikov, and G. Manzini. Arbitrary-Order Nodal Mimetic Discretizations of Elliptic Problems on Polygonal Meshes. *SIAM J. Numer. Anal.*, 49(5):1737–1760, 2011.

[8] L. Beirão da Veiga and G. Manzini. A virtual element method with arbitrary regularity. *IMA J. Numer. Anal.*, in press, 2013. DOI:10.1093/imanum/drt018

[9] J. E. Bishop. A displacement-based finite element formulation for general polyhedra using harmonic shape functions. *Int. J. Numer. Meth. Engrg.*, published online 29 Aug 2013.

[10] P. Bochev and J. M. Hyman. Principle of mimetic discretizations of differential operators. In D. Arnold, P. Bochev, R. Lehoucq, R. Nicolaides, and M. Shashkov, editors, *Compatible discretizations. Proceedings of IMA hot topics workshop on compatible discretizations*, IMA Volume 142. Springer-Verlag, 2006.

[11] J. Bonelle and A. Ern. Analysis of compatible discrete operator schemes for elliptic problems on polyhedral meshes. *ArXiv preprint* arXiv:1211.3354, 2012.

[12] F. Brezzi, A. Buffa, and K. Lipnikov. Mimetic finite differences for elliptic problems. *M2AN: Math. Model. Numer. Anal.*, 43:277–295, 2009.

[13] F. Brezzi, R.S. Falk and L.D. Marini. Basic Principles of Mixed Virtual Element Methods. (Submitted to ESAIM Math. Model. Numer. Anal.).

[14] F. Brezzi, K. Lipnikov, and M. Shashkov. Convergence of mimetic finite difference method for diffusion problems on polyhedral meshes. *SIAM J. Num. Anal.*, 43:1872–1896, 2005.

[15] F. Brezzi, K. Lipnikov, M. Shashkov, and V. Simoncini. A new discretization methodology for diffusion problems on generalized polyhedral meshes. *Comp. Meth. Appl. Mech. Engrg.*, 196:3682–3692, 2007.

[16] F. Brezzi, K. Lipnikov, and V. Simoncini. A family of mimetic finite difference methods on polygonal and polyhedral meshes. *Math. Models Methods Appl. Sci.*, 15:1533–1553, 2005.

[17] F. Brezzi and L.D. Marini. Virtual elements for plate bending problems. *Comput. Methods Appl. Mech. Engrg.*, 253:455–462, 2013.

[18] A. Cangiani, G. Manzini, and A. Russo. Convergence analysis of a mimetic finite difference method for general second-order elliptic problems. *SIAM J. Num. Anal.*, 47(4):2612âĂŞ-2637, 2009.

[19] P. Chow, M. Cross, and K. Pericleous. A natural extension of the conventional finite volume method into polygonal unstructured meshes for CFD application. *Applied Mathematical Modeling* 20(2):170–183, 1996.

[20] J. Droniou , R. Eymard, T.R. Gallouët, and R. Herbin. A unified approach to Mimetic Finite Difference, Hybrid FiniteVolume and Mixed Finite Volume methods. *Math. Models Methods Appl. Sci.*, 20(2):265–295, 2010.

[21] M. S. Floater, K. Hormann, and G. Kós. A general construction of barycentric coordinates over convex polygons. *Advances in Computational Mathematics*, 24(14):311–331, 2006.

[22] J.M. Hyman, M. Shashkov. The Orthogonal Decomposition Theorems for Mimetic Finite Difference Methods. *SIAM J. Num. Anal.*, 36 (3), 788–818, 1999.

[23] J.M. Hyman, M. Shashkov, and S. Steinberg. The effect of inner products for discrete vector fields on the accuracy of mimetic finite difference methods. *Comput. Math. Appl.*, 42:1527–1547, 2001.

[24] Y. Kuznetsov and S. Repin. New mixed finite element method on polygonal and polyhedral meshes. *Russ. J. Numer. Anal. Math. Model*, 18(3): 261âĂŞ-278, 2003.

[25] K. Lipnikov, G. Manzini, and M. Shashkov. Mimetic finite difference method. Review paper, to appear in *J. Comput. Phys*.

[26] K. Lipnikov, J.D. Moulton, and D. Svyatskiy. A multiscale multilevel mimetic (M3) method for two-phase flows in porous media. *J. Comput. Phys.*, 227, 6727-6753, 2008.

[27] K. Lipnikov, M. Shashkov, I. Yotov. Local flux mimetic finite difference methods. *Numer. Math.*, 112:115–152, 2009.

[28] K. Lipnikov, D. Svyatskiy, and Y. Vassilevski. Interpolation-free monotone finite volume method for diffusion equations on polygonal meshes. *J. Comput. Phys.*, 228(3):703–716, 2009.

[29] S. E. Mousavi and N. Sukumar. Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons. *Comput. Mech.*, 47(5):535–554, 2011.

[30] S. Natarajan, S. Bordas, and D. R. Mahapatra. Numerical integration over arbitrary polygonal domains based on Schwarz-Christoffel conformal mapping. *Int. J. Numer. Meth. Engrg.*, 80(1):103–134, 2009.

[31] S. Rjasanow and S. Weißer. Higher order BEM-based FEM on polygonal meshes. *SIAM J. Num. Anal.*, 50(5):2357–2378, 2012.

[32] Z. Sheng and G. Yuan. An improved monotone finite volume scheme for diffusion equation on polygonal meshes. *J. Comput. Phys.*, 231(9):3739–3754, 2012.

[33] N. Sukumar and A. Tabarraei. Conforming polygonal finite elements. *Int. J. Numer. Meth. Engrg.*, 61(12):2045–2066, 2004.

[34] A. Tabarraei and N. Sukumar. Extended finite element method on polygonal and quadtree meshes. *Comput. Methods Appl. Mech. Engrg.*, 197(5):425–438, 2008.

[35] C. Talischi, G. H. Paulino, A. Pereira, and I.F.M. Menezes. Polygonal finite elements for topology optimization: A unifying paradigm. *Int. J. Numer. Methods. Engrg.*, 82:671âĂŞ-698, 2010.

[36] C. Talischi, G. H. Paulino, A. Pereira, and I.F.M. Menezes. PolyMesher: a general-purpose mesh generator for polygonal elements written in Matlab. *Struct. Multidisc Optimiz.*. 45(3):309–328, 2012.

[37] M. Vohralik and B. Wohlmuth. Mixed finite element methods: implementation with one unknown per element, local flux expressions, positivity, polygonal meshes, and relations to other methods. *Math. Models Methods Appl. Sci.*, 23(5):803–838, 2013.

[38] E. Wachspress. A Rational Finite Element Basis. Academic Press, New York, 1975.

[39] E. Wachspress. Barycentric coordinates for polytopes. *Comput. Math. Appl.*, 61(11):3319–3321, 2011.