Virtual Element Implementation for General Elliptic Equations

Lourenco Beirão da Veiga, Franco Brezzi, Luisa Donatella Marini, and Alessandro Russo

Abstract In the present paper we detail the implementation of the Virtual Element Method for two dimensional elliptic equations in primal and mixed form with variable coefficients.

1 Introduction

The Virtual Element Method (VEM) is a recent generalization of the Finite Element Method that, in addition to other useful features, can easily handle general polygonal and polyhedral meshes. The interest in numerical methods that can use polytopal elements has a long and relevant history. We just recall the review works [3, 4, 14, 21, 22, 26, 27] and the references therein. However, the use of polytopes showed recently a significant growth both in the mathematical and in the engineering literature, with the emergence of a new class of methods where the traditional approach (based on the approximation and/or numerical integration of test and trial functions) was substituted by various alternative strategies based on suitable different formulations. Among these alternative frameworks (all, deep inside, very similar to each other) we could see the (older) Mimetic Finite Differences (see e.g. [9]

L. Beirão da Veiga • A. Russo (✉)
Department of Mathematics and Applications, University of Milano-Bicocca, via Cozzi 57, 20125 Milano, Italy

IMATI-CNR, via Ferrata 1, 27100 Pavia, Italy
e-mail: lourenco.beirao@unimib.it; alessandro.russo@unimib.it

F. Brezzi
Istituto di Matematica Applicata e Tecnologie Informatiche - Pavia, Consiglio Nazionale delle Ricerche, Pavia, Italy
e-mail: brezzi@imati.cnr.it

L.D. Marini
Department of Mathematics, University of Pavia, via Ferrata 1, 27100 Pavia, Italy

IMATI-CNR, via Ferrata 1, 27100 Pavia, Italy
e-mail: marini@imati.cnr.it

G.R. Barrenechea et al. (eds.), *Building Bridges: Connections and Challenges in Modern Approaches to Numerical Partial Differential Equations*,
Lecture Notes in Computational Science and Engineering 114,
DOI 10.1007/978-3-319-41640-3_2

and the references therein), the Hybridizable Discontinuous Galerkin (see e.g. [18] and the references therein) the Gradient Schemes (see e.g. [20] and the references therein) the Weak Galerkin Methods (see e.g. [29] and the references therein), and the Hybrid High Order methods (see e.g. [19] and the references therein), together with the main object of the present paper: the Virtual Element Method.

The subject of polygonal and polyhedral mesh generation is a very active area of research on its own. Here we only refer to [28] for a simple and reliable MATLAB polygonal mesh generator in 2D, and to [24] and the references therein for some insights into the issues of the three-dimensional case.

Very briefly, the key idea of the Virtual Element Method is to adopt also non-polynomial shape functions (that are necessary in order to build conforming discrete spaces on complex polytopal grids) but avoiding their explicit computation, not even in an approximate way. This is achieved by introducing the right set of degrees of freedom and defining computable projection operators on polynomial spaces. In the initial paper [6] the Virtual Element Method was presented for the two dimensional Poisson problem in primal form, while the three dimensional case (still for constant coefficients) was discussed later in [1]. In the more recent papers [12] and [11] the Virtual Element Method was then extended to more general elliptic equations (including variable coefficients with the possible presence of convection and reaction term), respectively in primal and mixed form. At the same time, the method has been applied with success to a wide range of other problems. We just recall [2, 5, 7, 10, 13, 15–17, 23, 25].

The present work can be considered as a natural continuation of [8], where all the coding aspects of the model scheme presented in [6] and [1] where detailed. Here we describe all the tools for the practical implementation of the methods analysed in [12] and [11]. Since the assembly of the global matrix follows the same identical procedure as in the Finite Element case, the focus of this work is on the construction of the local matrices. After a brief description of the discrete spaces and the associated degrees of freedom, we detail step by step the implementation of the projection operators and all the other involved matrices. At the end of each part the reader can find an “algorithm” section where the whole procedure is summarized. Although we believe that the VEM is very elegant and, once some familiarity is acquired, quite easy to implement, we advice the reader to look into the previous work [8] before reading the present one.

The paper is organized as follows. After presenting some minimal notation in Sect. 2, we briefly describe in Sect. 3 the problem under consideration, including its primal and mixed variational formulations. In Sects. 4 and 5 we briefly recall the discrete spaces, the degrees of freedom and the construction of the projection operator of [6]. In Sect. 6 we detail the implementation of the method analysed in [12]; a useful summary can be found in Sect. 7. Section 8 is devoted to a brief description of the discrete spaces and of the degrees of freedom introduced in [11], while the implementation aspects are described in Sects. 9 and 10. A useful summary can be found in Sect. 11.

In this paper we have studied in details the implementation of the Virtual Element Method in two dimensions only. The extension to the three dimensional case does not present any major difficulties, as long as all the 2D machinery is developed with respect to each face of a general polyhedron. We will soon release a full MATLAB implementation for both the 2D and the 3D case.

2 Basic Notation

In the present section we introduce some minimal notation needed in the rest of the paper.

2.1 *Polynomial Spaces*

For a given a domain $\mathcal{D} \subset \mathbb{R}^d$ and an integer $k \geqslant 1$, we will denote by $\mathcal{P}_k(\mathcal{D})$ the linear space of polynomials of degree less than or equal to k. When $d = 2$, the dimension of $\mathcal{P}_k(\mathcal{D})$ will be denoted by n_k:

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

2.2 *Polygons*

A generic polygon will be denoted by E; the number of vertices will be denoted by N_V and the number of edges by N_e. Of course $N_e = N_V$, but it will be useful to keep separate names. The diameter of the polygon E will be denoted by h_E and its centroid by (x_c, y_c). The outward normal to E will be denoted by $\boldsymbol{n}_E$ or simply by $\boldsymbol{n}$ when no confusion can arise. The normal $\boldsymbol{n}_E$ restricted to ad edge e will be indicated by $\boldsymbol{n}_e$.

2.3 *Scaled Monomials*

Let $\boldsymbol{\alpha} = (\alpha_x, \alpha_y)$ be a multi-index. We define the *scaled monomial* $m_{\boldsymbol{\alpha}}$ on E by:

$$m_{\boldsymbol{\alpha}}(x, y) := \left(\frac{x - x_c}{h_E}\right)^{\alpha_x} \left(\frac{y - y_c}{h_E}\right)^{\alpha_y}. \tag{1}$$

For k an integer, let

$$\mathcal{M}_k(E) := \{m_{\boldsymbol{\alpha}},\ 0 \leqslant |\boldsymbol{\alpha}| \leqslant k\} \tag{2}$$

where $|\boldsymbol{\alpha}| = \alpha_x + \alpha_y$. With a small abuse of notation we will indicate with α (in contrast with boldface $\boldsymbol{\alpha}$) a linear index running from 1 to n_k. Obviously, $\mathcal{M}_k(E)$ is a basis for $\mathcal{P}_k(E)$.

2.4 Functional Spaces

The scalar product in $L^2(\mathcal{D})$ will be denoted by $(\cdot,\cdot)_{0,\mathcal{D}}$ or simply by $(\cdot,\cdot)$ when the domain is clear from the context.

3 The Elliptic Problem

Let $\Omega \subset \mathbb{R}^2$ be a bounded convex polygonal domain with boundary Γ, let κ and γ be smooth functions $\Omega \to \mathbb{R}$ with $\kappa(\boldsymbol{x}) \geqslant \kappa_0 > 0$ for all $\boldsymbol{x} \in \Omega$, and let $\boldsymbol{b}$ be a smooth vector field $\Omega \to \mathbb{R}^2$. We consider the following elliptic problem:

$$\begin{cases} \mathcal{L}p := \operatorname{div}(-\kappa\nabla p + \boldsymbol{b}p) + \gamma\, p = f & \text{in } \Omega \\ p = 0 & \text{on } \Gamma. \end{cases} \tag{3}$$

We assume that problem (3) is solvable for any $f \in H^{-1}(\Omega)$, and that the a-priori estimate

$$\|p\|_{1,\Omega} \leqslant C\|f\|_{-1,\Omega} \tag{4}$$

and the regularity estimate

$$\|p\|_{2,\Omega} \leqslant C\|f\|_{0,\Omega} \tag{5}$$

hold with a constant C independent of f. As shown in [12] and [11], these hypotheses are sufficient to prove the convergence of the Virtual Element approximation, both in primal and in mixed form.

3.1 The Primal Variational Formulation

Set:

$$a(p,q) := \int_\Omega \kappa\,\nabla p\cdot\nabla q\,\mathrm{d}\boldsymbol{x}, \quad b(p,q) := -\int_\Omega p\,(\boldsymbol{b}\cdot\nabla q)\,\mathrm{d}\boldsymbol{x},$$
$$c(p,q) := \int_\Omega \gamma\,p\,q\,\mathrm{d}\boldsymbol{x}, \quad (f,q) = \int_\Omega f\,q\,\mathrm{d}\boldsymbol{x},$$

and define

$$B(p,q) := a(p,q) + b(p,q) + c(p,q). \tag{6}$$

The primal variational formulation of problem (3) is then

$$\begin{cases} \text{find } p \in V := H_0^1(\Omega) \quad \text{such that} \\ B(p,q) = (f,q) \quad \text{for all } q \in V. \end{cases} \tag{7}$$

3.2 The Mixed Variational Formulation

In order to build the mixed variational formulation of problem (3), we define

$$\nu := \kappa^{-1}, \quad \boldsymbol{\beta} := \kappa^{-1}\boldsymbol{b},$$

and re-write (3) as

$$\boldsymbol{u} = \nu^{-1}(-\nabla p + \boldsymbol{\beta} p), \quad \operatorname{div} \boldsymbol{u} + \gamma p = f \text{ in } \Omega, \quad p = 0 \text{ on } \Gamma. \tag{8}$$

Introducing the spaces

$$\boldsymbol{V} := \boldsymbol{H}(\operatorname{div};\Omega), \quad \text{and} \quad Q := L^2(\Omega),$$

the mixed variational formulation of problem (3) is:

$$\begin{cases} \text{Find } (\boldsymbol{u},p) \in \boldsymbol{V} \times Q \text{ such that} \\ (\nu\boldsymbol{u},\boldsymbol{v}) - (p, \operatorname{div} \boldsymbol{v}) - (\boldsymbol{\beta}\cdot\boldsymbol{v}, p) = 0 \quad \text{for all } \boldsymbol{v} \in \boldsymbol{V}, \\ (\operatorname{div} \boldsymbol{u}, q) + (\gamma p, q) = (f,q) \quad \text{for all } q \in Q. \end{cases} \tag{9}$$

4 Approximation with the Virtual Element Method

The Virtual Element approximation of problems (7) and (9) fits in the classical conforming Galerkin methods: in principle, in both cases we define finite-dimensional subspaces $V_h \subset V$ (for problem (7)) and $\boldsymbol{V}_h \subset \boldsymbol{V}$, $Q_h \subset Q$ (for problem (9)) and we restrict the various bilinear forms to the spaces V_h and $\boldsymbol{V}_h \times Q_h$ respectively. However, given that for the VEM the functions are not explicitly known, we will also have to *approximate* the various bilinear forms.

As usual, the virtual spaces V_h, $\boldsymbol{V}_h$ and Q_h will be defined at the element level, and on the boundary of the elements the degrees of freedom will be chosen in such a way that they will nicely glue together.

Hence, given a polygon E of the decomposition, we will first define the local virtual spaces $V_h(E)$, $\boldsymbol{V}_h(E)$ and $Q_h(E)$ and then we will set

$$V_h = \{ p \in V \text{ such that } p_{|E} \in V_h(E)\} \tag{10}$$

$$\boldsymbol{V}_h = \{\boldsymbol{v} \in \boldsymbol{V} \text{ such that } \boldsymbol{v}_{|E} \in \boldsymbol{V}_h(E)\} \tag{11}$$

$$Q_h = \{q \in Q \text{ such that } q_{|E} \in Q_h(E)\}. \tag{12}$$

Also the approximation of the various bilinear forms will be made element by element.

To encourage the reader, we point out that the space Q_h will consist, as usual in finite element methods, of piecewise discontinuous polynomials of degree k.

5 Virtual Element Space for the Primal Formulation

Before defining the local virtual space $V_h(E)$, we need to become familiar with the projection operator Π_k^∇ which will play a major role in the rest of the paper.

The operator Π_k^∇ is the orthogonal projection onto the space of polynomials of degree k with respect to the scalar product $\int_E \nabla p \cdot \nabla q \,\mathrm{d}\boldsymbol{x}$. Given a function $p \in H^1(E)$, the polynomial $\Pi_k^\nabla p$ is defined by the condition

$$\int_E \nabla(\Pi_k^\nabla p - p) \cdot \nabla r_k \,\mathrm{d}\boldsymbol{x} = 0 \quad \text{for all } r_k \in \mathcal{P}_k(E). \tag{13}$$

When r_k is a constant, condition (13) is the identity $0 \equiv 0$ so the polynomial $\Pi_k^\nabla p$ itself is determined up to a constant. This is fixed by imposing an extra condition, for instance,

$$\int_{\partial E} (\Pi_k^\nabla p - p) \,\mathrm{d}s = 0. \tag{14}$$

The following easy lemma will be useful throughout the section:

Lemma 1 *The polynomial $\Pi_k^\nabla p$ depends only on*

- *the value of p on the boundary of E;*
- *the moments of p in E up to order $k-2$.*

Proof By Eqs. (13) and (14) it is clear that the polynomial $\Pi_k^\nabla p$ is completely determined by the integrals

$$\int_E \nabla p \cdot \nabla r_k \,\mathrm{d}\boldsymbol{x} \qquad \text{and} \qquad \int_{\partial E} p \,\mathrm{d}s.$$

The second integral clearly depends only on the value of p on the boundary of E. Concerning the first integral, integrating by parts we have

$$\int_E \nabla p \cdot \nabla r_k \, \mathrm{d}\boldsymbol{x} = -\int_E p \, \Delta r_k \, \mathrm{d}\boldsymbol{x} + \int_{\partial E} p \, \frac{\partial r_k}{\partial n} \, \mathrm{d}s$$

and since $\Delta r_k \in \mathcal{P}_{k-2}(E)$ the proof is completed.

We are now ready to introduce the local virtual space $V_h(E)$. The space $V_h(E)$ consists of functions p_h such that:

- p_h is continuous on E;
- p_h on each edge e is a polynomial of degree k;
- $\Delta p_h \in \mathcal{P}_k(E)$;
- $\displaystyle\int_E p_h \, m_{\boldsymbol{\alpha}} \, \mathrm{d}\boldsymbol{x} = \int_E \Pi_k^{\nabla} p_h \, m_{\boldsymbol{\alpha}} \, \mathrm{d}\boldsymbol{x}$ for $|\boldsymbol{\alpha}| = n_k - 1$ and $|\boldsymbol{\alpha}| = n_k$.

In [1, 8] we have shown the following results:

1. $V_h(E)$ has dimension $N_V + (k-1)N_e + n_{k-2} = kN_V + n_{k-2}$;
2. $\mathcal{P}_k(E) \subset V_h(E)$;
3. for the space $V_h(E)$ we can take the following degrees of freedom:

Boundary degrees of freedom [$\boldsymbol{N_V + (k-1) \times N_e = k \times N_V}$]

- the values of p_h at the N_V vertices of the polygon E;
- for each edge e, the values of p_h at $k-1$ distinct points of e (for instance equispaced points).

Internal degrees of freedom (only for $\boldsymbol{k > 1}$) [$\boldsymbol{n_{k-2}}$]

- the moments of p_h up to degree $k-2$, i.e. the integrals

$$\frac{1}{|E|} \int_E p_h \, m_{\boldsymbol{\alpha}} \, \mathrm{d}\boldsymbol{x}, \quad |\boldsymbol{\alpha}| \leqslant k-2.$$

We will indicate by $\mathrm{dof}_i(p_h)$ ($i = 1, \ldots, N_{\mathrm{dof}} := \dim V_h(E)$) the degrees of freedom of p_h. We define the *local basis functions* $\phi_i \in V_h(E)$, $i = 1, \ldots, N_{\mathrm{dof}}$, by the property:

$$\mathrm{dof}_i(\phi_j) = \delta_{ij} \quad i,j = 1, \ldots, N_{\mathrm{dof}} \tag{15}$$

so that we have a Lagrange-type decomposition:

$$p_h = \sum_{i=1}^{N_{\mathrm{dof}}} \mathrm{dof}_i(p_h) \, \phi_i. \tag{16}$$

Given a function $p_h \in V_h(E)$, by Lemma 1 the polynomial $\Pi_k^\nabla p_h$ depends only on the value of p_h on the boundary of E and on the moments of p_h in E up to order $k-2$. Hence, the polynomial $\Pi_k^\nabla p_h$ depends only on the degrees of freedom of p_h. In [8] it is shown that also the L^2 projection $\Pi_k^0 p_h$ of a function $p_h \in V_h(E)$ onto $\mathcal{P}_k(E)$ depends only on its degrees of freedom, and all the details to compute and code $\Pi_k^\nabla \phi_i$ and $\Pi_k^0 \phi_i$, for a generic basis function ϕ_i, are given. For the convenience of the reader we report here the various steps. Write

$$\Pi_k^\nabla \phi_i = \sum_{\alpha=1}^{n_k} s_i^\alpha m_\alpha, \quad i = 1, \dots N_{\text{dof}} \tag{17}$$

and define

$$\mathrm{P}_0 \phi_i := \int_{\partial E} \phi_i \, \mathrm{d}s.$$

Then, defining

$$\mathbf{G} = \begin{bmatrix} \mathrm{P}_0 m_1 & \mathrm{P}_0 m_2 & \dots & \mathrm{P}_0 m_{n_k} \\ 0 & (\nabla m_2, \nabla m_2)_{0,E} & \dots & (\nabla m_2, \nabla m_{n_k})_{0,E} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & (\nabla m_{n_k}, \nabla m_2)_{0,E} & \dots & (\nabla m_{n_k}, \nabla m_{n_k})_{0,E} \end{bmatrix}, \tag{18}$$

$$\boldsymbol{b}_i = \begin{bmatrix} \mathrm{P}_0 \phi_i \\ (\nabla m_2, \nabla \phi_i)_{0,E} \\ \vdots \\ (\nabla m_{n_k}, \nabla \phi_i)_{0,E} \end{bmatrix}, \tag{19}$$

for each i, the coefficients s_i^α, $\alpha = 1, \dots, n_k$ are solution of the $n_k \times n_k$ linear system:

$$\mathbf{G} \boldsymbol{s}_i = \boldsymbol{b}_i.$$

Denoting by $\mathbf{B}$ the $n_k \times N_{\text{dof}}$ matrix given by

$$\mathbf{B} := \left[\boldsymbol{b}_1 \, \boldsymbol{b}_2 \, \dots \, \boldsymbol{b}_{N_{\text{dof}}} \right] = \begin{bmatrix} \mathrm{P}_0 \phi_1 & \dots & \mathrm{P}_0 \phi_{N_{\text{dof}}} \\ (\nabla m_2, \nabla \phi_1)_{0,E} & \dots & (\nabla m_2, \nabla \phi_{N_{\text{dof}}})_{0,E} \\ \vdots & \ddots & \vdots \\ (\nabla m_{n_k}, \nabla \phi_1)_{0,E} & \dots & (\nabla m_{n_k}, \nabla \phi_{N_{\text{dof}}})_{0,E} \end{bmatrix}, \tag{20}$$

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

$$\overset{*}{\boldsymbol{\Pi}}{}^{\nabla}_{k} = \mathbf{G}^{-1}\mathbf{B}. \tag{21}$$

We will also need the matrix representation, in the basis (15), of the same operator Π^{∇}_k, this time thought as an operator $V_h(E) \longrightarrow V_h(E)$. Hence, let

$$\Pi^{\nabla}_k \phi_i = \sum_{j=1}^{N_{\mathrm{dof}}} \pi_i^j \phi_j, \quad i = 1, \dots N_{\mathrm{dof}},$$

with

$$\pi_i^j = \mathrm{dof}_j\big(\Pi^{\nabla}_k \phi_i\big).$$

From (17) and (16) we have

$$\Pi^{\nabla}_k \phi_i = \sum_{\alpha=1}^{n_k} s_i^\alpha m_\alpha = \sum_{\alpha=1}^{n_k} s_i^\alpha \sum_{j=1}^{N_{\mathrm{dof}}} \mathrm{dof}_j(m_\alpha)\, \phi_j$$

so that

$$\pi_i^j = \sum_{\alpha=1}^{n_k} s_i^\alpha\, \mathrm{dof}_j(m_\alpha). \tag{22}$$

In order to express (22) in matrix form, we define the $N_{\mathrm{dof}} \times n_k$ matrix $\mathbf{D}$ by:

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

that is,

$$\mathbf{D} = \begin{bmatrix} \mathrm{dof}_1(m_1) & \mathrm{dof}_1(m_2) & \dots & \mathrm{dof}_1(m_{n_k}) \\ \mathrm{dof}_2(m_1) & \mathrm{dof}_2(m_2) & \dots & \mathrm{dof}_2(m_{n_k}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{dof}_{N_{\mathrm{dof}}}(m_1) & \mathrm{dof}_{N_{\mathrm{dof}}}(m_2) & \dots & \mathrm{dof}_{N_{\mathrm{dof}}}(m_{n_k}) \end{bmatrix}. \tag{23}$$

Equation (22) 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* $\mathbf{\Pi}_k^{\nabla}$ of the operator $\Pi_k^{\nabla} : V_h(E) \longrightarrow V_h(E)$ in the basis (15), is given by

$$\mathbf{\Pi}_k^{\nabla} = \mathbf{D}\mathbf{G}^{-1}\mathbf{B} = \mathbf{D}\overset{*}{\mathbf{\Pi}}{}_k^{\nabla}. \tag{24}$$

Remark 1 We point out that, as shown in [8], the matrix $\mathbf{G}$ can be expressed in terms of the matrices $\mathbf{D}$ and $\mathbf{B}$ as

$$\mathbf{G} = \mathbf{B}\mathbf{D}. \tag{25}$$

Always following [8], we can show that also the L^2 projection onto $\mathcal{P}_k(E)$ of a function $p_h \in V_h(E)$ depends only on its degrees of freedom. If we write

$$\Pi_k^0 \phi_i = \sum_{i=1}^{N_{\text{dof}}} t_i^{\alpha} m_{\alpha},$$

and define

$$\mathbf{H} = \begin{bmatrix} (m_1, m_1)_{0,E} & (m_1, m_2)_{0,E} & \dots & (m_1, m_{n_k})_{0,E} \\ (m_2, m_1)_{0,E} & (m_2, m_2)_{0,E} & \dots & (m_2, m_{n_k})_{0,E} \\ \vdots & \vdots & \ddots & \vdots \\ (m_{n_k}, m_1)_{0,E} & (m_{n_k}, m_2)_{0,E} & \dots & (m_{n_k}, m_{n_k})_{0,E} \end{bmatrix}, \tag{26}$$

$$\boldsymbol{c}_i = \begin{bmatrix} (m_1, \phi_i)_{0,E} \\ (m_2, \phi_i)_{0,E} \\ \vdots \\ (m_{n_k}, \phi_i)_{0,E} \end{bmatrix}, \tag{27}$$

then, for each i, the coefficients t_i^{α}, $\alpha = 1, \dots, n_k$ are solution of the $n_k \times n_k$ linear system:

$$\mathbf{H}\,\boldsymbol{t}_i = \boldsymbol{c}_i, \tag{28}$$

which descends directly from the definition of the L^2-projection.

We denote by $\mathbf{C}$ the $n_k \times N_{\text{dof}}$ matrix given by

$$\mathbf{C} := \left[\boldsymbol{c}_1\ \boldsymbol{c}_2\ \dots\ \boldsymbol{c}_{N_{\text{dof}}}\right] = \begin{bmatrix} (m_1, \phi_1)_{0,E} & (m_1, \phi_2)_{0,E} & \dots & (m_1, \phi_{N_{\text{dof}}})_{0,E} \\ (m_2, \phi_1)_{0,E} & (m_2, \phi_2)_{0,E} & \dots & (m_2, \phi_{N_{\text{dof}}})_{0,E} \\ \vdots & \vdots & \ddots & \vdots \\ (m_{n_k}, \phi_1)_{0,E} & (m_{n_k}, \phi_2)_{0,E} & \dots & (m_{n_k}, \phi_{N_{\text{dof}}})_{0,E} \end{bmatrix}. \tag{29}$$

The first n_{k-2} lines of the matrix $\mathbf{C}$ can be computed directly from the degrees of freedom, and the resulting matrix is

$$\text{first } n_{k-2} \text{ lines of } \mathbf{C} = |E| \left[\begin{array}{ccccc:cccc} 0 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 1 \end{array}\right]$$

where the rightmost block is the identity matrix of size $n_{k-2} \times n_{k-2}$. The last $n_k - n_{k-2}$ lines of the matrix $\mathbf{C}$ correspond to m_α being a monomial of degree $k-1$ or k and we need to resort to the fundamental property

$$\int_E \phi_i \, m_\alpha \, \mathrm{d}\boldsymbol{x} = \int_E \Pi_k^\nabla \phi_i \, m_\alpha \, \mathrm{d}\boldsymbol{x}.$$

Hence in this case we have

$$\mathbf{C}_{\alpha i} = (\mathbf{H}\mathbf{G}^{-1}\mathbf{B})_{\alpha i}, \quad n_{k-2} < \alpha \leqslant n_k.$$

It follows that the matrix representation $\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0}}_{\boldsymbol{k}}$ of the operator Π_k^0 acting from $V_h(E)$ to $\mathcal{P}_k(E)$ in the basis $\mathcal{M}_k(E)$ is given by $(\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0}}_{\boldsymbol{k}})_{\alpha i} = t_i^\alpha$, that is,

$$\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0}}_{\boldsymbol{k}} = \mathbf{H}^{-1}\mathbf{C}. \tag{30}$$

Arguing as before, the matrix representation, in the basis (15), of the same operator Π_k^0, this time thought as an operator $V_h(E) \longrightarrow V_h(E)$, is

$$\boldsymbol{\Pi}^{\mathbf{0}}_{\boldsymbol{k}} = \mathbf{D}\mathbf{H}^{-1}\mathbf{C} = \mathbf{D}\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0}}_{\boldsymbol{k}}. \tag{31}$$

In a similar fashion we can also compute the matrix representations $\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0}}_{\boldsymbol{k-1}}$ and $\boldsymbol{\Pi}^{\mathbf{0}}_{\boldsymbol{k-1}}$ of the L^2 projection onto the space of polynomials of degree $k-1$. To this end, we consider:

- the $n_{k-1} \times n_{k-1}$ matrix $\mathbf{H}'$ obtained by taking the first n_{k-1} rows and the first n_{k-1} columns of the matrix $\mathbf{H}$ defined in (26);
- the $n_{k-1} \times N_{\mathrm{dof}}$ matrix $\mathbf{C}'$ obtained by taking the first n_{k-1} lines of the matrix $\mathbf{C}$ defined in (29);
- the $N_{\mathrm{dof}} \times n_{k-1}$ matrix $\mathbf{D}'$ obtained by taking the first n_{k-1} columns of the matrix $\mathbf{D}$ defined in (23).

Then we have:

$$\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0}}_{\boldsymbol{k-1}} = (\mathbf{H}')^{-1}\mathbf{C}' \quad \text{and} \quad \boldsymbol{\Pi}^{\mathbf{0}}_{\boldsymbol{k-1}} = \mathbf{D}' \, \overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0}}_{\boldsymbol{k-1}}.$$

To summarize, given a "virtual" function $p_h \in V_h(E)$, we can compute the polynomials $\Pi_k^\nabla p_h$, $\Pi_k^0 p_h$ and $\Pi_{k-1}^0 p_h$ in terms of its degrees of freedom.

6 VEM Approximation of the Primal Formulation

As shown in [6], the projectors Π_k^∇ and Π_{k-1}^0 allow us to solve the Laplace equation with a reaction term. Indeed, according to [1], if problem (3) reduces to

$$\begin{cases} -\Delta p + \gamma p = f & \text{in } \Omega \\ u = g & \text{on } \partial\Omega \end{cases}$$

then we have

$$a(p,q) := \int_\Omega \nabla p \cdot \nabla q \,\mathrm{d}\boldsymbol{x}, \quad b(p,q) := 0, \quad c(p,q) := \int_\Omega \gamma\, p\, q \,\mathrm{d}\boldsymbol{x}.$$

The local VEM approximation for $a(\cdot,\cdot)$ is

$$a_h^E(p_h,q_h) := \int_E \nabla \Pi_k^\nabla p_h \cdot \nabla \Pi_k^\nabla q_h \,\mathrm{d}\boldsymbol{x} + \mathcal{S}_E\big((I-\Pi_k^\nabla)p_h, (I-\Pi_k^\nabla)q_h\big)$$

where the stability term $\mathcal{S}_E(\cdot,\cdot)$ is the symmetric and positive definite bilinear form which is the identity on the basis function, i.e. $\mathcal{S}_E(\phi_i,\phi_j) = \delta_{ij}$. The local VEM approximation for $c(\cdot,\cdot)$ is

$$c_h^E(p_h,q_h) := \int_E \gamma\, \Pi_{k-1}^0 p_h\, \Pi_{k-1}^0 q_h \,\mathrm{d}\boldsymbol{x}$$

and similarly the load term (f,q_h) is approximated locally by $(f, \Pi_{k-1}^0 q_h)_{0,E}$.

If the diffusion κ is not constant or a first-order term is present, then we cannot simply approximate ∇p_h with $\nabla \Pi_k^\nabla p_h$; as shown in [12], we would loose the optimal convergence rates. Instead, we should approximate

$$\nabla p_h \qquad \text{with} \qquad \Pi_{k-1}^0 \nabla p_h.$$

Note that for $k=1$ the two approximations of ∇p_h coincide; in fact,

$$\nabla \Pi_1^\nabla p_h = \frac{1}{|E|}\int_E \nabla p_h \,\mathrm{d}\boldsymbol{x} = \Pi_0^0 \nabla p_h.$$

We will see now how to compute $\Pi_{k-1}^0 \nabla p_h$ in terms of the degrees of freedom. To this end, we observe that in order to obtain $\Pi_{k-1}^0 \nabla p_h$, we need to compute

$$\int_E \nabla p_h \cdot \boldsymbol{r}_{k-1} \,\mathrm{d}\boldsymbol{x}$$

where $\boldsymbol{r}_{k-1}$ is any vector whose components are polynomials of degree $k-1$. Integrating by parts, we have

$$\int_E \nabla p_h \cdot \boldsymbol{r}_{k-1}\, \mathrm{d}\boldsymbol{x} = -\int_E p_h \operatorname{div} \boldsymbol{r}_{k-1}\, \mathrm{d}\boldsymbol{x} + \int_{\partial E} p_h \boldsymbol{r}_{k-1} \cdot \boldsymbol{n}\, \mathrm{d}s$$

and since $\operatorname{div} \boldsymbol{r}_{k-1} \in \mathcal{P}_{k-2}(E)$, both integrals are directly computable from the degrees of freedom of p_h. In order to find the matrix representations of the operator $\Pi^0_{k-1}\nabla$, we define the $n_{k-1} \times N_{\mathrm{dof}}$ matrix $\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{x}}_{\boldsymbol{k-1}}$ by

$$\Pi^0_{k-1}\phi_{i,x} = \sum_{\alpha=1}^{n_{k-1}} \big(\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{x}}_{\boldsymbol{k-1}}\big)_{\alpha i}\, m_\alpha. \tag{32}$$

The polynomial $\Pi^0_{k-1}\phi_{i,x}$ is defined by

$$\int_E \Pi^0_{k-1}\phi_{i,x}\, m_\beta\, \mathrm{d}\boldsymbol{x} = \int_E \phi_{i,x}\, m_\beta\, \mathrm{d}\boldsymbol{x}, \quad \beta = 1, \dots, n_{k-1}$$

which becomes the linear system

$$\sum_{\alpha=1}^{n_{k-1}} \big(\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{x}}_{\boldsymbol{k-1}}\big)_{\alpha i} \int_E m_\alpha\, m_\beta\, \mathrm{d}\boldsymbol{x} = \int_E \phi_{i,x}\, m_\beta\, \mathrm{d}\boldsymbol{x}, \quad \beta = 1, \dots, n_{k-1}.$$

The term $\int_E \phi_{i,x}\, m_\beta\, \mathrm{d}\boldsymbol{x}$ can be computed integrating by parts:

$$\int_E \phi_{i,x}\, m_\beta\, \mathrm{d}\boldsymbol{x} = -\int_E \phi_i\, m_{\beta,x}\, \mathrm{d}\boldsymbol{x} + \int_{\partial E} \phi_i\, m_\beta\, \boldsymbol{n}_x. \tag{33}$$

If we define the matrices $\mathbf{E}^{\boldsymbol{x}}$ and $\mathbf{E}^{\boldsymbol{y}}$ by

$$\big(\mathbf{E}^{\boldsymbol{x}}\big)_{i\beta} = \int_E \phi_{i,x}\, m_\beta\, \mathrm{d}\boldsymbol{x}, \quad \big(\mathbf{E}^{\boldsymbol{y}}\big)_{i\beta} = \int_E \phi_{i,y}\, m_\beta\, \mathrm{d}\boldsymbol{x}, \quad \beta = 1, \dots n_{k-1} \tag{34}$$

then we have:

$$\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{x}}_{\boldsymbol{k-1}} = \hat{\mathbf{H}}^{-1}\mathbf{E}^{\boldsymbol{x}}, \quad \overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{y}}_{\boldsymbol{k-1}} = \hat{\mathbf{H}}^{-1}\mathbf{E}^{\boldsymbol{y}}$$

where $\hat{\mathbf{H}}$ is the submatrix of $\mathbf{H}$ defined in (26) obtained taking the first n_{k-1} rows and columns of $\mathbf{H}$.

We can now compute the local VEM stiffness matrices for the variable coefficient case.

6.1 Diffusion Term

We have:

$$(\mathbf{K}^a)_{ij} := a_h^E(\phi_j, \phi_i) = \int_E \kappa\, \Pi^0_{k-1}\nabla\phi_j \cdot \Pi^0_{k-1}\nabla\phi_i \,\mathrm{d}\boldsymbol{x}$$
$$+ \bar{\kappa}\, \mathcal{S}_E\big((I - \Pi^\nabla_k)\phi_j, (I - \Pi^\nabla_k)\phi_i\big)$$

where $\bar{\kappa}$ is a constant approximation of κ (for instance, the mean value). We compute separately the consistency term and the stability term.

- **consistency term:**

$$(\mathbf{K}^a_\mathbf{c})_{ij} := \int_E \kappa\, \Pi^0_{k-1}\nabla\phi_j \cdot \Pi^0_{k-1}\nabla\phi_i \,\mathrm{d}\boldsymbol{x}$$
$$= \int_E \kappa \left\{[\Pi^0_{k-1}\phi_{j,x}][\Pi^0_{k-1}\phi_{i,x}] + [\Pi^0_{k-1}\phi_{j,y}][\Pi^0_{k-1}\phi_{i,y}]\right\} \mathrm{d}\boldsymbol{x}$$

and

$$\int_E \kappa\, [\Pi^0_{k-1}\phi_{j,x}][\Pi^0_{k-1}\phi_{i,x}]\,\mathrm{d}\boldsymbol{x} = \sum_{\alpha,\beta=1}^{n_{k-1}} \big(\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0},x}_{k-1}\big)_{\alpha j}\big(\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0},x}_{k-1}\big)_{\beta i} \int_E \kappa\, m_\alpha\, m_\beta \,\mathrm{d}\boldsymbol{x},$$

$$\int_E \kappa\, [\Pi^0_{k-1}\phi_{j,y}][\Pi^0_{k-1}\phi_{i,y}]\,\mathrm{d}\boldsymbol{x} = \sum_{\alpha,\beta=1}^{n_{k-1}} \big(\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0},y}_{k-1}\big)_{\alpha j}\big(\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0},y}_{k-1}\big)_{\beta i} \int_E \kappa\, m_\alpha\, m_\beta \,\mathrm{d}\boldsymbol{x}.$$

If we define the $n_{k-1} \times n_{k-1}$ matrix $\mathbf{H}^\kappa$ by

$$(\mathbf{H}^\kappa)_{\alpha\beta} := \int_E \kappa\, m_\alpha\, m_\beta \,\mathrm{d}\boldsymbol{x}, \quad 1 \leqslant \alpha, \beta \leqslant n_{k-1},$$

then we have

$$\mathbf{K}^a_\mathbf{c} = \big(\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0},x}_{k-1}\big)^{\mathrm{T}} \mathbf{H}^\kappa\, \overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0},x}_{k-1} + \big(\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0},y}_{k-1}\big)^{\mathrm{T}} \mathbf{H}^\kappa\, \overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0},y}_{k-1}$$

which can be written as

$$\mathbf{K}^a_\mathbf{c} = \left[\big(\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0},x}_{k-1}\big)^{\mathrm{T}} \;\; \big(\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0},y}_{k-1}\big)^{\mathrm{T}}\right] \begin{bmatrix} \mathbf{H}^\kappa & \mathbf{0} \\ \mathbf{0} & \mathbf{H}^\kappa \end{bmatrix} \begin{bmatrix} \overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0},x}_{k-1} \\ \overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0},y}_{k-1} \end{bmatrix}. \tag{35}$$

- **stability term:**

$$
\begin{aligned}
(\mathsf{K}_{\mathsf{s}}^{a})_{ij} &:= \bar{\kappa}\, \mathcal{S}_E\big((I-\Pi_k^{\nabla})\phi_j, (I-\Pi_k^{\nabla})\phi_i\big) \\
&= \bar{\kappa} \sum_{k,\ell=1}^{N_{\text{dof}}} \big(\delta_{jk} - (\boldsymbol{\Pi}_k^{\boldsymbol{\nabla}})_{jk}\big)\, \mathcal{S}_E(\phi_k, \phi_\ell)\, \big(\delta_{i\ell} - (\boldsymbol{\Pi}_k^{\boldsymbol{\nabla}})_{i\ell}\big) \\
&= \bar{\kappa} \sum_{\ell=1}^{N_{\text{dof}}} \big(\delta_{j\ell} - (\boldsymbol{\Pi}_k^{\boldsymbol{\nabla}})_{j\ell}\big)\big(\delta_{i\ell} - (\boldsymbol{\Pi}_k^{\boldsymbol{\nabla}})_{i\ell}\big)
\end{aligned}
$$

i.e.

$$
\mathsf{K}_{\mathsf{s}}^{a} = \bar{\kappa}\, (\mathsf{I} - \boldsymbol{\Pi}_k^{\boldsymbol{\nabla}})^{\mathrm{T}} (\mathsf{I} - \boldsymbol{\Pi}_k^{\boldsymbol{\nabla}}). \tag{36}
$$

If the diffusion κ happens to be a 2×2 symmetric matrix, i.e.

$$
\kappa = \begin{bmatrix} \kappa_{xx} & \kappa_{xy} \\ \kappa_{xy} & \kappa_{yy} \end{bmatrix},
$$

then we can proceed similarly by defining the $n_{k-1} \times n_{k-1}$ matrices $\mathsf{H}^{\kappa_{xx}}$, $\mathsf{H}^{\kappa_{xy}}$ and $\mathsf{H}^{\kappa_{yy}}$ as follows:

$$
(\mathsf{H}^{\kappa_{xx}})_{\alpha\beta} := \int_E \kappa_{xx}\, m_\alpha\, m_\beta \,\mathrm{d}\boldsymbol{x}, \quad (\mathsf{H}^{\kappa_{xy}})_{\alpha\beta} := \int_E \kappa_{xy}\, m_\alpha\, m_\beta \,\mathrm{d}\boldsymbol{x}, \quad \ldots
$$

and the local virtual diffusion consistency matrix $\mathsf{K}_{\mathsf{c}}^{a}$ can be written as

$$
\mathsf{K}_{\mathsf{c}}^{a} = \Big[\big(\overset{*}{\boldsymbol{\Pi}}{}_{k-1}^{0,x}\big)^{\mathrm{T}} \;\; \big(\overset{*}{\boldsymbol{\Pi}}{}_{k-1}^{0,y}\big)^{\mathrm{T}} \Big] \begin{bmatrix} \mathsf{H}^{\kappa_{xx}} & \mathsf{H}^{\kappa_{xy}} \\ \mathsf{H}^{\kappa_{xy}} & \mathsf{H}^{\kappa_{yy}} \end{bmatrix} \begin{bmatrix} \overset{*}{\boldsymbol{\Pi}}{}_{k-1}^{0,x} \\ \overset{*}{\boldsymbol{\Pi}}{}_{k-1}^{0,y} \end{bmatrix}.
$$

In this case, the stability matrix $\mathsf{K}_{\mathsf{s}}^{a}$ can still be taken of the form (36), where this time the constant scalar $\bar{\kappa}$ can be defined as the arithmetic mean of the mean values of κ_{xx} and κ_{yy}. Note that here we are not considering the problem of optimizing the stability matrix with respect to the anisotropy of the diffusion matrix κ, but we are only interested in the convergence as h goes to zero.

6.2 Transport Term

The local VEM approximation for the transport term is

$$
b_h^E(p_h, q_h) := -\int_E \Pi_{k-1}^0 p_h\, (\boldsymbol{b} \cdot \Pi_{k-1}^0 \nabla q_h)\, \mathrm{d}\boldsymbol{x}
$$

and the corresponding local matrix is

$$(\mathsf{K}^{\boldsymbol{b}})_{ij} := b_h^E(\phi_j, \phi_i) = -\int_E \Pi_{k-1}^0 \phi_j \, (\boldsymbol{b} \cdot \Pi_{k-1}^0 \nabla \phi_i) \, \mathrm{d}\boldsymbol{x}.$$

Define the $n_{k-1} \times n_{k-1}$ matrices $\mathsf{H}^{\boldsymbol{b}_x}$ and $\mathsf{H}^{\boldsymbol{b}_y}$ by

$$(\mathsf{H}^{\boldsymbol{b}_x})_{\alpha\beta} := \int_E \boldsymbol{b}_x \, m_\alpha \, m_\beta \, \mathrm{d}\boldsymbol{x}, \quad (\mathsf{H}^{\boldsymbol{b}_y})_{\alpha\beta} := \int_E \boldsymbol{b}_y \, m_\alpha \, m_\beta \, \mathrm{d}\boldsymbol{x}.$$

By (32) we have

$$\begin{aligned} \boldsymbol{b} \cdot [\Pi_{k-1}^0 \nabla \phi_i] &= \boldsymbol{b}_x [\Pi_{k-1}^0 \nabla \phi_{i,x}] + \boldsymbol{b}_y [\Pi_{k-1}^0 \nabla \phi_{i,y}] \\ &= \boldsymbol{b}_x \sum_{\beta=1}^{n_{k-1}} (\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{x}}_{\boldsymbol{k-1}})_{\beta i} \, m_\beta + \boldsymbol{b}_y \sum_{\beta=1}^{n_{k-1}} (\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{y}}_{\boldsymbol{k-1}})_{\beta i} \, m_\beta \end{aligned}$$

so that

$$\begin{aligned} & -\int_E \Pi_{k-1}^0 \phi_j \, (\boldsymbol{b} \cdot \Pi_{k-1}^0 \nabla \phi_i) \, \mathrm{d}\boldsymbol{x} = \\ & -\int_E \left[\sum_{\alpha=1}^{n_{k-1}} (\boldsymbol{\Pi}^{\boldsymbol{0}}_{\boldsymbol{k-1}})_{\alpha j} \, m_\alpha \right] \left[\boldsymbol{b}_x \sum_{\beta=1}^{n_{k-1}} (\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{x}}_{\boldsymbol{k-1}})_{\beta i} \, m_\beta + \boldsymbol{b}_y \sum_{\beta=1}^{n_{k-1}} (\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{y}}_{\boldsymbol{k-1}})_{\beta i} \, m_\beta \right] \mathrm{d}\boldsymbol{x} = \\ & -\int_E \left\{ \boldsymbol{b}_x \sum_{\alpha,\beta=1}^{n_{k-1}} (\boldsymbol{\Pi}^{\boldsymbol{0}}_{\boldsymbol{k-1}})_{\alpha j} \, (\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{x}}_{\boldsymbol{k-1}})_{\beta i} \, m_\beta m_\alpha + \boldsymbol{b}_y \sum_{\alpha,\beta=1}^{n_{k-1}} (\boldsymbol{\Pi}^{\boldsymbol{0}}_{\boldsymbol{k-1}})_{\alpha j} \, (\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{y}}_{\boldsymbol{k-1}})_{\beta i} \, m_\beta m_\alpha \right\} \mathrm{d}\boldsymbol{x} = \\ & -\sum_{\alpha,\beta=1}^{n_{k-1}} (\boldsymbol{\Pi}^{\boldsymbol{0}}_{\boldsymbol{k-1}})_{\alpha j} \, (\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{x}}_{\boldsymbol{k-1}})_{\beta i} \int_E \boldsymbol{b}_x m_\beta m_\alpha \, \mathrm{d}\boldsymbol{x} - \sum_{\alpha,\beta=1}^{n_{k-1}} (\boldsymbol{\Pi}^{\boldsymbol{0}}_{\boldsymbol{k-1}})_{\alpha j} \, (\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{y}}_{\boldsymbol{k-1}})_{\beta i} \int_E \boldsymbol{b}_y m_\beta m_\alpha \, \mathrm{d}\boldsymbol{x} = \\ & -\sum_{\alpha,\beta=1}^{n_{k-1}} (\boldsymbol{\Pi}^{\boldsymbol{0}}_{\boldsymbol{k-1}})_{\alpha j} \, (\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{x}}_{\boldsymbol{k-1}})_{\beta i} \, (\mathsf{H}^{\boldsymbol{b}_x})_{\alpha\beta} - \sum_{\alpha,\beta=1}^{n_{k-1}} (\boldsymbol{\Pi}^{\boldsymbol{0}}_{\boldsymbol{k-1}})_{\alpha j} \, (\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{y}}_{\boldsymbol{k-1}})_{\beta i} \, (\mathsf{H}^{\boldsymbol{b}_y})_{\alpha\beta} = \\ & -\left[(\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{x}}_{\boldsymbol{k-1}})^{\mathrm{T}} \mathsf{H}^{\boldsymbol{b}_x} \boldsymbol{\Pi}^{\boldsymbol{0}}_{\boldsymbol{k-1}} + (\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{y}}_{\boldsymbol{k-1}})^{\mathrm{T}} \mathsf{H}^{\boldsymbol{b}_y} \boldsymbol{\Pi}^{\boldsymbol{0}}_{\boldsymbol{k-1}} \right]_{ij} = \\ & -\left[\left((\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{x}}_{\boldsymbol{k-1}})^{\mathrm{T}} \mathsf{H}^{\boldsymbol{b}_x} + (\overset{*}{\boldsymbol{\Pi}}{}^{\boldsymbol{0},\boldsymbol{y}}_{\boldsymbol{k-1}})^{\mathrm{T}} \mathsf{H}^{\boldsymbol{b}_y} \right) \boldsymbol{\Pi}^{\boldsymbol{0}}_{\boldsymbol{k-1}} \right]_{ij}. \end{aligned}$$

Hence the elementary VEM matrix for the transport term is

$$\mathbf{K}^{\boldsymbol{b}} = -\Big((\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0},\boldsymbol{x}}_{\boldsymbol{k-1}})^{\mathrm{T}}\mathbf{H}^{\boldsymbol{b}_x} + (\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0},\boldsymbol{y}}_{\boldsymbol{k-1}})^{\mathrm{T}}\mathbf{H}^{\boldsymbol{b}_y}\Big)\boldsymbol{\Pi}^{\mathbf{0}}_{\boldsymbol{k-1}}. \tag{37}$$

6.3 Reaction Term

The local VEM approximation for the reaction term is

$$c_h^E(p_h, q_h) := \int_E \gamma\, [\Pi^0_{k-1} p_h]\, [\Pi^0_{k-1} q_h]\, \mathrm{d}\boldsymbol{x}$$

and in matrix form

$$(\mathbf{K}^c)_{ij} := c_h^E(\phi_j, \phi_i) = \int_E \gamma\, [\Pi^0_{k-1}\phi_j]\, [\Pi^0_{k-1}\phi_i]\, \mathrm{d}\boldsymbol{x}.$$

Define the matrix

$$(\mathbf{H}^{\gamma})_{\alpha\beta} := \int_E \gamma\, m_\alpha m_\beta\, \mathrm{d}\boldsymbol{x}$$

and we have immediately

$$(\mathbf{K}^c)_{ij} = \int_E \gamma \Big[\sum_{\alpha=1}^{n_{k-1}} (\boldsymbol{\Pi}^{\mathbf{0}}_{\boldsymbol{k-1}})_{\alpha j}\, m_\alpha\Big]\Big[\sum_{\beta=1}^{n_{k-1}} (\boldsymbol{\Pi}^{\mathbf{0}}_{\boldsymbol{k-1}})_{\beta i}\, m_\beta\Big]\, \mathrm{d}\boldsymbol{x} =$$

$$\sum_{\alpha,\beta=1}^{n_{k-1}} (\boldsymbol{\Pi}^{\mathbf{0}}_{\boldsymbol{k-1}})_{\alpha j} (\boldsymbol{\Pi}^{\mathbf{0}}_{\boldsymbol{k-1}})_{\beta i} \int_E \gamma\, m_\alpha m_\beta\, \mathrm{d}\boldsymbol{x} = \big[(\boldsymbol{\Pi}^{\mathbf{0}}_{\boldsymbol{k-1}})^{\mathrm{T}}\mathbf{H}^{\boldsymbol{\gamma}}\,\boldsymbol{\Pi}^{\mathbf{0}}_{\boldsymbol{k-1}}\big]_{ij}$$

i.e.

$$\mathbf{K}^c = (\boldsymbol{\Pi}^{\mathbf{0}}_{\boldsymbol{k-1}})^{\mathrm{T}}\mathbf{H}^{\boldsymbol{\gamma}}\,\boldsymbol{\Pi}^{\mathbf{0}}_{\boldsymbol{k-1}}. \tag{38}$$

7 Algorithm for the Primal Formulation

For the convenience of the reader, we summarize the results of the previous Section in form of an algorithm ready to be implemented.

7.1 Projectors

1. Compute the $n_k \times N_{\mathrm{dof}}$ matrix $\mathbf{B}$ given by

$$\mathbf{B} = \begin{bmatrix} \mathrm{P}_0\phi_1 & \dots & \mathrm{P}_0\phi_{N_{\mathrm{dof}}} \\ (\nabla m_2, \nabla\phi_1)_{0,E} & \dots & (\nabla m_2, \nabla\phi_{N_{\mathrm{dof}}})_{0,E} \\ \vdots & \ddots & \vdots \\ (\nabla m_{n_k}, \nabla\phi_1)_{0,E} & \dots & (\nabla m_{n_k}, \nabla\phi_{N_{\mathrm{dof}}})_{0,E} \end{bmatrix},$$

   where the terms of type $(\nabla m_\alpha, \nabla\phi_i)_{0,E}$ can be determined as shown in Lemma 1.
2. Compute the $N_{\mathrm{dof}} \times n_k$ matrix $\mathbf{D}$ defined by:

$$\mathbf{D}_{i\alpha} = \mathrm{dof}_i(m_\alpha), \quad i = 1, \dots, N_{\mathrm{dof}},\ \alpha = 1, \dots, n_k.$$

3. Set

$$\mathbf{G} = \mathbf{B}\mathbf{D}. \tag{39}$$

   Note that the $n_k \times n_k$ matrix $\mathbf{G}$ can be computed independently (see (18)), and (39) can be used as a check of the correctness of the code.
4. Set

$$\overset{*}{\mathbf{\Pi}}{}^{\nabla}_{k} = \mathbf{G}^{-1}\mathbf{B} \quad \text{and} \quad \mathbf{\Pi}^{0}_{k} = \mathbf{D}\overset{*}{\mathbf{\Pi}}{}^{0}_{k}.$$

5. Compute the $n_k \times n_k$ matrix $\mathbf{H}$ defined by:

$$\mathbf{H}_{\alpha\beta} = \int_E m_\alpha m_\beta \,\mathrm{d}\boldsymbol{x} \quad \alpha, \beta = 1, \dots, n_k.$$

6. Compute the $n_k \times N_{\mathrm{dof}}$ matrix $\mathbf{C}$ defined by

$$\mathbf{C}_{\alpha i} = \int_E m_\alpha\, \phi_i \,\mathrm{d}\boldsymbol{x}, \quad \alpha = 1, \dots, n_k,\ i = 1, \dots, N_{\mathrm{dof}}.$$

   The matrix $\mathbf{C}$ has the following structure:

   • first n_{k-2} lines of $\mathbf{C} = |E| \left[\begin{array}{ccccc:cccc} 0 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 1 \end{array}\right]$ where the last block is the identity matrix of size $n_{k-2} \times n_{k-2}$;
   • last $n_k - n_{k-2}$ lines of $\mathbf{C}$:

$$\mathbf{C}_{\alpha i} = (\mathbf{H}\,\overset{*}{\mathbf{\Pi}}{}^{\nabla}_{k})_{\alpha i}, \quad n_{k-2} < \alpha \leqslant n_k.$$

7. Set

$$\overset{*}{\mathbf{\Pi}}{}^{\mathbf{0}}_{k} = \mathbf{H}^{-1}\mathbf{C} \quad \text{and} \quad \mathbf{\Pi}^{\mathbf{0}}_{k} = \mathbf{D}\overset{*}{\mathbf{\Pi}}{}^{\mathbf{0}}_{k}.$$

8. Compute the $N_{\text{dof}} \times n_{k-1}$ matrices $\mathbf{E}^{x}$ and $\mathbf{E}^{y}$ (see (33) and (34)) by

$$\left(\mathbf{E}^{x}\right)_{i\beta} = \int_E \phi_{i,x}\, m_\beta \,\mathrm{d}\boldsymbol{x}, \quad \left(\mathbf{E}^{y}\right)_{i\beta} = \int_E \phi_{i,y}\, m_\beta \,\mathrm{d}\boldsymbol{x}.$$

9. Set

$$\overset{*}{\mathbf{\Pi}}{}^{\mathbf{0},x}_{k-1} = \hat{\mathbf{H}}^{-1}\mathbf{E}^{x}, \quad \overset{*}{\mathbf{\Pi}}{}^{\mathbf{0},y}_{k-1} = \hat{\mathbf{H}}^{-1}\mathbf{E}^{y}$$

where $\hat{\mathbf{H}}$ is the submatrix of $\mathbf{H}$ obtained by taking the first n_{k-1} rows and columns of $\mathbf{H}$.

7.2 *Coefficient Matrices*

Compute the $n_{k-1} \times n_{k-1}$ matrices

$$(\mathbf{H}^{\kappa})_{\alpha\beta} = \int_E \kappa\, m_\alpha\, m_\beta \,\mathrm{d}\boldsymbol{x}, \tag{40}$$

$$(\mathbf{H}^{b_x})_{\alpha\beta} = \int_E \boldsymbol{b}_x\, m_\alpha\, m_\beta \,\mathrm{d}\boldsymbol{x}, \quad (\mathbf{H}^{b_y})_{\alpha\beta} = \int_E \boldsymbol{b}_y\, m_\alpha\, m_\beta \,\mathrm{d}\boldsymbol{x}, \tag{41}$$

$$(\mathbf{H}^{\gamma})_{\alpha\beta} = \int_E \gamma\, m_\alpha m_\beta \,\mathrm{d}\boldsymbol{x}. \tag{42}$$

7.3 *Local Stiffness Matrices*

Finally, set

$$\begin{aligned}
\mathbf{K}^{a} &= \left[\left(\overset{*}{\mathbf{\Pi}}{}^{\mathbf{0},x}_{k-1}\right)^{\mathrm{T}} \ \left(\overset{*}{\mathbf{\Pi}}{}^{\mathbf{0},y}_{k-1}\right)^{\mathrm{T}}\right] \begin{bmatrix} \mathbf{H}^{\kappa} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}^{\kappa} \end{bmatrix} \begin{bmatrix} \overset{*}{\mathbf{\Pi}}{}^{\mathbf{0},x}_{k-1} \\ \overset{*}{\mathbf{\Pi}}{}^{\mathbf{0},y}_{k-1} \end{bmatrix} + \bar{\kappa}\,(\mathbf{I} - \mathbf{\Pi}^{\nabla}_{k})^{\mathrm{T}}(\mathbf{I} - \mathbf{\Pi}^{\nabla}_{k}) \\
\mathbf{K}^{b} &= -\Big((\overset{*}{\mathbf{\Pi}}{}^{\mathbf{0},x}_{k-1})^{\mathrm{T}}\mathbf{H}^{b_x} + (\overset{*}{\mathbf{\Pi}}{}^{\mathbf{0},y}_{k-1})^{\mathrm{T}}\mathbf{H}^{b_y}\Big)\, \mathbf{\Pi}^{\mathbf{0}}_{k-1} \\
\mathbf{K}^{c} &= (\mathbf{\Pi}^{\mathbf{0}}_{k-1})^{\mathrm{T}}\mathbf{H}^{\gamma}\,\mathbf{\Pi}^{\mathbf{0}}_{k-1}.
\end{aligned}$$

8 Virtual Element Spaces for the Mixed Formulation

Before defining the virtual space $\boldsymbol{V}_h(E)$, we need to study certain spaces of polynomials which will play a major role in the definition of the degrees of freedom.

We start by defining an easily computable basis $\{\boldsymbol{m}_I\}$ for $[\mathcal{P}_k(E)]^2$. Let I be an index running from 1 to $2\,n_k = \dim[\mathcal{P}_k(E)]^2$. Set:

$$\begin{cases} \boldsymbol{m}_I := \begin{bmatrix} m_I \\ 0 \end{bmatrix} & \text{if } 1 \leqslant I \leqslant n_k \\ \boldsymbol{m}_I := \begin{bmatrix} 0 \\ m_{I-n_k} \end{bmatrix} & \text{if } n_k + 1 \leqslant I \leqslant 2n_k. \end{cases}$$

We introduce the (vector) polynomial spaces

$$\boldsymbol{\mathcal{G}}_k^{\nabla}(E) := \nabla \mathcal{P}_{k+1}(E)$$

and

$$\boldsymbol{\mathcal{G}}_k^{\perp}(E) := L^2\text{-orthogonal complement of } \boldsymbol{\mathcal{G}}_k^{\nabla}(E) \text{ in } [\mathcal{P}_k(E)]^2$$

or, more generally,

$$\boldsymbol{\mathcal{G}}_k^{\oplus}(E) := \text{any complement of } \boldsymbol{\mathcal{G}}_k^{\nabla}(E) \text{ in } [\mathcal{P}_k(E)]^2.$$

An easy computation shows that

$$\dim \boldsymbol{\mathcal{G}}_k^{\nabla}(E) = n_k^{\nabla} := n_k + (k+1) \quad \text{and} \quad \dim \boldsymbol{\mathcal{G}}_k^{\oplus}(E) = n_k^{\oplus} := n_k - (k+1).$$

We construct now a basis for $\boldsymbol{\mathcal{G}}_k^{\nabla}(E)$ and $\boldsymbol{\mathcal{G}}_k^{\oplus}(E)$. It is easy to check that a basis for $\boldsymbol{\mathcal{G}}_k^{\nabla}(E)$ is given by

$$\boldsymbol{g}_\alpha^{\nabla,k} := \nabla m_{\alpha+1}, \quad \alpha = 1, \ldots, n_k^{\nabla}.$$

Let now the $n_k^{\nabla} \times 2n_k$ matrix T^{∇} be such that

$$\boldsymbol{g}_\alpha^{\nabla,k} = \sum_{I=1}^{2n_k} \mathsf{T}_{\alpha I}^{\nabla} \boldsymbol{m}_I, \quad \alpha = 1, \ldots, n_k^{\nabla}.$$

A way to obtain a basis in $\boldsymbol{\mathcal{G}}_k^{\oplus}(E)$ is to complete the matrix T^{∇} with a $n_k^{\oplus} \times 2n_k$ matrix $\mathsf{T}^{\oplus}$ to form a non-singular $(n_k^{\nabla} + n_k^{\oplus} = 2n_k) \times 2n_k$ square matrix $\mathsf{T} = \begin{bmatrix} \mathsf{T}^{\nabla} \\ \mathsf{T}^{\oplus} \end{bmatrix}$.

A basis for $\mathcal{G}_k^{\oplus}(E)$ is then given by

$$\boldsymbol{g}_\gamma^{\oplus,k} := \sum_{I=1}^{2n_k} \mathsf{T}^{\oplus}_{\gamma I} \boldsymbol{m}_I, \quad \gamma = 1, \dots, n_k^{\oplus}.$$

An obvious way of constructing the matrix T is to define the rows of $\mathsf{T}^{\oplus}$ as a basis for the kernel of T^{∇}. This can be easily done symbolically in MATLAB:

```
TO = null(TN)'; T = [TN; TO]; go = T*m;
```

where TN $= \mathsf{T}^{\nabla}$ and TO $= \mathsf{T}^{\oplus}$. In the appendix we present the basis so obtained up to $k = 5$.

8.1 The Space $V_h(E)$

We are ready now to define the local VEM space $\boldsymbol{V}_h(E)$ which consists of functions $\boldsymbol{v}_h$ such that:

- $\boldsymbol{v}_h \in \boldsymbol{H}(\mathrm{div}; E) \cap \boldsymbol{H}(\mathrm{rot}; E)$;
- $\boldsymbol{v}_h \cdot \boldsymbol{n}_e$ is a polynomial of degree k on each edge e;
- $\mathrm{div}\, \boldsymbol{v}_h \in \mathcal{P}_k(E)$;
- $\mathrm{rot}\, \boldsymbol{v}_h \in \mathcal{P}_{k-1}(E)$.

In [11] we have shown the following results:

1. the dimension of $\boldsymbol{V}_h(E)$ on a polygon E is

$$\begin{aligned} N_{\mathrm{dof}} := \dim \boldsymbol{V}_h(E) &= N_{\boldsymbol{e}} \times (k+1) + \dim \mathcal{G}_{k-1}^{\nabla}(E) + \dim \mathcal{G}_k^{\oplus}(E) \\ &= N_{\boldsymbol{e}} \times (k+1) + n_{k-1}^{\nabla} + n_k^{\oplus} = N_{\boldsymbol{e}} \times (k+1) + 2n_k - k - 2 \end{aligned}$$

2. $[\mathcal{P}_k(E)]^2 \subset \boldsymbol{V}_h(E)$;
3. for the space $\boldsymbol{V}_h(E)$ we can take the following degrees of freedom:

   - **Edge dofs [$N_e \times (k+1)$]**

     Since on each edge $\boldsymbol{v}_h \cdot \boldsymbol{n}_e$ is a polynomial of degree k and no continuity is enforced at the vertices, we need to identify a polynomial of degree k on each edge without using the values at the vertices.

     This can be done in several ways, the most natural being taking the value of $\boldsymbol{v}_h \cdot \boldsymbol{n}_e$ at $k+1$ internal distinct $\{\boldsymbol{x}_\ell^e\}$ points of the edge e, obtained by subdividing e in $k+2$ equal parts:

$$\mathrm{dof}_\ell^e(\boldsymbol{v}_h) := (\boldsymbol{v}_h \cdot \boldsymbol{n}_e)(\boldsymbol{x}_\ell^e), \quad \ell = 1, \dots, k+1.$$

This choice automatically ensures the continuity of $\boldsymbol{v}_h \cdot \boldsymbol{n}_e$ across two adjacent elements.

- **Internal ∇ dofs [$n^{\nabla}_{k-1} = n_k - 1$]**
  Let α be an index running from 1 to $\dim \boldsymbol{\mathcal{G}}^{\nabla}_{k-1}(E) = n^{\nabla}_{k-1}$. We define:

$$\mathrm{dof}^{\nabla}_{\alpha}(\boldsymbol{v}_h) := \frac{1}{|E|}\int_E \boldsymbol{v}_h \cdot \boldsymbol{g}^{\nabla,k-1}_{\alpha}\,\mathrm{d}\boldsymbol{x}, \qquad \boldsymbol{g}^{\nabla,k-1}_{\alpha} \in \boldsymbol{\mathcal{G}}^{\nabla}_{k-1}(E).$$

- **Internal $\oplus$ dofs [$n^{\oplus}_{k} = n_k - (k+1)$]**
  Let γ be an index running from 1 to $\dim \boldsymbol{\mathcal{G}}^{\oplus}_{k}(E) = n^{\oplus}_{k}$. We define:

$$\mathrm{dof}^{\oplus}_{\gamma}(\boldsymbol{v}_h) := \frac{1}{|E|}\int_E \boldsymbol{v}_h \cdot \boldsymbol{g}^{\oplus,k}_{\gamma}\,\mathrm{d}\boldsymbol{x}, \qquad \boldsymbol{g}^{\oplus,k}_{\gamma} \in \boldsymbol{\mathcal{G}}^{\oplus}_{k}(E).$$

Let i be an index running through all dofs. We define $\boldsymbol{\phi}_i \in \boldsymbol{V}_h(E)$ by

$$\mathrm{dof}_j(\boldsymbol{\phi}_i) = \delta_{ij}, \quad j = 1, \ldots, N_{\mathrm{dof}}$$

in such a way that we have again a Lagrange-type identity:

$$\boldsymbol{v}_h = \sum_{i=1}^{N_{\mathrm{dof}}} \mathrm{dof}_i(\boldsymbol{v}_h)\,\boldsymbol{\phi}_i.$$

8.2 *The Space $Q_h(E)$*

As promised, the space $Q_h(E)$ is simply the space $\mathcal{P}_k(E)$ and as basis functions we take the set of scaled monomials $\mathcal{M}_k(E)$ defined in (2).

9 VEM Approximation of the Mixed Formulation

As show in [11], the VEM approximation of problem (9) is

$$\begin{cases} \text{Find } (\boldsymbol{u}_h, p_h) \in \boldsymbol{V}_h \times Q_h \text{ such that} \\ \displaystyle\sum_E \left\{a^E_h(\boldsymbol{u}_h, \boldsymbol{v}_h) - (p_h, \operatorname{div} \boldsymbol{v}_h)_{0,E} - (\boldsymbol{\beta}\cdot \Pi^0_k \boldsymbol{v}_h, p_h)_{0,E}\right\} = 0 \qquad \text{for all } \boldsymbol{v}_h \in \boldsymbol{V}_h, \\ \displaystyle\sum_E (\operatorname{div} \boldsymbol{u}_h, q_h)_{0,E} + (\gamma p_h, q_h)_{0,\Omega} = (f, q_h)_{0,\Omega} \qquad \text{for all } q_h \in Q_h \end{cases}$$

where

$$a_h^E(\boldsymbol{u}_h, \boldsymbol{v}_h) := (\nu\, \Pi_k^0 \boldsymbol{u}_h, \Pi_k^0 \boldsymbol{v}_h)_{0,E} + \mathcal{S}_E\big((I - \Pi_k^0)\boldsymbol{u}_h, (I - \Pi_k^0)\boldsymbol{v}_h\big).$$

The symmetric and positive bilinear form $\mathcal{S}_E(\cdot,\cdot)$, needed for the stability of the method, is defined by requiring

$$\mathcal{S}_E(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) = \overline{\nu}|E|\,\delta_{ij},$$

with $\overline{\nu}$ = mean value of ν on E, or $\overline{\nu} = \nu(x_c, y_c)$. The corresponding local stiffness matrices are obtained by restricting all integrals to E and by setting $\boldsymbol{u}_h = \boldsymbol{\phi}_j$, $\boldsymbol{v}_h = \boldsymbol{\phi}_i$, $p_h = m_\alpha$, $q_h = m_\beta$.

9.1 Computation of the L^2-projection in $V_h(E)$

Let $\boldsymbol{\phi}_i$ be a basis function for $\boldsymbol{V}_h(E)$. We need to compute $\Pi_k^0 \boldsymbol{\phi}_i \in [\mathcal{P}_k(E)]^2$. We shall write $\Pi_k^0 \boldsymbol{\phi}_i$ in terms of the basis $\{\boldsymbol{g}_I^k\} = \{\boldsymbol{g}_\alpha^{\nabla,k}, \boldsymbol{g}_\gamma^{\oplus,k}\}$ of $[\mathcal{P}_k(E)]^2$:

$$\Pi_k^0 \boldsymbol{\phi}_i = \sum_{\alpha=1}^{n_k^\nabla} s_i^\alpha\, \boldsymbol{g}_\alpha^{\nabla,k} + \sum_{\gamma=1}^{n_k^\oplus} s_i^\gamma\, \boldsymbol{g}_\gamma^{\oplus,k} = \sum_{I=1}^{2n_k} s_i^I \boldsymbol{g}_I^k. \tag{43}$$

Multiplying by $\{\boldsymbol{g}_\beta^{\nabla,k}, \boldsymbol{g}_\gamma^{\oplus,k}\}$ and integrating, we get a linear system in the unknowns $\{s_i^\alpha, s_i^\gamma\} = s_i^I$ (note that $\int_E \Pi_k^0 \boldsymbol{\phi}_i \cdot \boldsymbol{p}_k \,\mathrm{d}\boldsymbol{x} = \int_E \boldsymbol{\phi}_i \cdot \boldsymbol{p}_k \,\mathrm{d}\boldsymbol{x}$):

$$\begin{cases} \displaystyle\sum_{\alpha=1}^{n_k^\nabla} s_i^\alpha \int_E \boldsymbol{g}_\alpha^{\nabla,k} \cdot \boldsymbol{g}_\beta^{\nabla,k} \,\mathrm{d}\boldsymbol{x} + \sum_{\gamma=1}^{n_k^\oplus} s_i^\gamma \int_E \boldsymbol{g}_\gamma^{\oplus,k} \cdot \boldsymbol{g}_\beta^{\nabla,k} \,\mathrm{d}\boldsymbol{x} = \int_E \boldsymbol{\phi}_i \cdot \boldsymbol{g}_\beta^{\nabla,k} \,\mathrm{d}\boldsymbol{x} \\ \displaystyle\sum_{\alpha=1}^{n_k^\nabla} s_i^\alpha \int_E \boldsymbol{g}_\alpha^{\nabla,k} \cdot \boldsymbol{g}_\delta^{\oplus,k} \,\mathrm{d}\boldsymbol{x} + \sum_{\gamma=1}^{n_k^\oplus} s_i^\gamma \int_E \boldsymbol{g}_\gamma^{\oplus,k} \cdot \boldsymbol{g}_\delta^{\oplus,k} \,\mathrm{d}\boldsymbol{x} = \int_E \boldsymbol{\phi}_i \cdot \boldsymbol{g}_\delta^{\oplus,k} \,\mathrm{d}\boldsymbol{x}. \end{cases}$$

Set

$$\mathbf{G}_{IJ} := \int_E \boldsymbol{g}_I^k \cdot \boldsymbol{g}_J^k \,\mathrm{d}\boldsymbol{x},$$

and define the $2n_k \times N_{\mathrm{dof}}$ matrices

$$[\overset{*}{\boldsymbol{\Pi}}{}_k^{\mathbf{0}}]_{Ii} := s_i^I \quad \text{and} \quad \mathbf{B}_{Ii} := \int_E \boldsymbol{\phi}_i \cdot \boldsymbol{g}_I^k \,\mathrm{d}\boldsymbol{x}. \tag{44}$$

We have

$$\sum_{J=1}^{2n_k} \mathbf{G}_{IJ}\,[\overset{*}{\mathbf{\Pi}}{}^{\mathbf{0}}_{\boldsymbol{k}}]_{Ji} = \mathbf{B}_{Ii} \quad \text{i.e.} \quad \mathbf{G}\,\overset{*}{\mathbf{\Pi}}{}^{\mathbf{0}}_{\boldsymbol{k}} = \mathbf{B} \quad \text{so that} \quad \overset{*}{\mathbf{\Pi}}{}^{\mathbf{0}}_{\boldsymbol{k}} = \mathbf{G}^{-1}\,\mathbf{B}.$$

We split $\mathbf{B}$ as

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}^{\boldsymbol{\nabla}} \\ \mathbf{B}^{\oplus} \end{bmatrix}.$$

We start from $\mathbf{B}^{\boldsymbol{\nabla}}_{\beta i} = \int_E \boldsymbol{\phi}_i \cdot \boldsymbol{g}_\beta^{\nabla,k}\,\mathrm{d}\boldsymbol{x}$. Since

$$\boldsymbol{g}_\beta^{\nabla,k} = \nabla m_{\beta+1},$$

we have

$$\mathbf{B}^{\boldsymbol{\nabla}}_{\beta i} = \int_E \boldsymbol{\phi}_i \cdot \nabla m_{\beta+1}\,\mathrm{d}\boldsymbol{x} = -\int_E \operatorname{div}\boldsymbol{\phi}_i\, m_{\beta+1}\,\mathrm{d}\boldsymbol{x} + \int_{\partial E} \boldsymbol{\phi}_i \cdot \boldsymbol{n}_E\, m_{\beta+1}\,\mathrm{d}s$$
$$=: \mathbf{B}^{\boldsymbol{\nabla}}_1 + \mathbf{B}^{\boldsymbol{\nabla}}_2.$$

The term $\mathbf{B}^{\boldsymbol{\nabla}}_2$ can be readily computed because $\boldsymbol{\phi}_i \cdot \boldsymbol{n}$ is a known polynomial on the boundary of E. Concerning the term $\mathbf{B}^{\boldsymbol{\nabla}}_1$, we first observe that we can directly compute $\operatorname{div}\boldsymbol{\phi}_i \in \mathcal{P}_k(E)$. In fact, write $\operatorname{div}\boldsymbol{\phi}_i$ as

$$\operatorname{div}\boldsymbol{\phi}_i = \sum_{\sigma=1}^{n_k} d_i^\sigma\, m_\sigma,$$

multiply by m_τ and integrate over E:

$$\sum_{\sigma=1}^{n_k} d_i^\sigma \int_E m_\sigma m_\tau\,\mathrm{d}\boldsymbol{x} = \int_E \operatorname{div}\boldsymbol{\phi}_i\, m_\tau\,\mathrm{d}\boldsymbol{x}.$$

Define the $n_k \times n_k$ matrix $\mathbf{H}$ (as already done in (26)) by

$$\mathbf{H}_{\sigma\tau} := \int_E m_\sigma m_\tau\,\mathrm{d}\boldsymbol{x},$$

and the $n_k \times N_{\mathrm{dof}}$ matrices $\mathbf{V}$ and $\mathbf{W}$ as

$$\mathbf{V}_{\sigma i}: = d_i^\sigma, \quad \mathbf{W}_{\tau i} := \int_E \operatorname{div}\boldsymbol{\phi}_i\, m_\tau\,\mathrm{d}\boldsymbol{x} \tag{45}$$

so that

$$\mathbf{H}\mathbf{V} = \mathbf{W} \quad \text{and} \quad \mathbf{V} = \mathbf{H}^{-1}\mathbf{W}.$$

Now,

$$\begin{aligned}\mathbf{W}_{\tau i} = \int_E \operatorname{div} \boldsymbol{\phi}_i \, m_\tau \, \mathrm{d}\boldsymbol{x} &= -\int_E \boldsymbol{\phi}_i \cdot \nabla m_\tau \, \mathrm{d}\boldsymbol{x} + \int_{\partial E} \boldsymbol{\phi}_i \cdot \boldsymbol{n}_E \, m_\tau \, \mathrm{d}s \\ &=: [\mathbf{W_1}]_{\tau i} + [\mathbf{W_2}]_{\tau i}.\end{aligned}$$

Observing that

$$\nabla m_\tau = \boldsymbol{g}_{\tau-1}^{\nabla,k-1},$$

we have

$$[\mathbf{W_1}]_{\tau i} = -|E| \, \mathrm{dof}_{\tau-1}^{\boldsymbol{g}}(\boldsymbol{\phi}_i) = \begin{cases} -|E| & \text{if } i \text{ corresponds to } \tau - 1 \\ 0 & \text{otherwise.} \end{cases} \tag{46}$$

Concerning the term $\mathbf{W_2}$, we observe that it can be immediately computed since $\boldsymbol{\phi}_i \cdot \boldsymbol{n}_E$ is a known polynomial on the boundary. Consider now $\mathbf{B_1^\nabla}$:

$$[\mathbf{B_1^\nabla}]_{\beta i} = -\int_E \operatorname{div} \boldsymbol{\phi}_i \, m_{\beta+1} \, \mathrm{d}\boldsymbol{x} = -\sum_{\sigma=1}^{n_k} d_i^\sigma \int_E m_\sigma \, m_{\beta+1} \, \mathrm{d}\boldsymbol{x}.$$

Define the $n_k^\nabla \times n_k$ matrix

$$\mathbf{H}^\#_{\beta\sigma} := \int_E m_\sigma \, m_{\beta+1} \, \mathrm{d}\boldsymbol{x}.$$

Obviously, most of the entries of the matrix $\mathbf{H}^\#$ are also entries of the matrix $\mathbf{H}$ already computed. Then

$$-\int_E \operatorname{div} \boldsymbol{\phi}_i \, m_{\beta+1} \, \mathrm{d}\boldsymbol{x} = -[\mathbf{H}^\#\mathbf{V}]_{\beta i} = -[\mathbf{H}^\#\mathbf{H}^{-1}\mathbf{W}]_{\beta i}$$

so that

$$\mathbf{B}_1^\nabla = -\mathbf{H}^\#\mathbf{H}^{-1}(\mathbf{W_1} + \mathbf{W_2}).$$

Concerning the term $\mathbf{B}^\oplus$, we simply observe that

$$\mathbf{B}^\oplus_{\delta i} = \int_E \boldsymbol{\phi}_i \cdot \boldsymbol{g}_\delta^{\oplus,k} = |E| \, \mathrm{dof}_\delta^\oplus(\boldsymbol{\phi}_i) = \begin{cases} |E| & \text{if } \delta \text{ corresponds to } i \\ 0 & \text{otherwise.} \end{cases}$$

We will also need $\Pi_k^0 \boldsymbol{\phi}_i$ in terms of the basis $\{\boldsymbol{\phi}_i\}$ itself. To this end, we define π_i^j as

$$\Pi_k^0 \boldsymbol{\phi}_i = \sum_{j=1}^{N_{\mathrm{dof}}} \pi_i^j \boldsymbol{\phi}_j \quad \text{or } \pi_i^j := \mathrm{dof}_j(\Pi_k^0 \boldsymbol{\phi}_i) \tag{47}$$

and the $N_{\mathrm{dof}} \times N_{\mathrm{dof}}$ matrix $\boldsymbol{\Pi}_{\boldsymbol{k}}^{\mathbf{0}}$ as

$$[\boldsymbol{\Pi}_{\boldsymbol{k}}^{\mathbf{0}}]_{ji} := \pi_i^j.$$

From (43) we have

$$\Pi_k^0 \boldsymbol{\phi}_i = \sum_{I=1}^{2n_k} s_i^I \boldsymbol{g}_I^k = \sum_{I=1}^{2n_k} s_i^I \left[\sum_{j=1}^{N_{\mathrm{dof}}} \mathrm{dof}_j(\boldsymbol{g}_I^k) \boldsymbol{\phi}_j \right] = \sum_{j=1}^{N_{\mathrm{dof}}} \left[\sum_{I=1}^{2n_k} s_i^I \mathrm{dof}_j(\boldsymbol{g}_I^k) \right] \boldsymbol{\phi}_j,$$

and comparing with (47) we obtain

$$\pi_i^j = \sum_{I=1}^{2n_k} s_i^I \mathrm{dof}_j(\boldsymbol{g}_I^k).$$

If we define the $N_{\mathrm{dof}} \times 2n_k$ matrix

$$\mathsf{D}_{jI} := \mathrm{dof}_j(\boldsymbol{g}_I^k)$$

we have:

$$\boldsymbol{\Pi}_{\boldsymbol{k}}^{\mathbf{0}} = \mathsf{D} \overset{*}{\boldsymbol{\Pi}}{}_{\boldsymbol{k}}^{\mathbf{0}} \quad \text{i.e.} \quad \boldsymbol{\Pi}_{\boldsymbol{k}}^{\mathbf{0}} = \mathsf{D}\mathsf{G}^{-1}\mathsf{B}.$$

We observe that

$$\mathsf{G}_{IJ} = \int_E \boldsymbol{g}_I^k \cdot \boldsymbol{g}_J^k \,\mathrm{d}\boldsymbol{x}, \quad \text{and} \quad \boldsymbol{g}_J^k = \sum_{i=1}^{N_{\mathrm{dof}}} \mathrm{dof}_i(\boldsymbol{g}_J^k) \boldsymbol{\phi}_i$$

so that

$$\mathsf{G}_{IJ} = \sum_{i=1}^{N_{\mathrm{dof}}} \mathrm{dof}_i(\boldsymbol{g}_J^k) \int_E \boldsymbol{g}_I^k \cdot \boldsymbol{\phi}_i \,\mathrm{d}\boldsymbol{x} = \sum_{i=1}^{N_{\mathrm{dof}}} \mathsf{D}_{iJ} \mathsf{B}_{Ii} \quad \text{hence} \quad \mathsf{G} = \mathsf{B}\mathsf{D}. \tag{48}$$

We have the following useful identities:

$$\overset{*}{\boldsymbol{\Pi}}{}_{\boldsymbol{k}}^{\mathbf{0}} \mathsf{D} = \mathsf{I} \quad \text{since} \quad \overset{*}{\boldsymbol{\Pi}}{}_{\boldsymbol{k}}^{\mathbf{0}} \mathsf{D} = \mathsf{G}^{-1}\mathsf{B}\mathsf{D} = \mathsf{G}^{-1}\mathsf{G} = \mathsf{I}$$

and

$$\mathbf{\Pi}_k^0 \mathbf{D} = \mathbf{D} \quad \text{since} \quad \mathbf{\Pi}_k^0 \mathbf{D} = \mathbf{D} \overset{*}{\mathbf{\Pi}}{}_k^0 \mathbf{D} = \mathbf{D}\mathbf{I} = \mathbf{D}.$$

Another way of arguing is that since Π_k^0 is a projection, then $(\mathbf{\Pi}_k^0)^2 = \mathbf{\Pi}_k^0$. Hence

$$(\mathbf{\Pi}_k^0)^2 = \mathbf{D}\mathbf{G}^{-1}\mathbf{B}\mathbf{D}\mathbf{G}^{-1}\mathbf{B} = \mathbf{D}[\mathbf{G}^{-1}\mathbf{B}\mathbf{D}]\mathbf{G}^{-1}\mathbf{B} = \mathbf{\Pi}_k^0 = \mathbf{D}\mathbf{G}^{-1}\mathbf{B}$$

hence $\mathbf{G}^{-1}\mathbf{B}\mathbf{D}$ must be the identity matrix as stated in (48).

Remark 2 It can be shown that the lower part of the matrix $\mathbf{\Pi}_k^0$ corresponding to the internal dofs (last $n_{k-1}^{\nabla} + n_k^{\oplus}$ rows) is the identity matrix. This property can be exploited in the definition of the stability matrix (50) described below (see [11]).

10 Local Matrices

We are now ready to compute the VEM local matrices for the mixed formulation.

10.1 Term $a_h^E(u_h, v_h)$

The corresponding local matrix is given by

$$\begin{aligned} a_h^E(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) &= (\nu\, \Pi_k^0 \boldsymbol{\phi}_j, \Pi_k^0 \boldsymbol{\phi}_i)_{0,E} + \mathcal{S}_E\big((I - \Pi_k^0)\boldsymbol{\phi}_j, (I - \Pi_k^0)\boldsymbol{\phi}_i\big) \\ &:= (\mathbf{K}_{\mathbf{c}}^{a})_{ij} + (\mathbf{K}_{\mathbf{s}}^{a})_{ij}. \end{aligned}$$

Using (43), the *consistency* matrix $\mathbf{K}_{\mathbf{c}}^{a}$ is given by

$$[\mathbf{K}_{\mathbf{c}}^{a}]_{ij} = \sum_{I=1}^{2n_k} \sum_{J=1}^{2n_k} s_i^I s_j^J \int_E \nu\, \boldsymbol{g}_I^k \cdot \boldsymbol{g}_J^k \, \mathrm{d}\boldsymbol{x}.$$

Defining the $2n_k \times 2n_k$ matrix $\mathbf{G}^{\nu}$

$$\mathbf{G}_{IJ}^{\nu} := \int_E \nu\, \boldsymbol{g}_I^k \cdot \boldsymbol{g}_J^k \, \mathrm{d}\boldsymbol{x},$$

and using (44) we obtain:

$$[\mathbf{K}_{\mathbf{c}}^{a}]_{ij} = \sum_{I=1}^{2n_k} \sum_{J=1}^{2n_k} [\overset{*}{\mathbf{\Pi}}{}_k^0]_{Ii} [\overset{*}{\mathbf{\Pi}}{}_k^0]_{Jj} \mathbf{G}_{IJ}^{\nu}$$

i.e.

$$\mathbf{K}^a_{\mathbf{c}} = [\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0}}_k]^{\mathrm{T}} \mathbf{G}^{\nu} \overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0}}_k. \tag{49}$$

If $\boldsymbol{\nu}(x) \equiv 1$, i.e. we have the Laplace operator, then $\mathbf{G}^{\nu} = \mathbf{G}$ and

$$\mathbf{K}^a_{\mathbf{c}} = [\mathbf{G}^{-1}\mathbf{B}]^{\mathrm{T}}\, \mathbf{G}\, [\mathbf{G}^{-1}\mathbf{B}] = \mathbf{B}^{\mathrm{T}}\mathbf{G}^{-1}\mathbf{B}.$$

The *stability* matrix $\mathbf{K}^a_{\mathbf{s}}$ can be taken as

$$\mathbf{K}^a_{\mathbf{s}} = \bar{\nu}\, |E|\, (\mathbf{I} - \boldsymbol{\Pi}^{\mathbf{0}}_k)^{\mathrm{T}}\, (\mathbf{I} - \boldsymbol{\Pi}^{\mathbf{0}}_k) \tag{50}$$

where $\bar{\nu}$ is a constant approximation of ν.

10.2 Term $-(p_h, div\ v_h)_{0,E}$

By (45) we see that the corresponding local matrix is $-\mathbf{W}^{\mathrm{T}}$ which has already been computed.

The local matrix $\mathbf{K}$ corresponding to $\boldsymbol{\beta} = (0, 0)$ and $\gamma = 0$ is then given by:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}^a_{\mathbf{c}} + \mathbf{K}^a_{\mathbf{s}} & -\mathbf{W}^{\mathrm{T}} \\ \mathbf{W} & 0 \end{bmatrix}$$

10.3 Term $-(\boldsymbol{\beta} \cdot \Pi^{0}_k v_h, p_h)_{0,E}$

The corresponding local matrix is

$$\mathbf{T}^{\boldsymbol{\beta}}_{j\sigma} := -\int_E [\boldsymbol{\beta} \cdot \Pi^0_k \boldsymbol{\phi}_j]\, m^k_\sigma \,\mathrm{d}\boldsymbol{x} = -\sum_{I=1}^{2n_k} [\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0}}_k]_{Ij} \int_E [\boldsymbol{\beta} \cdot \boldsymbol{g}^k_I]\, m^k_\sigma \,\mathrm{d}\boldsymbol{x}.$$

Defining the matrix

$$\mathbf{U}_{I\sigma} := \int_E [\boldsymbol{\beta} \cdot \boldsymbol{g}^k_I]\, m^k_\sigma \,\mathrm{d}\boldsymbol{x}$$

we have

$$\mathbf{T}^{\boldsymbol{\beta}} = -(\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0}}_k)^{\mathrm{T}}\mathbf{U}.$$

10.4 Term $(\gamma p_h, q_h)_{0,E}$

The corresponding local matrix is H^γ defined in (38).

10.5 Complete Stiffness Matrix

The local stiffness matrix K for the complete problem is then given by:

$$\mathsf{K} := \begin{bmatrix} \mathsf{K}_\mathsf{c}^a + \mathsf{K}_\mathsf{s}^a & -\mathsf{W}^\mathrm{T} + \mathsf{T}^\beta \\ \mathsf{W} & \mathsf{H}^\gamma \end{bmatrix}.$$

11 Algorithm for the Mixed Formulation

We summarize here the steps needed to compute the VEM local matrix for the mixed approximation. We indicate in square brackets the size of each matrix.

11.1 L^2 Projection

1. Compute

$$\mathsf{G}_{IJ} = \int_E \boldsymbol{g}_I^k \cdot \boldsymbol{g}_J^k \,\mathrm{d}\boldsymbol{x} \qquad [2n_k \times 2n_k]$$

2. Compute the $[n_k \times N_\mathrm{dof}]$ matrix $\mathsf{W}_\mathbf{1}$

$$[\mathsf{W}_\mathbf{1}]_{\tau i} = -|E|\,\mathrm{dof}^{\boldsymbol{g}}_{\tau-1}(\boldsymbol{\phi}_i) = \begin{cases} -|E| & \text{if } i \text{ corresponds to } \tau - 1 \\ 0 & \text{otherwise} \end{cases}$$

3. Compute

$$\mathsf{W}_\mathbf{2} \quad \text{(boundary term)} \qquad [n_k \times N_\mathrm{dof}]$$

4. Set

$$\mathsf{W} = \mathsf{W}_\mathbf{1} + \mathsf{W}_\mathbf{2} \qquad [n_k \times N_\mathrm{dof}]$$

5. Compute

$$\mathbf{H}_{\sigma\tau} = \int_E m_\sigma m_\tau \,\mathrm{d}\boldsymbol{x} \qquad [n_k \times n_k]$$

6. Compute

$$\mathbf{H}^{\#}_{\beta\sigma} = \int_E m_\sigma \, m_{\beta+1} \,\mathrm{d}\boldsymbol{x} \qquad [n_k^{\nabla} \times n_k]$$

7. Set

$$\mathbf{B}_1^{\nabla} = -\mathbf{H}^{\#}\mathbf{H}^{-1}\mathbf{W} \qquad [n_k^{\nabla} \times N_{\mathrm{dof}}]$$

8. Compute

$$\mathbf{B}_2^{\nabla} \quad \text{(boundary term)} \qquad [n_k^{\nabla} \times N_{\mathrm{dof}}]$$

9. Set

$$\mathbf{B}^{\nabla} = \mathbf{B}_1^{\nabla} + \mathbf{B}_2^{\nabla} \qquad [n_k^{\nabla} \times N_{\mathrm{dof}}]$$

10. Compute the $[n_k^{\oplus} \times N_{\mathrm{dof}}]$ matrix $\mathbf{B}^{\oplus}$

$$[\mathbf{B}^{\oplus}]_{\delta i} = |E| \,\mathrm{dof}_{\delta}^{\oplus}(\boldsymbol{\phi}_i) = |E| \,\delta_{\delta i} = \begin{cases} |E| & \text{if } i \text{ corresponds to } \delta \\ 0 & \text{otherwise} \end{cases}$$

11. Set

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}^{\nabla} \\ \mathbf{B}^{\oplus} \end{bmatrix} \qquad [2n_k \times N_{\mathrm{dof}}]$$

12. Set

$$\overset{*}{\boldsymbol{\Pi}}{}_k^{\mathbf{0}} = \mathbf{G}^{-1}\mathbf{B} \qquad [2n_k \times N_{\mathrm{dof}}]$$

13. Compute

$$\mathbf{D}_{jI} := \mathrm{dof}_j(\boldsymbol{g}_I^k) \qquad [N_{\mathrm{dof}} \times 2n_k]$$

14. Set

$$\boldsymbol{\Pi}_k^{\mathbf{0}} = \mathbf{D}\overset{*}{\boldsymbol{\Pi}}{}_k^{\mathbf{0}} \qquad [2N_{\mathrm{dof}} \times 2N_{\mathrm{dof}}]$$

15. Check that

$$\mathbf{G} = \mathbf{B}\mathbf{D}$$

11.2 Coefficient Matrices

1. Compute

$$\mathbf{G}^{\boldsymbol{\nu}}_{IJ} = \int_E \boldsymbol{\nu}\, \boldsymbol{g}^k_I \cdot \boldsymbol{g}^k_J \, \mathrm{d}\boldsymbol{x} \qquad [2n_k \times 2n_k]$$

2. Define

$$\mathbf{U}_{I\sigma} = \int_E [\boldsymbol{\beta} \cdot \boldsymbol{g}^k_I]\, m^k_\sigma \, \mathrm{d}\boldsymbol{x} \qquad [2n_k \times n_k]$$

3. Set

$$\mathbf{T}^{\boldsymbol{\beta}} = -(\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0}}_{\boldsymbol{k}})^{\mathrm{T}}\mathbf{U}. \qquad [2n_k \times n_k]$$

4. Define

$$(\mathbf{H}^{\gamma})_{\alpha\beta} := \int_E \gamma\, m_\alpha m_\beta \, \mathrm{d}\boldsymbol{x} \qquad [n_k \times n_k]$$

11.3 Local Matrix

Set

$$\mathbf{K}^{\boldsymbol{a}}_{\mathbf{c}} = [\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0}}_{\boldsymbol{k}}]^{\mathrm{T}}\mathbf{G}^{\nu}\,\overset{*}{\boldsymbol{\Pi}}{}^{\mathbf{0}}_{\boldsymbol{k}} \quad \text{and} \quad \mathbf{K}^{\boldsymbol{a}}_{\mathbf{s}} = \bar{\nu}\,|E|\,(\mathbf{I} - \boldsymbol{\Pi}^{\mathbf{0}}_{\boldsymbol{k}})^{\mathrm{T}}\,(\mathbf{I} - \boldsymbol{\Pi}^{\mathbf{0}}_{\boldsymbol{k}}).$$

The full local matrix is then

$$\mathbf{K} := \begin{bmatrix} \mathbf{K}^{\boldsymbol{a}}_{\mathbf{c}} + \mathbf{K}^{\boldsymbol{a}}_{\mathbf{s}} & -\mathbf{W}^{\mathrm{T}} + \mathbf{T}^{\boldsymbol{\beta}} \\ \mathbf{W} & \mathbf{H}^{\gamma} \end{bmatrix}.$$

Appendix

We list here the basis $\boldsymbol{g}_\alpha^{\nabla,k}$ and $\boldsymbol{g}_\gamma^{\oplus,k}$ obtained with MATLAB for k up to 5. We point out that in order to have the right scaling, the variable x and y must be replaced by $\left(\frac{x - x_c}{h_E}\right)$ and $\left(\frac{x - y_c}{h_E}\right)$ respectively.

	$\boldsymbol{g}_\alpha^{\nabla,k}$	$\boldsymbol{g}_\gamma^{\oplus,k}$
k=1	[1, 0]	[-y, x]
	[0, 1]	
	[2*x, 0]	
	[y, x]	
	[0, 2*y]	
k=2	[3*x^2, 0]	[-(x*y)/2, x^2]
	[2*x*y, x^2]	[-2*y^2, x*y]
	[y^2, 2*x*y]	
	[0, 3*y^2]	
k=3	[4*x^3, 0]	[-(x^2*y)/3, x^3]
	[3*x^2*y, x^3]	[-x*y^2, x^2*y]
	[2*x*y^2, 2*x^2*y]	[-3*y^3, x*y^2]
	[y^3, 3*x*y^2]	
	[0, 4*y^3]	
k=4	[5*x^4, 0]	[-(x^3*y)/4, x^4]
	[4*x^3*y, x^4]	[-(2*x^2*y^2)/3, x^3*y]
	[3*x^2*y^2, 2*x^3*y]	[-(3*x*y^3)/2, x^2*y^2]
	[2*x*y^3, 3*x^2*y^2]	[-4*y^4, x*y^3]
	[y^4, 4*x*y^3]	
	[0, 5*y^4]	
k=5	[6*x^5, 0]	[-(x^4*y)/5, x^5]
	[5*x^4*y, x^5]	[-(x^3*y^2)/2, x^4*y]
	[4*x^3*y^2, 2*x^4*y]	[-x^2*y^3, x^3*y^2]
	[3*x^2*y^3, 3*x^3*y^2]	[-2*x*y^4, x^2*y^3]
	[2*x*y^4, 4*x^2*y^3]	[-5*y^5, x*y^4]
	[y^5, 5*x*y^4]	
	[0, 6*y^5]	

References

1. B. Ahmad, A. Alsaedi, F. Brezzi, L.D. Marini, A. Russo, Equivalent projectors for virtual element methods. Comput. Math. Appl. **66**(3), 376–391 (2013)
2. P.F. Antonietti, L. Beirão da Veiga, D. Mora, M. Verani, A stream virtual element formulation of the Stokes problem on polygonal meshes. SIAM J. Numer. Anal. **52**(1), 386–404 (2014)
3. M. Arroyo, M. Ortiz, Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods. Int. J. Numer. Methods Eng. **65**(13), 2167–2202 (2006)
4. I. Babuska, U. Banerjee, J.E. Osborn, Survey of meshless and generalized finite element methods: a unified approach. Acta Numer. **12**, 1–125 (2003)
5. L. Beirão da Veiga, G. Manzini, A virtual element method with arbitrary regularity. IMA J. Numer. Anal. **34**(2), 759–781 (2014)

6. L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L.D. Marini, A. Russo, Basic principles of virtual element methods. Math. Models Methods Appl. Sci. **23**(1), 199–214 (2013)
7. L. Beirão da Veiga, F. Brezzi, L.D. Marini, Virtual elements for linear elasticity problems. SIAM J. Numer. Anal. **51**(2), 794–812 (2013)
8. L. Beirão da Veiga, F. Brezzi, L.D. Marini, A. Russo, The hitchhiker's guide to the virtual element method. Math. Models Methods Appl. Sci. **24**(8), 1541–1573 (2014)
9. L. Beirão da Veiga, K. Lipnikov, G. Manzini, *The Mimetic Finite Difference Method for Elliptic Problems*. MS&A, Modeling, Simulation and Applications, vol. 11 (Springer, Berlin, 2014)
10. L. Beirão da Veiga, F. Brezzi, L.D. Marini, A. Russo, $H(div)$ and $H(curl)$-conforming VEM. Numer. Math. **133**(2), 303–332 (2015)
11. L. Beirão da Veiga, F. Brezzi, L.D. Marini, A. Russo, Mixed virtual element methods for general second order elliptic problems on polygonal meshes. ESAIM: M2AN **50**(3), 727–747 (2016)
12. L. Beirão da Veiga, F. Brezzi, L.D. Marini, A. Russo virtual element methods for general second order elliptic problems on polygonal meshes. Math. Models Methods. Appl. Sci. **26**(4), 729–750 (2016)
13. M.F. Benedetto, S. Berrone, S. Pieraccini, S. Scialò, The virtual element method for discrete fracture network simulations. Comput. Methods Appl. Mech. Eng. **280**, 135–156 (2014)
14. J.E. Bishop, A displacement-based finite element formulation for general polyhedra using harmonic shape functions. Int. J. Numer. Methods Eng. **97**(1), 1–31 (2014)
15. F. Brezzi, L.D. Marini, Virtual element methods for plate bending problems. Comput. Methods Appl. Mech. Eng. **253**, 455–462 (2013)
16. F. Brezzi, R.S. Falk, L.D. Marini, Basic principles of mixed virtual element methods. ESAIM Math. Model. Numer. Anal. **48**(4), 1227–1240 (2014)
17. C. Chinosi, L.D. Marini, Virtual Element Methods for fourth order problems: L^2 Estimates. Comput. Math. Appl. (2016)
18. B. Cockburn, The hybridizable discontinuous Galerkin methods, in *Proceedings of the International Congress of Mathematicians*, vol. IV (Hindustan Book Agency, New Delhi, 2010), pp. 2749–2775
19. D. Di Pietro, A. Ern, Hybrid high-order methods for variable-diffusion problems on general meshes. C.R. Acad. Sci. Paris. Ser. I **353**, 31–34 (2015)
20. J. Droniou, R. Eymard, T. Gallouët, R. Herbin, Gradient schemes: a generic framework for the discretisation of linear, nonlinear and nonlocal elliptic and parabolic equations. Math. Models Methods Appl. Sci. **23**(13), 2395–2432 (2013)
21. M.S. Floater, Generalized barycentric coordinates and applications. Acta Numer. **24**, 215–258 (2015)
22. T.-P. Fries, T. Belytschko, The extended/generalized finite element method: an overview of the method and its applications. Int. J. Numer. Methods Eng. **84**(3), 253–304 (2010)
23. A.L. Gain, C. Talischi, G.H. Paulino, On the virtual element method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes. Comput. Methods Appl. Mech. Eng. **282**, 132–160 (2014)
24. R.V. Garimella, J. Kim, M. Berndt, Polyhedral mesh generation and optimization for non-manifold domains, in *Proceedings of the 22nd International Meshing Roundtable*, ed. by J. Sarrate, M. Staten (Springer, Berlin, 2013)
25. D. Mora, G. Rivera, R. Rodríguez, A virtual element method for the Steklov eigenvalue problem. Math. Models Methods Appl. Sci. **25**(08), 1421–1445 (2015)
26. N. Sukumar, E.A. Malsch, Recent advances in the construction of polygonal finite element interpolants. Arch. Comput. Methods Eng. **13**(1), 129–163 (2006)
27. C. Talischi, G.H. Paulino, A. Pereira, I.F.M. Menezes, Polygonal finite elements for topology optimization: a unifying paradigm. Int. J. Numer. Methods Eng. **82**(6), 671–698 (2010)
28. C. Talischi, G.H. Paulino, A. Pereira, I.F.M. Menezes, PolyMesher: a general-purpose mesh generator for polygonal elements written in Matlab. J. Struct. Multidiscip. Optim. **45**(3), 309–328 (2012)
29. J. Wang, X. Ye, A weak Galerkin mixed finite element method for second order elliptic problems. Math. Comput. **83**(289), 2101–2126 (2014)