NTNU
Norwegian University of
Science and Technology

The Virtual Element Method as a Common Framework for Finite Element and Finite Difference Methods

Numerical and Theoretical Analysis

Øystein S Klemetsdal

Master of Science in Physics and Mathematics
Submission date: June 2016
Supervisor: Xavier Raynaud, MATH
Co-supervisor: Halvor Møll Nilsen, SINTEF ICT

Norwegian University of Science and Technology
Department of Mathematical Sciences

Abstract

Consistent discretizations of differential equations on polygonal and polyhedral grids is an active area of research. This is of particular interest in applications where the grid is constructed to capture the physical properties of the domain on which the differential equation is defined, for example in subsurface modelling. Recent developments in this area includes the mimetic finite difference method, which mimics the physical and mathematical properties of the problem. Construction of such methods involves choosing a term to ensure stability. Mimetic finite differences has later evolved into a finite element-like approach called the virtual element method, which is the topic of this thesis.

The work presented here consists of three parts: *(i)* A literature study of the virtual element method for Poisson problems, *(ii)* an analysis of the construction of the stability term of the virtual element method, and *(iii)* a MATLAB implementation of the virtual element method in two and three dimensions, for first and second order, accompanied by numerical examples.

A detailed proof of the well-posedness of the operator involved in constructing the virtual element function space is presented. We also present a detailed proof of that this projection can be calculated exactly for any function in the virtual element function space. Moreover, we show how the stability term can be chosen such that the bilinear form of the virtual element method equals the bilinear form it approximates. This is used to show that the stability term can be chosen such that local stiffness matrix of the first order virtual element is equal to the local stiffness matrix of the first order finite element method for certain cell geometries. It is also used to show that we can choose the stability term to obtain the same system of linear equations as for finite difference stencils. The MATLAB implementation is used to solve simple problems in two and three dimensions. We also use the implementation to investigate how the stability term can be used to minimize the error of the approximation in different norms. Finally, the implementation is compared to some of the most common methods used in reservoir simulations: Two-point and multipoint flux approximation, and mimetic finite differences. The implementation is robust to irregular cell geometries and high aspect ratios, and converges as predicted by the theory. Moreover, for pressure problems, it produces solutions which are close to or better than the solutions produced by the mentioned reservoir simulation methods.

Sammendrag

Konsistente diskretiseringer av differensialligninger på polygonale og polyhedrale grid er et aktivt forskningsfelt. Dette er spesielt interessant i anvendelser der griddet er konstruert for a fange de fysiske egenskapene til domenet differensialligningen er definert, for eksempel i undergrunnsmodellering. En nyvinning innen dette feltet er mimetiske differansemetoder, som hermer etter de fysiske og matematiske egenskapene til problemet. Konstruksjonen av slike metoder involverer å velge en stabiliseringsterm. Mimetiske differansemetoder har senere utviklet seg til en endelig element-lignende tilnærming kalt den virtuelle element metoden, som er temaet for denne masteroppgaven.

Arbeidet presentert her består av tre deler: *(i)* En litteraturstudie av den virtuelle elementmetoden for Poisson problemer, *(ii)* analyse av den virtuelle elementmetodens stabiliseringsterm, og *(iii)* en MATLAB implementasjon av den virtuelle elementmetoden i to og tre dimensjoner, for første og andre orden, akkompagnert av numeriske eksempler.

Vi presenterer et detaljert bevis av at projeksjonsoperatoren involvert i konstruksjonen av virtuelle elementmetoder er veldefinert, samt et detaljert bevis for at denne projeksonen kan regnes eksakt for enhver funksjon i det virtuelle funksjonsrommet. Videre viser vi hvordan stabilitetstermen kan velges slik at den bilineære formen i den virtuelle elementmetoden blir lik den bilineære formen den approksimerer. Dette brukes til å vise at stabiliseringstermen kan velges slik at den lokale stivhetsmatrisen til første ordens virtuelle elementmetoder er lik den lokale stivhetsmatrisen til første ordens endelige elementmetoder for visse cellegeometrier. Videre bruker vi dette til å vise at vi kan velge stabiliseringstermen slik at det resulterende lineære ligningssystemet er likt det vi får fra differansemetoder. MATLAB implementasjonen brukes til å løse enkle problemer i to og tre dimensjoner. Videre brukes den til å minimere feilen til den approksimerte løsningen i ulike normer. Til slutt sammenligner vi implementasjonen med noen av de vanligste metodene innen reservoarsimulering: Topunkts og multipunkts fluksapproksimasjon, og mimetiske differansemetoder. Implementasjonen er robust med hensyn på irregulære cellegeometrier og store aspektratioer, og konvergerer i henhold til teori. Videre produserer den løsninger på trykkproblemer som er nært oppmot eller mer nøyaktig enn løsningene fra de nevnte reservoarsimuleringsmetodene.

Preface

This thesis concludes my master's degree in Industrial Mathematics at the Norwegian University of Science and Technology (NTNU). The work has been preformed in the period from the 10th of January to the 10th of June 2016, and builds upon the work done in the course TMA4500.

I would like to thank everyone at SINTEF ICT Applied Mathematics for showing interest in my work, and patiently sharing their knowledge. In particular, I would like to thank my supervisor, Xavier Raynaud, for thoroughly answering all my questions. Your genuine interest in, and understanding of, mathematics has been a great source of information and inspiration. I would also like to thank my second supervisor, Halvor Møll Nilsen. Your intuitive physicist approach to complex problem solving has been invaluable. Moreover, I would like to thank my fellow students at Industrial Mathematics at NTNU – Sharing knowledge and discussing with fellow students is by far the best way to learn. In particular, I would like to thank Runar Lie Berge for providing me with grids to test my implementation. I can honestly say that this period has been the most fruitful during my time as a student.

Finally, I would like to thank my family and friends for support, and for keeping my mind of this thesis when needed, during these five months of almost complete isolation. And last but not least, I would like to thank my girlfriend for seven years, for patiently supporting me every day. I love you.

Oslo,
June 10, 2016

Øystein Strengehagen Klemetsdal

Contents

Chapter 1

Introduction

> *Perhaps I can best describe my experience of doing mathematics in terms of a journey through a dark unexplored mansion. You enter the first room of the mansion and it's completely dark. You stumble around bumping into the furniture, but gradually you learn where each piece of furniture is.*
>
> Andrew John Wiles

From a mathematical-historical point of view, approximating solutions to differential equations is a rather new topic, and started in the 1930s with the method of finite differences. One of the first rigour definitions of finite differences appeared in a paper in 1928 by Courant, Friedrichs and Lewy [11], where they, among other, discussed approximations of second-order elliptic differential equations. Interestingly enough, the purpose was not to develop numerical techniques for solving differential equations, but rather to derive existence results for such problems. This is also the paper in which they presented the famous CFL condition.

In 1943, building on ideas by Galerkin, Courant developed the finite element method (FEM) [12]. The core idea of FEM is to consider the weak formulation of the differential equation, and divide the domain on which the differential equation is defined into a grid of triangular or tetrahedral cells. One then construct finite-dimensional subspaces of the trial and test function spaces, and defines a suitable set of basis functions for the these. The discrete variational formulation can then be formulated as a set of linear equations, which can be solved to find the approximated solution values at the vertices of the grid. These values are referred to as the degrees of freedom of the method.

Later, the finite element method has been extended to more general cell geometries such as quadrilaterals, hexahedra and triangular prisms (see for example [5, 29]). However, the need for explicit knowledge of the basis functions makes it impossible to define a consistent discretization on a general polygonal or polyhedral grid using the classical finite element framework.

Another widely used numerical method for solving differential equations is the finite volume method (FVM) (see for example [36]). This method is based on discretizing the conservation law associated with the differential equation, and thus captures the physics of the problem. Hence, the method is conservative, in the sense that the discrete formulation also guarantees conservation of the quantity the equation describes. This property makes FVM well-suited for solving flow problems. In contrast to FEM, one allows for trial and test functions that are discontinuous across the cell boundaries. This makes the finite volume method more flexible with respect to cell geometries.

A typical example where we encounter polygonal or polyhedral grids is subsurface modelling [19, 35, 20]. This is an active area of research, and its applications ranges from modeling groundwater reserves, to oil reservoir simulation. In any event, the complexity of the porous media subsurface and the flow impose a close connection between the grid and the reservoir model. In contrast to many other flow simulation disciplines, a given degree of accuracy of the numerical method cannot be obtained on an arbitrary grid – it has to capture the geological and petrophysical properties of the reservoir [20, 19].

As a result of significant research, there exists several consistent discretizations for polygonal and polyhedral grids. Some, like the smoothed FEM [22] and polygonal FEM [31], are similar to the classical finite element method. Others, like the two-point [20, 19] and multipoint flux approximation methods [1, 2] are closer to finite volume methods.

Recently, it has become clear that it is possible to construct consistent first-order finite volume-like methods on polygonal and polyhedral grids by mimicking the underlying properties of the physical problem, and also the mathematical properties of the resulting differential equation. Such methods are know as mimetic finite differences (MFD), see [35] for a comprehensive introduction. Higher-order MFD methods can be obtained by introducing more degrees of freedom, which are associated with quantities other than point values. Specifically, these are moments over the edges, faces and cells of the grid. However, the construction of such higher-order MFD methods has proved cumbersome [33, 35].

It has later turned out that if we disregards the requirement of explicit knowledge of the basis functions on the cell interior, it is possible to obtain a much simpler discretization method, and a more unified theoretical framework, using the same degrees of freedom. The idea is to instead require the basis functions to satisfy a certain differential equation on the interior of each cell, and has lead to a finite element-like method called the virtual element method (VEM) [33, 3, 34]. As a result, we do not know what the basis functions look like on the interior of the cells. However, defining a suitable projection operator from the VEM function space to the space of polynomials of a given degree, we are able to compute this projection of the basis functions using the degrees of freedom. This way, we obtain a consistent method for general polygonal and polyhedral grids.

It is interesting to note that all the methods mentioned above ultimately yields a system of linear equations. Hence, if two different numerical methods

yield the same system of equations, they are, in a way, two different perspectives on the same problem, more than entirely different approaches. Common for both MFD and VEM, is that their construction involves choosing a bilinear form to ensure stability. This stability term can take a number of forms. For MFD methods, it can be chosen such that MFD is equivalent to other methods, see, for example [16, 21, 8]. Analysis of the stability term for VEM has also been done [33, 9], and equivalence between VEM and the polygonal and smoothed finite element methods is discussed in [23].

Remarks

This thesis builds upon the work presented in [17]. Initially, the goal was to develop VEM for reservoir simulations. However, in this thesis, the focus has shifted to a more theoretical perspective, with emphasis on the stability term and equivalence analysis. Some of the material presented in this thesis can also be found in [17]. In particular, a first version of Chapter 2 and Chapter 4 was initially presented here. Part of the MATLAB implementation of second order VEM in two dimensions is also based on this work.

Based on the work presented in this thesis, in the thesis [4], and on work done at SINTEF ICT, an abstract has been submitted and accepted for the SPE Reservoir Simulation Conference 2017 [28], and the work will be presented in a talk at this conference.

Outline

In Chapter 2, the most important theoretical aspects of the virtual element method for the Poisson equation in two and three dimensions are described. In Chapter 3, we discuss how to construct the stability term, and show how it can be chosen in order to make VEM equivalent to other numerical methods. Chapter 4 describes in detail the implementation process of VEM for Poisson problems in two and three dimensions. In Chapter 5, we discuss some of the details of the MATLAB implementation of VEM implemented for this thesis. Then, we use it to solve simple problems in two and three dimensions, look at how the stability term can be used to minimize the error, and compare the implementation to standard reservoir simulation methods. Finally, in Chapter 6, we conclude and discuss further developments of the virtual element method.

For better readability theorems, propositions and lemmas are written in italic types. Moreover, a square (□) indicates the end of a proof, and a circle (∘) indicates the end of an assumption, definition, remark or example.

Chapter 2

The Virtual Element Method

Before I came here I was confused about this subject. Having listened to your lecture I am still confused. But on a higher level, and about more important things.

Unknown

In this chapter, we will go through the theoretical aspects of the virtual element method for Poisson problems. The theory presented here is mainly based on what is found in [33, 3], supplemented with detailed proofs, definitions and remarks. In particular, we present a detailed proof of the well-posedness of the projection operator Π^{∇}.

Before we go into the details of the virtual element method, and explain terms such as the local stiffness bilinear form, we summarize the core idea:

(i) The virtual element space, in which the trial and test functions are contained, consists of all polynomials of degree less than or equal to k, along with some possibly non-polynomial functions.

(ii) We choose the degrees of freedom and the virtual element space in such a way that when one of the two entries is a polynomial of degree less than or equal to k, the local stiffness bilinear form of each element can be computed exactly, *using only the degrees of freedom*.

(iii) In the cases when none of the two entries is a polynomial of degree less than or equal to k, the stiffness bilinear form is only approximated to the right order of magnitude, while we ensure that stability properties are satisfied.

As we will see, we never actually compute the non-polynomial basis functions, but rather, their polynomial projection, which we can calculate from the degrees of freedom. Hence, the term *virtual*.

2.1 Notation

Vectors will be denoted by lower-case bold types. By convention, all vectors are column vectors. For a vector $\boldsymbol{v} = (v_1, \ldots, v_n)^T \in \mathbb{R}^n$, we denote the Euclidean norm by

$$|\boldsymbol{v}| = \sqrt{\sum_{i=1}^{n} v_i^2}.$$

Matrices will be denoted by upper-case bold types. For a matrix $\boldsymbol{M} \in \mathbb{R}^{n\times m}$, we denote the element in the ith row and jth column by $\boldsymbol{M}_{i,j}$. The identity matrix will be denoted by $\boldsymbol{I}$.

A multi-index is an n-tuple of nonnegative integers, and will be denoted in bold types. In particular, if $\boldsymbol{\alpha}$ is a multi-index, we write $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$, and $|\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_n$. Moreover, for $\boldsymbol{x} \in \mathbb{R}^n$, and $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_n)$, we adopt the notation $\boldsymbol{x}^{\boldsymbol{\alpha}} = \prod_{k=1}^{n} x_k^{\alpha_k}$.

For a subset $\Omega \subset \mathbb{R}^d$, we will denote the length ($d = 1$), area ($d = 2$) or volume ($d = 3$) by $|\Omega|$. We will always assume that Ω is bounded. We recall the following Sobolev norm and seminorm on Ω:

$$\|w\|_{H^m(\Omega)} = \left(\sum_{|\boldsymbol{\alpha}|\le m} \|D^{\boldsymbol{\alpha}} w\|^2_{L^2(\Omega)}\right)^{1/2}, \quad |w|_{H^m(\Omega)} = \left(\sum_{|\boldsymbol{\alpha}|= m} \|D^{\boldsymbol{\alpha}} w\|^2_{L^2(\Omega)}\right)^{1/2}.$$

where m is a positive integer, $D^{\boldsymbol{\alpha}}$ is the partial derivative

$$D^{\boldsymbol{\alpha}} = \frac{\partial^{\alpha_1} \cdots \partial^{\alpha_d}}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}},$$

and $\|\cdot\|_{L^2(\Omega)}$ is the L^2-norm:

$$\|w\|_{L^2(\Omega)} = \left((w, w)_{L^2(\Omega)}\right)^{1/2} = \left(\int_\Omega |w|^2 \,\mathrm{d}\boldsymbol{x}\right)^{1/2}.$$

We also recall the following Sobolev spaces

$$H^m(\Omega) := \left\{w : \|w\|_{H^m(\Omega)} < \infty\right\}, \quad H^m_0(\Omega) := \left\{w \in H^m(\Omega) : w|_{\partial\Omega} = 0\right\}.$$

Further, for $H^m(\Omega)$, we use the short-hand notation (m, Ω). In particular, we will use

$$\|\cdot\|_{L^2(\Omega)} = \|\cdot\|_{0,\Omega}, \quad \|\cdot\|_{H^m(\Omega)} = \|\cdot\|_{m,\Omega}, \quad |\cdot|_{H^m(\Omega)} = |\cdot|_{m,\Omega},$$
$$(u, v)_{L^2(\Omega)} = (u, v)_{0,\Omega}.$$

For a differentiable function v we denote its normal derivative $\nabla v \cdot \boldsymbol{n}$ by $\partial_n v$, where $\boldsymbol{n}$ is the outward normal of $\partial\Omega$.

Finally, we denote the space of all polynomials on Ω of degree less than or equal to k to by $\mathbb{P}_k(\Omega)$. By convention, $\mathbb{P}_{-1} = \{0\}$.

This notation will be used throughout the thesis.

2.2 A model problem

In order to have something more than just an abstract framework, we will introduce a simple model problem. As mentioned in Chapter 1, consistent discretizations for polygonal and polyhedral grids are particularly suited for applications in reservoir simulation. Hence, we will consider the case of an incompressible fluid flowing through a porous medium. This can be described by the equations

$$\nabla \cdot \boldsymbol{v} = q, \tag{2.1a}$$

$$\boldsymbol{v} = -\frac{\boldsymbol{K}}{\mu}\big(\nabla p - \rho g \nabla z\big), \tag{2.1b}$$

where p is the fluid pressure, μ is the dynamic viscosity of the fluid, g is the gravitational constant, z is the vertical coordinate, and q represents a source or sink term. The tensor $\boldsymbol{K}$ is the permeability of the porous medium – its ability to transmit a single fluid for a given pressure drop. Equation (2.1b) is known as Darcy's law, where $\boldsymbol{v}$ is the macroscopic velocity. See, for instance, [19, Chapter 6.1] for a derivation. For simplicity, we assume that $\boldsymbol{K} = K\boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix, which corresponds to an isotropic porous medium, and K, μ and ρ are constant. We can now combine equation (2.1a) and (2.1b) into

$$-\frac{K}{\mu}\Delta\big(p - \rho g z\big) = q.$$

We introduce the fluid potential $u \coloneqq p - \rho g z$, and the modified source term $f \coloneqq \frac{\mu}{K} q$, and assume that the subsurface occupies a polygonal domain $\Omega \subset \mathbb{R}^d$, where $d = 2$ or 3. If u vanishes on the boundary of Ω, this leads to the Poisson equation:

$$\begin{aligned} -\Delta u &= f, \quad \boldsymbol{x} \in \Omega, \\ u &= 0, \quad \boldsymbol{x} \in \partial\Omega. \end{aligned} \tag{2.2}$$

We assume that the source term f is in $L^2(\Omega)$. For simplicity, we write $V \coloneqq H_0^1(\Omega)$, and multiply by a function $v \in V$ and integrate over Ω to obtain

$$\begin{aligned} -(\Delta u, v)_{0,\Omega} &= -(\partial_n u, v)_{0,\partial\Omega} + (\nabla u, \nabla v)_{0,\Omega} \\ &= (\nabla u, \nabla v)_{0,\Omega} \quad \forall v \in V. \end{aligned}$$

This leads to the weak formulation of (2.2):

$$\text{Find } u \in V \text{ such that} \quad a(u, v) = (f, v)_{0,\Omega} \quad \forall v \in V, \tag{2.3}$$

where $a(u, v) = (\nabla u, \nabla v)_{0,\Omega}$. The finite element method was popularized as a method for solving problems in continuum mechanics [32]. Due to this fact,

we usually refer to a as the stiffness bilinear form, and the right-hand side as the load term. The function space V is called the test function space, and the functions v are called test functions. In problems with inhomogeneous boundary conditions, we sometimes need a second function space U, called the trial function space, and we seek a solution u among all trial functions. In this case, however, the trial and test function spaces coincide.

We note that (2.3) has a unique solution. Indeed; the Poincaré inequality [15, Chapter 5.6] gives

$$\|v\|_{1,\Omega} = \|v\|_{0,\Omega} + |v|_{1,\Omega} \leq C|v|_{1,\Omega} \quad \forall v \in V,$$

where C is a positive constant. It follows that a is coercive:

$$a(v,v) = |v|^2_{1,\Omega} \geq C^{-2}\|v\|^2_{1,\Omega} \quad \forall v \in V.$$

Moreover, a is continuous, since

$$a(u,v) \leq |u|_{1,\Omega}|v|_{1,\Omega} \leq \|u\|_{1,\Omega}\|v\|_{1,\Omega} \quad \forall u, v \in V.$$

Finally, we have that $(f,\cdot)_{0,\Omega}$ is continuous:

$$(f,v)_{0,\Omega} \leq \|f\|_{0,\Omega}\|v\|_{0,\Omega} \leq \|f\|_{1,\Omega}\|v\|_{1,\Omega} \quad \forall v \in V.$$

Hence, by the Lax-Milgram theorem [15, Chapter 6.2], the weak formulation (2.3) has a unique solution.

2.3 Abstract framework

For the sake of simplicity, we will follow the weak formulation (2.3). However, the following is easily generalized to Poisson problems with inhomogeneous boundary conditions, both Dirichlet and Neumann.

We will need the following definitions:

Definition 2.1. Let K be a polygon in $\mathbb{R}^2$ or a polyhedron in $\mathbb{R}^3$. The diameter h_K of K is the longest distance between any two vertices of K. ◦

Definition 2.2. A grid is a set $\mathcal{T}_h$ of adjacent, non-overlapping polygons in $\mathbb{R}^2$ or polyhedra in $\mathbb{R}^3$. We refer to a polygon or polyhedron K of $\mathcal{T}_h$ as a cell. The diameter h of $\mathcal{T}_h$ is defined as $h := \max_{K\in\mathcal{T}_h} h_K$. ◦

We let the grid $\mathcal{T}_h$ be a decomposition of Ω into a finite number of non-overlapping polygons in $\mathbb{R}^2$, or polyhedra $\mathbb{R}^3$, with diameter h. The bilinear form $a(\cdot,\cdot)$ and seminorm $|\cdot|_{1,\Omega}$ can be split as follows:

$$a(u,v) = \sum_{K\in\mathcal{T}_h} a^K(u,v), \quad |v|_{1,\Omega} = \left(\sum_{K\in\mathcal{T}_h} |v|^2_{1,K}\right)^{1/2}, \quad \forall u, v \in V. \tag{2.4}$$

Since we will work with functions that are only piecewise in $H^1(\Omega)$, we will also need the following definition:

Definition 2.3. We define the function space

$$H^1(\mathcal{T}_h) \coloneqq \prod_{K\in\mathcal{T}_h} H^1(K),$$

along with the broken H^1-seminorm,

$$|v|_{h,1} \coloneqq \left(\sum_{K\in\mathcal{T}_h} |\nabla v|^2_{0,K}\right)^{1/2}. \tag{2.5}$$

∘

Note that $|\cdot|_{h,1}$ is indeed a seminorm, and not a norm, since it is zero for all piecewise constant functions.

For a given degree of accuracy $k \geq 1$, we consider grids $\mathcal{T}_h$ that satisfies the following:

Assumption 2.4. For each h, there exists

(i) a subspace $V_k(K) \subset H^1(K)$ for each $K \in \mathcal{T}_h$;

(ii) a subspace $V_h \subset V \cap \prod_{K\in\mathcal{T}_h} V_k(K)$;

(iii) a symmetric, bilinear form $a_h : V_h \times V_h \to \mathbb{R}$, which can be split as

$$a_h(u_h, v_h) = \sum_{K\in\mathcal{T}_h} a_h^K(u_h, v_h) \quad \forall u_h, v_h \in V_h, \tag{2.6}$$

where each $a_h^K(\cdot,\cdot)$ is a symmetric, positive definite bilinear form on $V_h|_K \times V_h|_K$;

(iv) an element $f_h \in V_h'$, where V_h' is the dual space of V_h.

∘

We want to define these objects in such a way that we have a unique solution $u_h \in V_h$ to the discrete version of (2.3):

$$\text{Find } u_h \in V_h \text{ such that} \quad a_h(u_h, v_h) = \langle f_h, v_h\rangle \quad \forall v_h \in V_h, \tag{2.7}$$

where $\langle\cdot,\cdot\rangle$ denotes the pairing of V_h with V_h'. Moreover, we want the solution to approximate the function nicely. In particular, if u is a sufficiently smooth solution to (2.3), and $k \geq 1$ is the target degree of accuracy, we want

$$|u - u_h|_{1,\Omega} \leq Ch^k |u|_{k+1,\Omega}$$

for some positive constant C. To this end, we make the following assumptions on our bilinear form:

Assumption 2.5. For all h, and all $K \in \mathcal{T}_h$, we have an integer $k \geq 1$ such that, for all $v_h \in V_h|_K$, we have

(i) k**-consistency**:

$$a_h^K(v_h, p) = a^K(v_h, p) \quad \forall p \in \mathbb{P}_k(K). \tag{2.8}$$

(ii) **Stability**: There exists two positive constants c_1 and c_2, independent of h and K, such that

$$c_1 a^K(v_h, v_h) \leq a_h^K(v_h, v_h) \leq c_2 a^K(v_h, v_h). \tag{2.9}$$

The integer k is then said to be the *degree of accuracy* of the method. ◦

The symmetry and stability properties implies continuity of a_h:

$$\begin{aligned} a_h^K(u_h, v_h) &\leq \left(a_h^K(u_h, u_h)\right)^{1/2} \left(a_h^K(v_h, v_h)\right)^{1/2} \\ &\leq c_2 \left(a^K(u_h, u_h)\right)^{1/2} \left(a^K(v_h, v_h)\right)^{1/2} \\ &= c_2 |u_h|_{1,K} |v_h|_{1,K} \quad \forall u_h, v_h \in V_h|_K. \end{aligned}$$

We now have the following abstract convergence theorem:

Theorem 2.6. *Let u be the solution to (2.3). Under Assumption 2.4 and Assumption 2.5, the discrete problem (2.7) has a unique solution u_h. Moreover, for every approximation $u_I \in V_h$ of u, and every approximation u_π of u that is piecewise in $\mathbb{P}_k(\Omega)$, we have*

$$|u - u_h|_{1,\Omega} \leq C\Big(|u - u_I|_{1,\Omega} + |u - u_\pi|_{h,1} + \|f - f_h\|_{V_h'}\Big),$$

where C is a constant depending only c_1 and c_2, and $\|\cdot\|_{V_h'}$ is the dual norm:

$$\|f - f_h\|_{V_h'} \coloneqq \sup_{v_h \in V_h} \frac{\langle f - f_h, v_h \rangle}{\|v_h\|_V}. \tag{2.10}$$

This theorem is presented in [33], and a detailed proof is given in [17]. Note that we have three contributions to the error of the approximation u_h of u: The first is the global error $|u - u_I|_{1,\Omega}$, reflecting the error one gets by using of V_h in stead of V. The second is the local error $|u - u_\pi|_{h,1}$, which reflects the error of the polynomial approximation. The third term reflects the error $\|f - f_h\|_{V_h'}$ of approximating f by f_h.

2.4 The virtual element spaces and projectors

We will now explain how to construct the virtual element spaces and projectors. In order to do this, we need the following definitions:

Definition 2.7. Let K be a polygon in $\mathbb{R}^2$ or a polyhedron in $\mathbb{R}^3$. We define the centroid $\boldsymbol{x}_K$ of K by

$$\boldsymbol{x}_K = \frac{1}{|K|} \int_K \boldsymbol{x} \, \mathrm{d}\boldsymbol{x}.$$

○

Definition 2.8. Let K be a polygon in $\mathbb{R}^2$ or a polyhedron in $\mathbb{R}^3$, with diameter h_K and centroid $\boldsymbol{x}_K$. The scaled monomials of degree less than or equal to k is defined as

$$\mathcal{M}_k(K) = \left\{ m^{\boldsymbol{\alpha}} := \left(\frac{\boldsymbol{x} - \boldsymbol{x}_K}{h_K} \right)^{\boldsymbol{\alpha}} : |\boldsymbol{\alpha}| \leq k \right\}.$$

We also define the function space

$$\mathcal{M}_k^*(K) = \left\{ m^{\boldsymbol{\alpha}} := \left(\frac{\boldsymbol{x} - \boldsymbol{x}_K}{h_K} \right)^{\boldsymbol{\alpha}} : |\boldsymbol{\alpha}| = k \right\}.$$

○

We will simply refer to a scaled monomial as a monomial. It follows immediately that $\dim \mathcal{M}_k(K) = \dim \mathbb{P}_k(K)$. Moreover, the scaled monomials are linearly independent, so that we can use $\mathcal{M}_k(K)$ as a basis for $\mathbb{P}_k(K)$.

In order to simplify notation, we introduce the following mapping between the positive integers and multi-indices of two elements:

$$\begin{aligned} &1 \leftrightarrow (0,0), \quad 2 \leftrightarrow (1,0), \quad 3 \leftrightarrow (0,1), \\ &4 \leftrightarrow (2,0), \quad 5 \leftrightarrow (1,1), \quad 6 \leftrightarrow (0,2), \quad \ldots, \end{aligned}$$

and between integers and multi-indices of three elements:

$$\begin{aligned} &1 \leftrightarrow (0,0,0), \quad 2 \leftrightarrow (1,0,0), \quad 3 \leftrightarrow (0,1,0), \quad 4 \leftrightarrow (0,0,1), \quad 5 \leftrightarrow (2,0,0), \\ &6 \leftrightarrow (1,1,0), \quad 7 \leftrightarrow (0,2,0), \quad 8 \leftrightarrow (1,0,1), \quad 9 \leftrightarrow (0,1,1), \quad 10 \leftrightarrow (0,0,2), \quad \ldots \end{aligned}$$

This gives a natural way of ordering the monomials. For example, in two dimensions, we have $m^{(1,0)} = m^2$. Note that for a monomial m^α with $\alpha \leq n_k$, we then know that $m^\alpha \in \mathcal{M}_k(K)$.

Next, we define the L^2-orthogonal projection operator Π_k^Ω onto the space of polynomials of degree less than or equal to k:

Definition 2.9. Consider a domain $\Omega \subset \mathbb{R}^d$. For $k \geq 0$, the L^2-orthogonal projection operator $\Pi_k^\Omega : L^2(\Omega) \to \mathbb{P}_k(\Omega)$ is defined such that, for a function $v \in L^2(\Omega)$, $\Pi_k^\Omega v$ is the unique element in $\mathbb{P}_k(\Omega)$ satisfying

$$\left(v - \Pi_k^\Omega v, p\right)_{0,\Omega} = 0 \quad \forall p \in \mathbb{P}_k(\Omega).$$

○

Whenever it is clear from the context, we omit the degree k and domain Ω, and simply write Π.

2.4.1 The two-dimensional case

As mentioned earlier, the polynomials of degree less than or equal to k is contained in the virtual element space. We recall the dimension of $\mathbb{P}_k(\mathbb{R}^2)$:

$$n_k \coloneqq \dim \mathbb{P}_k\left(\mathbb{R}^2\right) = \frac{1}{2}(k+1)(k+2).$$

Constructing the local 2D virtual element space

We consider a polygon K, with n edges, centroid $\boldsymbol{x}_K$ and diameter h_K. An example of such a polygon is shown in Figure 2.1.

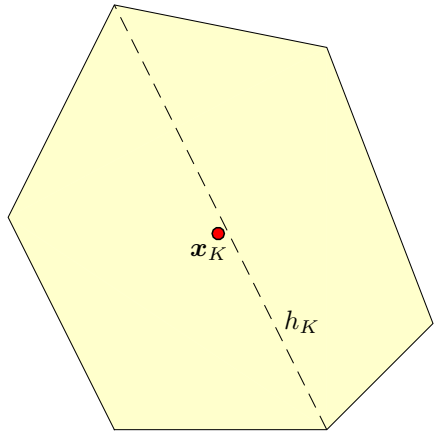


Figure 2.1: Example polygon, with centroid $\boldsymbol{x}_K$ and cell diameter h_K indicated.

For $k \geq 1$, we define the function space

$$B_k(\partial K) \coloneqq \left\{v \in C^0(\partial K) : v|_E \in \mathbb{P}_k(E) \quad \forall E \subset \partial K\right\}. \tag{2.11}$$

A function $v \in B_k(\partial K)$ is uniquely determined by its values at each of the n vertices, together with the values at $k-1$ additional points on each of the edges. Hence, $B_k(\partial K)$ is a linear space of dimension $n + (k-1)n = nk$.

We now introduce an important ingredient in the virtual element method:

Definition 2.10. Consider a polygon K. For $k \geq 1$, we define the projection operator $\Pi_k^{\nabla,K} : H^1(K) \to \mathbb{P}_k(K)$ such that, for a function $v \in H^1(K)$, $\Pi_k^{\nabla,K} v$ is the unique element in $\mathbb{P}_k(K)$ satisfying

$$a^K\left(v - \Pi_k^{\nabla,K} v, p\right) = 0, \quad \forall p \in \mathbb{P}_k(K), \tag{2.12}$$

and

$$\int_{\partial K} \left(v - \Pi_k^{\nabla,K} v\right) \mathrm{d}s = 0 \quad \text{for } k = 1, \tag{2.13a}$$

$$\int_K \left(v - \Pi_k^{\nabla,K} v\right) \mathrm{d}\boldsymbol{x} = 0 \quad \text{for } k \geq 2. \tag{2.13b}$$

∘

Again, whenever the degree and domain is clear, we simply write Π^∇. We note that $\Pi^\nabla p = p$ for all $p \in \mathbb{P}_k(K)$, since, by (2.12), their gradients coincide, while (2.13a) or (2.13b) takes care of the constant part. Hence, Π^∇ is indeed a projection onto $\mathbb{P}_k(K)$.

It is natural to wonder if, for a given $v \in H^1(K)$, its projection $\Pi^\nabla v$ well defined. This is indeed the case, and we state it as a theorem:

Theorem 2.11. *The projection operator Π^∇ is well-posed.*

Proof. Let $u \in H^1(K)$. We want to show that the set of equations defining $\Pi^\nabla u$ has a unique solution. For a function $v \in H^1(K)$, we define

$$\bar{v} = \begin{cases} \int_{\partial K} v \,\mathrm{d}s & \text{for } k = 1, \\ \int_K v \,\mathrm{d}\boldsymbol{x} & \text{for } k \geq 2. \end{cases}$$

Writing $w = \Pi^\nabla u$, we see that the equations defining w can be written as

$$\begin{aligned} a^K(w, p) &= a^K(u, p), \quad \forall p \in \mathbb{P}_k(K), \\ \bar{w} &= \bar{u}. \end{aligned} \tag{2.14}$$

We define

$$q = \begin{cases} w - \frac{1}{|\partial K|}\bar{u} & \text{for } k = 1, \\ w - \frac{1}{|K|}\bar{u} & \text{for } k \geq 2, \end{cases}$$

and reformulate (2.14) as

$$a^K(q, p) = a^K(u - \bar{u}, p) \quad \forall p \in \mathbb{P}_k(K), \tag{2.15a}$$

$$\bar{q} = 0. \tag{2.15b}$$

Next, we define the function space

$$\bar{\mathbb{P}}_k(\Omega) = \{p \in \mathbb{P}_k(\Omega) : \bar{p} = 0\}.$$

Since both w and $\bar{u}$ are in $\mathbb{P}_k(K)$, it is clear from (2.15b) that $q \in \bar{\mathbb{P}}_k(K)$. Moreover, we note that any polynomial $p \in \mathbb{P}_k(K)$ can be decomposed into a constant term and a term in $\bar{\mathbb{P}}_k(K)$ as

$$p = \begin{cases} \frac{1}{|\partial K|}\bar{p} + \left(p - \frac{1}{|\partial K|}\bar{p}\right) & \text{for } k = 1, \\ \frac{1}{|K|}\bar{p} + \left(p - \frac{1}{|K|}\bar{p}\right) & \text{for } k \geq 2. \end{cases}$$

Since $a^K(u, C) = (\nabla u, \nabla C)_{0,K} = 0$ for any constant C, it is clear that we can replace $\mathbb{P}_k(K)$ by $\bar{\mathbb{P}}_k(K)$ in (2.15a). We now define the functional

$$\langle f, \cdot \rangle := a^K(u - \bar{u}, \cdot),$$

and rewrite (2.15) in the following way:

$$\text{Find } q \in \bar{\mathbb{P}}_k(K) \text{ such that} \quad a^K(q, p) = \langle f, p \rangle \quad \forall p \in \bar{\mathbb{P}}_k(K). \tag{2.16}$$

Now, we define a basis $\{p_i\}_i$ of $\bar{\mathbb{P}}_k(K)$, and express q in this basis as $q = \sum_i \xi_i p_i$, where $\xi_i \in \mathbb{R}$ for all i. The weak formulation (2.16) can now be stated in matrix form as

$$\boldsymbol{A}\boldsymbol{\xi} = \boldsymbol{b}, \tag{2.17}$$

where

$$\boldsymbol{\xi} = \left(\xi_1, \dots, \xi_{\dim \bar{\mathbb{P}}_k(K)}\right)^T,$$

the matrix $\boldsymbol{A}$ is such that $\boldsymbol{A}_{i,j} = a^K(p_i, p_j)$, and $\boldsymbol{b}$ is the vector such that $\boldsymbol{b}_i = \langle f, p_i \rangle$.

It is now clear that Π^∇ is well-posed if (2.17) has a unique solution. To this end, consider a function $p \in \bar{\mathbb{P}}_k(K)$ such that $a^K(p, p) = 0$. Then $p = C$, where C is a constant. Moreover,

$$0 = \bar{p} = \begin{cases} C|\partial K| & \text{for } k = 1, \\ C|K| & \text{for } k \geq 2, \end{cases}$$

which gives that $C = 0$. Hence, $a^K(p, p) = 0$ if and only if $p = 0$. It follows that $\boldsymbol{A}$ is invertible, so that (2.17) has a unique solution. We conclude that the projection operator Π^∇ is well-posed. □

Remark 2.12. Note that Π^∇ is linear: Let K be a polygon in $\mathbb{R}^2$, and let u and v be two functions in $H^1(K)$. Then, the definition of $\Pi^\nabla(u + v)$ reads

$$\begin{aligned} a^K\left(u + v - \Pi^\nabla(u + v), p\right) &= 0 \quad \forall p \in \mathbb{P}_k(K), \\ \int_{\partial K} \left(u + v - \Pi^\nabla(u + v)\right) \mathrm{d}s &= 0 \quad \text{for } k = 1, \\ \int_K \left(u + v - \Pi^\nabla(u + v)\right) \mathrm{d}\boldsymbol{x} &= 0 \quad \text{for } k \geq 2. \end{aligned}$$

Moreover, we have

$$\begin{aligned} a^K\left(u + v - \left(\Pi^\nabla u + \Pi^\nabla v\right), p\right) &= 0 \quad \forall p \in \mathbb{P}_k(K), \\ \int_{\partial K} \left(u + v - \left(\Pi^\nabla u + \Pi^\nabla v\right)\right) \mathrm{d}s &= 0 \quad \text{for } k = 1, \\ \int_K \left(u + v - \left(\Pi^\nabla u + \Pi^\nabla v\right)\right) \mathrm{d}\boldsymbol{x} &= 0 \quad \text{for } k \geq 2. \end{aligned}$$

By the well-posedness of Π^∇, it follows that

$$\Pi^\nabla(u+v) = \Pi^\nabla u + \Pi^\nabla v.$$

∘

We can now define our local virtual element space:

Definition 2.13. For $k \geq 1$, we define the two-dimensional local virtual element space $V_k(K)$ on a polygon K:

$$\begin{aligned} V_k(K) \coloneqq \big\{ v_h \in & H^1(K) : v_h|_{\partial K} \in B_k(\partial K),\ \Delta v_h|_K \in \mathbb{P}_k(K), \\ & \big(v_h - \Pi_k^\nabla v_h, m\big)_{0,K} = 0 \quad \forall m \in \mathcal{M}_{k-1}^*(K) \cup \mathcal{M}_k^*(K) \big\}. \end{aligned} \tag{2.18}$$

∘

We equip our virtual element space with the following degrees of freedom:

Definition 2.14. For a function v_h in the local virtual element space $V_k(K)$, we choose the following degrees of freedom:

♢ $\mathcal{V}^K$: The values of v_h at the vertices of K,

♢ $\mathcal{E}^K$: For $k \geq 2$, the moments on each edge E of K

$$|E|^{-1}(v_h, m)_{0,E} \quad \forall m \in \mathcal{M}_{k-2}(E),$$

♢ $\mathcal{P}^K$: For $k \geq 2$, the moments

$$|K|^{-1}(v_h, m)_{0,K} \quad \forall m \in \mathcal{M}_{k-2}(K).$$

∘

In Figure 2.2, we have indicated the degrees of freedom of a second order VEM for the polygon shown in Figure 2.1.

Remark 2.15. Notice that for each edge E, since the dimension of $\mathcal{M}_k(E)$ is $k+1$, we have $k-1$ degrees of freedom for each of the n edges. For a function $v_h \in V_1(K)$, we know that $v_h|_E \in \mathbb{P}_k(E)$ for all edges E of K. Hence, we are free to use the following alternative definition of $\mathcal{E}^K$

♢ $\mathcal{E}^K$: For $k \geq 2$, the values of v_h at $k-1$ points on each edge E of K.

This is the degrees of freedom used in [33, 34]. ∘

We now have $n + (k-1)n + \dim \mathbb{P}_{k-2}(K)$ degrees of freedom in total, which we write as

$$N^K \coloneqq nk + \frac{1}{2}k(k-1).$$

We now have the following proposition:

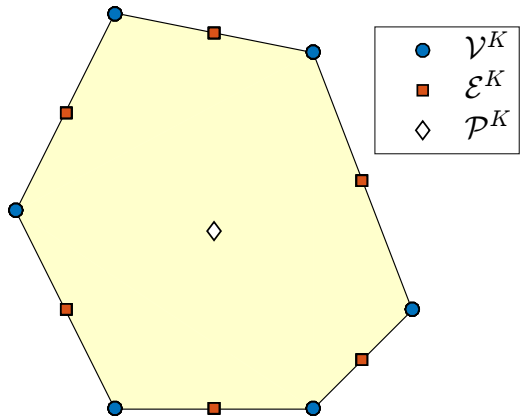


Figure 2.2: Degrees of freedom of $V_2(K)$.

Proposition 2.16. *Any function $v_h \in V_k(K)$ is uniquely determined from the degrees of freedom $\mathcal{V}^K$, $\mathcal{E}^K$ and $\mathcal{P}^K$.*

The proof is based on the observation that $\mathcal{V}^K$ and $\mathcal{E}^K$ prescribes a function $v_h \in V_k(K)$ on ∂K, while $\mathcal{P}^K$ prescribes $\Pi_{k-2} v_h$ in K. Hence, it is sufficient to show that if $v_h|_{\partial K} = 0$ and $\Pi_{k-2} v_h = 0$, then $v_h \equiv 0$. The complete proof is given in [33, 3].

Remark 2.17. For any $v_h \in V_k(K)$, we can compute its projection $\Pi^\nabla v_h$ using the degrees of freedom $\mathcal{V}^K$, $\mathcal{E}^K$ and $\mathcal{P}^K$, see Theorem 4.3 and [3, 33]. As we will see, it is this projection that allows us to construct a consistent discretization without knowing the exact expression of the function v_h on the interior of K. ∘

Remark 2.18. We could have made the somewhat easier choice of virtual element space

$$U_k(K) \coloneqq \{v_h \in H^1(K) : v_h|_{\partial K} \in B_k(\partial K),\ \Delta v_h|_K \in \mathbb{P}_{k-2}(K)\}. \tag{2.19}$$

This function space uses the same degrees of freedom as $V_k(K)$, and $\Pi^\nabla v_h$ is thus computable for any function $v_h \in U_k(K)$ as well. The reason why we are not using this definition, will become clear later. ∘

Projection error

In order to ensure existence of a local approximation $w_\pi \in \mathbb{P}_k(K)$ of a smooth function w on K with optimal approximation properties, we need another assumption on our grid $\mathcal{T}_h$. We denote the length of an edge by h_E, and assume the following:

Assumption 2.19. There exists a positive, real number γ, such that for all h,

(i) for every element $K \in \mathcal{T}_h$, for every edge E of K, $h_E \geq \gamma h_K$;

(ii) every element K is star-shaped with respect to all points in a disk of radius greater than or equal to γh_K.

◦

We can now state the following proposition:

Proposition 2.20. *Under Assumption 2.19, there exists a constant C, only dependent on k and γ, such that, for every s with $1 \leq s \leq k+1$, and for every $w \in H^s(K)$, there exists a $w_\pi \in \mathbb{P}_k(K)$ such that*

$$\|w - w_\pi\|_{0,K} + h_K |w - w_\pi|_{1,K} \leq C h_K^s |w|_{s,K}.$$

See [33] for details.

Constructing the global 2D virtual element space

We are now ready to patch together our global virtual element space.

Definition 2.21. For every decomposition $\mathcal{T}_h$ of Ω into polygonal cells K, and for every $k \geq 1$, we define the global virtual element space V_h:

$$V_h \coloneqq \{v_h \in V : v_h|_K \in V_k(K) \quad \forall K \in \mathcal{T}_h\}.$$

◦

Considering the Poisson problem (2.7), it follows from the definition of $V_k(K)$ that in order to uniquely determine a function $v_h \in V_h$, we need

$$N \coloneqq N_V + N_E(k-1) + N_P \frac{1}{2} k(k-1)$$

degrees of freedom, where N_V, N_E and N_P are the total number of internal vertices, internal edges, and polygons in $\mathcal{T}_h$. We define these as follows:

Definition 2.22. For a function v_h in the global virtual element space V_h, we choose the following N degrees of freedom:

♢ $\mathcal{V}$: The values of v_h at the internal vertices;

♢ $\mathcal{E}$: For $k \geq 2$, the moments on each internal edge E of $\mathcal{T}_h$

$$|E|^{-1}(v_h, m)_{0,E} \quad \forall m \in \mathcal{M}_{k-2}(E).$$

♢ $\mathcal{P}$: For $k \geq 2$, the moments on each polygon K in $\mathcal{T}_h$

$$|K|^{-1}(v_h, m)_{0,K} \quad \forall m \in \mathcal{M}_{k-2}(K).$$

◦

From Proposition 2.16, we know that any function $v_h \in V_h$ can be uniquely determined from the degrees of freedom $\mathcal{V}$, $\mathcal{E}$ and $\mathcal{P}$.

Interpolation error

From the construction of the local virtual element space, we know that for a sufficiently smooth function w vanishing on $\partial\Omega$, we can always find a function $w_I \in V_h$ such that w and w_I share the same degrees of freedom. In order to state this more precisely, we need the following definition:

Definition 2.23. Numbering the degrees of freedom of V_h from 1 to N, we define the functionals $\{\chi^i\}_{i=1}^N$, that, for a sufficiently smooth function ϕ, associates to it the ith degree of freedom $\chi^i(\phi)$, defined in Definition 2.22. We will also use these functionals locally on each cell. The degrees of freedom will then be the ones defined in Definition 2.14, and we number the functionals from 1 to N^K. ∘

Hence, for every sufficiently smooth function w vanishing on $\partial\Omega$, there exists a unique interpolant $w_I \in V_h$ such that

$$\chi^i(w - w_I) = 0 \quad \forall i = 1, \dots, N. \tag{2.20}$$

This leads to the following proposition:

Proposition 2.24. *Under Assumption 2.19, there exists a constant C, depending only on k and γ, such that for every s with $2 \leq s \leq k+1$, for every h, for all $K \in \mathcal{T}_h$ and for every $w \in H^s(K)$, the interpolant $w_I \in V_h$ defined in (2.20) satisfies*

$$\|w - w_I\|_{0,K} + h_K|w - w_I|_{1,K} \leq Ch_K^s|w|_{s,K}.$$

See [33, 3] for details.

Computing the L^2-projection

We recall from Definition 2.9 that $(v_h - \Pi_k v_h, m)_{0,K} = 0$ for all monomials m up to order k. Hence, in order to find $\Pi_k v_h$ for a function $v_h \in V_k(K)$, we need to compute $(v_h, m)_{0,K}$ for all monomials up to order k. But as explained in [3, 34], these are now given by the degrees of freedom $\mathcal{P}^K$, along with the the definition of $V_k(K)$. We state it here as a proposition:

Proposition 2.25. *For $k \geq 1$ and for all $v_h \in V_k(K)$, the L^2-projection $\Pi_k v_h$ can be computed from the degrees of freedom $\mathcal{P}^K$. In particular, for $k = 1$ and $k = 2$, $\Pi_k v_h = \Pi_k^\nabla v_h$.*

Proof. Let $v_h \in V_k(K)$. From the definition of Π_k, we know that we can find $\Pi_k v_h$ by computing $(v_h, m)_{0,K}$ for all $m \in \mathcal{M}_k(K)$. Since $v_h \in V_k(K)$, the first n_{k-2} of the inner products are given from $\mathcal{P}^K$. Moreover, from the definition of $V_k(K)$, we have

$$(v_h, m)_{0,K} = \left(\Pi_k^\nabla v_h, m\right)_{0,K} \quad \forall m \in \mathcal{M}_{k-1}^*(K) \cup \mathcal{M}_k^*(K).$$

Since we are able to compute $\Pi_k^\nabla v_h$, we can compute all inner products $(v_h, m)_{0,K}$, and hence $\Pi_k v_h$. In particular, for $k = 1$, we have, from the definitions of Π_1,

Π_1^∇ and $V_1(K)$, that for all $m \in \mathcal{M}_0^*(K) \cup \mathcal{M}_1^*(K) \equiv \mathcal{M}_1(K)$,

$$\left(\Pi_1^\nabla v_h, m\right)_{0,K} = (v_h, m)_{0,K} = \left(\Pi_1 v_h, m\right)_{0,K},$$

so that $\Pi_1 v_h = \Pi_1^\nabla v_h$ for all functions v_h in $V_1(K)$. For $k = 2$, we have from the definitions of Π_2^∇ and Π_2 that

$$\int_K \Pi_2^\nabla v_h \, \mathrm{d}\boldsymbol{x} = \int_K v_h \, \mathrm{d}\boldsymbol{x} = \int_K \Pi_2 v_h \, \mathrm{d}\boldsymbol{x}.$$

Moreover, the definition of $V_2(K)$ gives that

$$\left(\Pi_2^\nabla v_h, m\right)_{0,K} = (v, m)_{0,K} = \left(\Pi_2 v_h, m\right)_{0,K} \quad \forall m \in \mathcal{M}_1^*(K) \cup \mathcal{M}_2^*(K),$$

so that the first three moments of $\Pi_2^\nabla v_h$ and $\Pi_2 v_h$ coincide. Hence, $\Pi_2^\nabla v_h = \Pi_2 v_h$ for all function v_h in $V_2(K)$. □

Remark 2.26. Note that the proof relies on the fact that $\left(v_h - \Pi^\nabla v_h, m\right)_{0,K} = 0$ for all $m \in \mathcal{M}_{k-1}^*(K) \cup \mathcal{M}_k^*(K)$, which is a property for all functions $v_h \in V_k(K)$. This property does not hold in general for functions in the function space $U_k(K)$ from Remark 2.18. Thus, if we use function space $U_k(K)$ we are not be able to calculate the L^2-projection from the degrees of freedom. ◦

Before we investigate the virtual element spaces and projectors in 3D, we take a look at the the basis functions of $U_1(K)$ from Remark 2.18 for the polygon in Figure 2.1. They are similar to the basis functions of $V_1(K)$, and are shown in Figure 2.3.

2.4.2 The three-dimensional case

It is now time to leave Flatland, and ascend into three dimensions. We start by recalling the dimension of the space of polynomials:

$$n_k \coloneqq \dim \mathbb{P}_k(\mathbb{R}^3) = \frac{1}{6}(k+1)(k+2)(k+3).$$

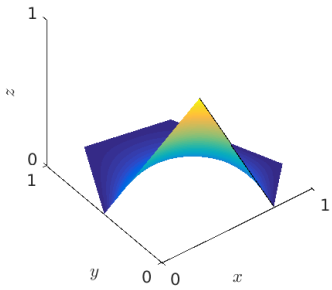
Constructing the local 3D virtual element space

Consider a polyhedron K with n_V vertices, n_E edges, and n_F faces. An example polyhedron is shown in Figure 2.4.

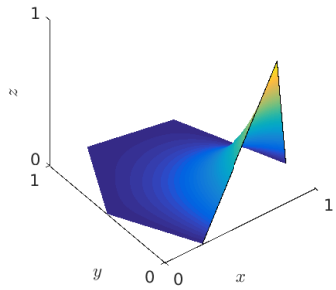
For every $k \geq 1$, and every face F of K, we have from the 2D discussion that $B_k(\partial F)$, defined as in (2.11), is a linear space of dimension $\nu^F k$, where ν^F is the number of edges of face F. Furthermore, we define $V_k(F)$ in the same way as in (2.18). This leads to the natural extension of $V_k(F)$:

$$V_k(\partial K) \coloneqq \{v_h \in C^0(\partial K) : v_h|_F \in V_k(F) \quad \forall F \subset \partial K\}. \tag{2.21}$$

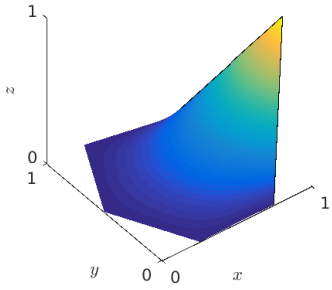
We continue by defining the 3D version of the projection operator $\Pi_k^{\nabla,K}$:



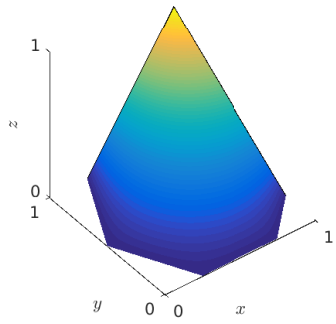
(a) Basis function ϕ^1.



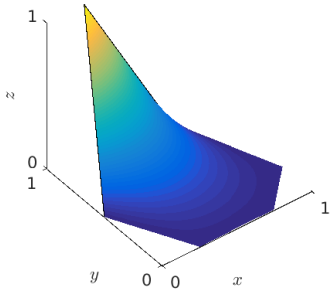
(b) Basis function ϕ^2.



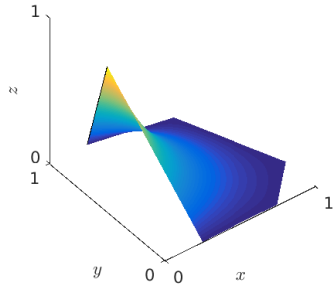
(c) Basisfunction ϕ^3.



(d) Basisfunction ϕ^4.



(e) Basisfunction ϕ^5.



(f) Basisfunction ϕ^6.

Figure 2.3: The 2D basis functions for the polygon in Figure 2.1.

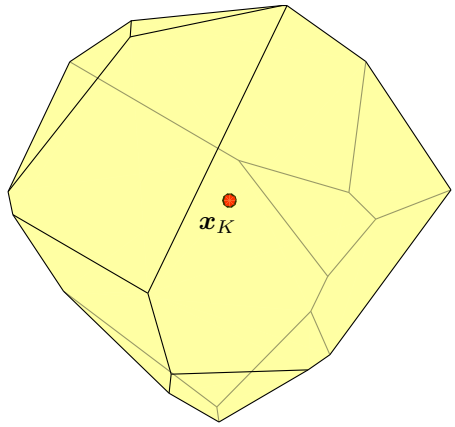


Figure 2.4: Example polyhedron, with centroid $\boldsymbol{x}_K$.

Definition 2.27. Consider a polyhedron K. For $k \geq 1$, we define the projection operator $\Pi_k^{\nabla,K} : H^1(K) \to \mathbb{P}_k(K)$ such that for a function $v \in H^1(K)$, $\Pi_k^{\nabla,K} v$ is the unique element in $\mathbb{P}_k(K)$ satisfying

$$a^K\left(v - \Pi_k^{\nabla,K} v, p\right) = 0 \quad \forall p \in \mathbb{P}_k(K),$$

and

$$\int_{\partial K} \left(\Pi_k^{\nabla,\partial K} v - \Pi_k^{\nabla,K} v\right) \mathrm{d}s = 0 \quad \text{for } k = 1,$$
$$\int_K \left(v - \Pi_k^{\nabla,K} v\right) \mathrm{d}\boldsymbol{x} = 0 \quad \text{for } k \geq 2.$$

∘

Again, whenever the degree k and domain K is clear from the context, we write Π^∇. Here, $\Pi^{\nabla,\partial K}$ is understood to be the projection operator such that for a face F of K, $\Pi^{\nabla,\partial K}|_F = \Pi^{\nabla,F}$. The latter term is the 2D operator from Definition 2.10. The well-posedness of Π^∇ follows in the same manner as in the proof of Theorem 2.11.

Remark 2.28. From Remark 2.12, we know that $\Pi^{\nabla,\partial K}$ is linear on each face of K. Thus, arguing as in $\mathbb{R}^2$, it is clear that Π^∇ is linear in $\mathbb{R}^3$ as well. ∘

We now arrive at the following definition:

Definition 2.29. For $k \geq 1$, we define the three-dimensional local virtual element space $\mathbb{V}_k(K)$ on a polyhedron K:

$$\begin{aligned}\mathbb{V}_k(K) \coloneqq \big\{ v_h \in & H^1(K) : v_h|_{\partial K} \in V_k(\partial K),\ \Delta v_h|_K \in \mathbb{P}_k(K), \\ & \left(v_h - \Pi_k^\nabla v_h, m\right)_{0,K} = 0 \quad \forall m \in \mathcal{M}_{k-1}^*(K) \cup \mathcal{M}_k^*(K) \big\},\end{aligned} \tag{2.22}$$

where $V_k(\partial K)$ is defined as in (2.21). ◦

We equip $\mathbb{V}_k(K)$ with the following degrees of freedom:

Definition 2.30. For a function v_h in the local virtual element space $\mathbb{V}_k(K)$, we choose the following degrees of freedom:

- ♢ $\mathcal{V}^K$: The values of v_h at the vertices of K,
- ♢ $\mathcal{E}^K$: For $k \geq 2$, the moments on each edge E of K

$$|E|^{-1}(v_h, m)_{0,E} \quad \forall m \in \mathcal{M}_{k-2}(E),$$

- ♢ $\mathcal{F}^K$: For $k \geq 2$, the moments on each face F of K

$$|F|^{-1}(v_h, m)_{0,F} \quad \forall m \in \mathcal{M}_{k-2}(F),$$

- ♢ $\mathcal{P}^K$: For $k \geq 2$, the moments

$$|K|^{-1}(v_h, m)_{0,K} \quad \forall m \in \mathcal{M}_{k-2}(K).$$

◦

In Figure 2.5, we have indicated the degrees of freedom of a second order VEM for the polyhedron in Figure 2.4.

Remark 2.31. As in 2D, for each edge E, since the dimensions of $\mathbb{P}_k(E) = k+1$, we have $k-1$ degrees of freedom for each of the n_E edges. Hence, we are free to use the alternative definition of $\mathcal{E}^K$ to be the values at $k-1$ points on each edge in 3D as well. ◦

We now have $n_V + n_E \dim \mathbb{P}_{k-2}(E) + n_F \dim \mathbb{P}_{k-2}(F) + \dim \mathbb{P}_{k-2}(K)$ degrees of freedom, which we write as

$$N^K \coloneqq n_V + n_E(k-1) + \frac{1}{2} n_F k(k-1) + \frac{1}{6} k(k^2-1).$$

As in two dimensions, we have the following proposition:

Proposition 2.32. *Any function $v_h \in \mathbb{V}_k(K)$ is uniquely determined from the degrees of freedom $\mathcal{V}^K$, $\mathcal{E}^K$, $\mathcal{F}^K$ and $\mathcal{P}^K$.*

The proof is similar to that of Proposition 2.16, see [33, 3].

Remark 2.33. Again, similar to what we found in 2D, we have that for any $v_h \in \mathbb{V}_k(K)$, we can compute its projection $\Pi^\nabla v_h$ using the degrees of freedom $\mathcal{V}^K$, $\mathcal{E}^K$ $\mathcal{F}^K$ and $\mathcal{P}^K$. See Theorem 4.9 and [3]. ◦

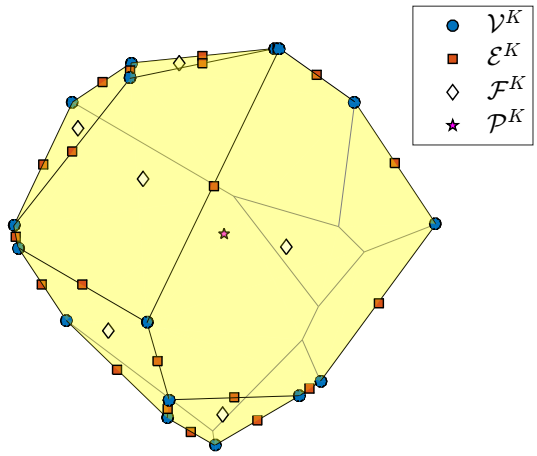


Figure 2.5: The degrees of freedom of $\mathbb{V}_2(K)$.

Remark 2.34. Similar to Remark 2.18, we could have chosen the virtual element space

$$\mathbb{U}_k(K) = \{v_h \in H^1(K) : v_h|_{\partial K} \in U_k(\partial K),\, \Delta v_h|_K \in \mathbb{P}_{k-2}(K)\},$$

which has the same degrees of freedom as $\mathbb{V}_k(K)$. However, with this choice, we are no longer able to calculate $\Pi^\nabla v_h$ for a function in $v_h \in \mathbb{U}_k(K)$ directly from the degrees of freedom. We will come back to this point in Section 2.5. ◦

Projection Error

We denote the diameter of a polygonal face F of a polyhedron K by h_F, and make the following geometric assumptions on our grid $\mathcal{T}_h$:

Assumption 2.35. There exists a positive, real number γ, such that

(i) for every element $K \in \mathcal{T}_h$, for every face F of K, and for every edge E of F

$$h_E \geq \gamma h_F \geq \gamma^2 h_K;$$

(ii) every element K is star-shaped with respect to all points in a ball of radius greater than or equal to γh_K;

(iii) every face F is star-shaped with respect to all points in a disk of radius greater than or equal to γh_F.

◦

Under this assumption, it can be shown that Proposition 2.20 also holds in three dimensions. See [3] for details.

Constructing the global 3D virtual element space

As in two dimensions, we can now proceed to define the global virtual element space:

Definition 2.36. For every decomposition $\mathcal{T}_h$ of Ω into polyhedral cells K, for every $k \geq 1$, we define the global virtual element space $\mathbb{V}_h$:

$$\mathbb{V}_h \coloneqq \left\{ v_h \in V : v_h|_K \in \mathbb{V}_k(K) \quad \forall K \in \mathcal{T}_h \right\},$$

where $\mathbb{V}_k(K)$ is defined as in (2.22). ◦

Similar to the 2D case, we have that in order to uniquely determine a function $v_h \in \mathbb{V}_h$, we need

$$N \coloneqq N_V + N_E(k-1) + N_F \frac{1}{2}k(k-1) + N_P \frac{1}{6}k(k^2-1)$$

degrees of freedom, where N_V, N_E, N_F, and N_P are the total number of internal vertices, internal edges, internal faces, and polyhedra in $\mathcal{T}_h$. We define these as follows:

Definition 2.37. For a function v_h in the global virtual element space $\mathbb{V}_h$, we choose the following N degrees of freedom:

♢ $\mathcal{V}$: The values of v_h at the internal vertices,

♢ $\mathcal{E}$: For $k \geq 2$, the moments on each internal edge E of $\mathcal{T}_h$

$$|E|^{-1}(v_h, m)_{0,E} \quad \forall m \in \mathcal{M}_{k-2}(E),$$

♢ $\mathcal{F}$: For $k \geq 2$, the moments on each internal face F of $\mathcal{T}_h$

$$|F|^{-1}(v_h, m)_{0,F} \quad \forall m \in \mathcal{M}_{k-2}(F),$$

♢ $\mathcal{P}$: For $k \geq 2$, the moments on each polyhedron K of $\mathcal{T}_h$

$$|K|^{-1}(v_h, m)_{0,K} \quad \forall m \in \mathcal{M}_{k-2}(K).$$

◦

From Proposition 2.32 , it follows that a function $v_h \in \mathbb{V}_h$ can be uniquely determined from the degrees of freedom $\mathcal{V}$, $\mathcal{E}$, $\mathcal{F}$ and $\mathcal{P}$.

Interpolation error

Following the 2D procedure, we define the functionals χ^i, $i = 1, \ldots, N$ just as in Definition 2.23, but with $\mathbb{V}_h$ in place of V_h. Under Assumption 2.35, it

can be shown that Proposition 2.24, with $\mathbb{V}_h$ in place of V_h, also holds in three dimensions. See [3] for details.

Computing the L^2-projection

We recall that in order to compute the L^2-projection $\Pi_k v_h$ for a function $v_h \in \mathbb{V}_k(K)$, we need to calculate $(v_h, m)_{0,K}$ for all monomials $m \in \mathcal{M}_k(K)$. As in 2D, the first n_{k-2} of these moments are given from $\mathcal{P}^K$, while the last ones are given from the definition of $\mathbb{V}_k(K)$, since

$$(v_h, m)_{0,K} = \left(\Pi_k^\nabla v_h, m\right)_{0,K} \quad \forall m \in \mathcal{M}_k^*(K) \cup \mathcal{M}_{k-1}^*(K).$$

Hence, assuming that we are able to compute $\Pi_k^\nabla v_h$, we conclude that Proposition 2.25 holds in 3D as well.

Remark 2.38. Again, we rely on that $(v_h, m)_{0,K} = \left(\Pi^\nabla v_h, m\right)_{0,K}$ for all monomials $m \in \mathcal{M}_{k-1}^*(K) \cup \mathcal{M}_k^*(K)$. Hence, using the alternative VEM space $\mathbb{U}_k(K)$ from Remark 2.34, we would not be able to calculate $\Pi_k v_h$ directly from the degrees of freedom. ∘

This concludes the construction of our VEM function spaces. Approximations of selected basis functions of $\mathbb{U}_1(K)$ for the polyhedron in Figure 2.4 are shown in Figure 2.6

2.5 Constructing the bilinear form

In what follows, we will denote the local virtual element space by $V_k(K)$, both in two and three dimensions.

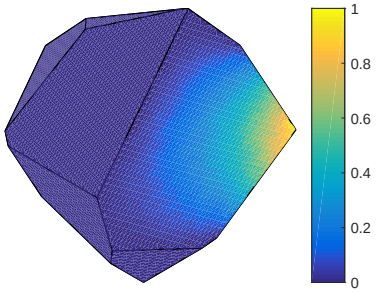
Let us now justify the choice of degrees of freedom in $V_k(K)$. As mentioned in the beginning of this chapter, we want to be able to calculate exactly the stiffness bilinear form a^K whenever one of the two functions is in $\mathbb{P}_k(K)$. Note that this is also necessary in order to compute $\Pi^\nabla v$ for a function $v \in H^1(K)$. To this end, let $p \in \mathbb{P}_k(K)$ and $v_h \in V_k(K)$. By Greens formula, we then have

$$a^K(v_h, p) = (\nabla v_h, \nabla p)_{0,K} = (v_h, \partial_n p)_{0,\partial K} - (v_h, \Delta p)_{0,K}. \tag{2.23}$$

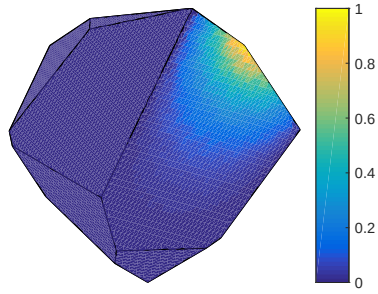
In two dimension, we see that $\partial_n p|_E \in \mathbb{P}_{k-1}(E)$ and $v_h|_E \in \mathbb{P}_k(E)$ for all edges E of ∂K. Hence, assuming we are able to compute integrals of polynomials over ∂K, the first term can be computed exactly from the degrees of freedom $\mathcal{V}^K$ and $\mathcal{E}^K$. Further, we note that $\Delta p \in \mathbb{P}_{k-2}(K)$, so that $\mathcal{P}^K$ gives us the exact value of the second term.

In three dimensions, the second term is again computable from $\mathcal{P}^K$. For the first term, we know that the degrees of freedom $\mathcal{F}^K$ enables us to compute $(v_h, \partial_n p)_{0,\partial K}$ when $\partial_n p|_F \in \mathbb{P}_{k-2}(F)$. But $\partial_n p|_F$ belongs, in general, to $\mathbb{P}_{k-1}(F)$. However, since $v_h|_{\partial K} \in V_k(\partial K)$, we know from Definition 2.13 that for a face F of K, we have

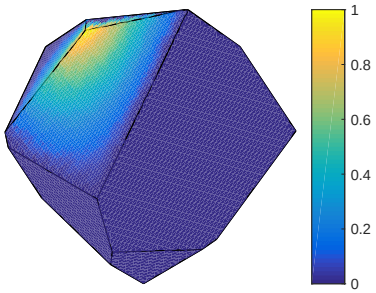
$$(v_h, m)_{0,F} = \left(\Pi_k^{\nabla,F} v_h, m\right)_{0,F} \quad \forall m \in \mathcal{M}_k^*(F) \cup \mathcal{M}_{k-1}^*(F).$$



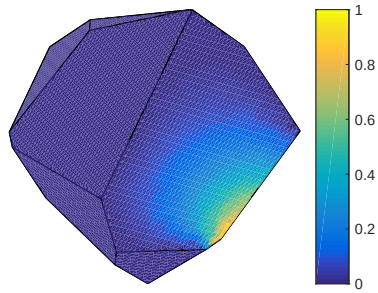
(a) Basis function ϕ^1.



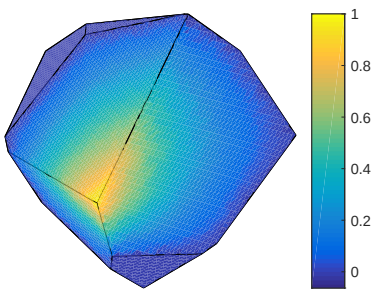
(b) Basis function ϕ^2.



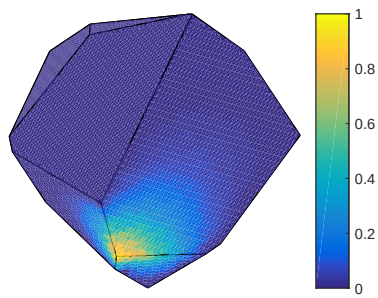
(c) Basisfunction ϕ^4.



(d) Basisfunction ϕ^9.



(e) Basisfunction ϕ^{11}.



(f) Basisfunction ϕ^{13}.

Figure 2.6: Some of the 3D basis functions for the polyhedron in Figure 2.4.

Hence, is follows that if p is of degree k, we have

$$(v_h, \partial_n p)_{0,\partial K} = \left(\Pi_k^{\nabla,\partial K} v_h, \partial_n p\right)_{0,\partial K},$$

and conclude we are able to compute both terms in (2.23) in three dimensions as well.

We now see that with the given construction of the virtual element space, we are able to calculate $a^K(v_h, p)$ for all functions $v_h \in V_k(K)$ and all polynomials $p \in \mathbb{P}_k(K)$ exactly, with explicit knowledge *only of the degrees of freedom* of v_h. From Remark 2.26, it is clear that this would not have been possible in $\mathbb{R}^3$ if we instead used the local VEM space $\mathbb{U}_k(K)$.

We note that for two functions $u, v \in H^1(K)$, we have the following:

$$\begin{aligned} &a^K\left(\Pi^\nabla u, \Pi^\nabla v\right) + a^K\left(u - \Pi^\nabla u, v - \Pi^\nabla v\right) \\ &\qquad = 2a^K\left(\Pi^\nabla u, \Pi^\nabla v\right) + a^K(u, v) - a^K\left(u, \Pi^\nabla v\right) - a^K\left(\Pi^\nabla u, v\right). \end{aligned}$$

Using the fact that $a^K\left(u, \Pi^\nabla v\right) = a^K\left(\Pi^\nabla u, \Pi^\nabla v\right)$, this reduces to the identity

$$a^K(u, v) = a^K\left(\Pi^\nabla u, \Pi^\nabla v\right) + a^K\left(u - \Pi^\nabla u, v - \Pi^\nabla v\right). \tag{2.24}$$

From Assumption 2.5, we recall that we want our bilinear form a_h^K to satisfy k-consistency (2.8) and stability (2.9). To this end, we mimic the identity (2.24), and choose a_h^K to be on the form

$$a_h^K(u, v) = a^K\left(\Pi^\nabla u, \Pi^\nabla v\right) + s^K\left(u - \Pi^\nabla u, v - \Pi^\nabla v\right),$$

where s^K is a symmetric bilinear form, which we refer to as the *stability term*. Note that any choice of s^K would ensure k-consistency, since, by the definition of Π^∇,

$$\begin{aligned} a_h^K(v_h, p) &= a^K(\Pi^\nabla v_h, \Pi^\nabla p) + s^K\left(v_h - \Pi^\nabla v_h, p - \Pi^\nabla p\right) \\ &= a^K(v_h, p) \quad \forall p \in \mathbb{P}_k(K),\, v_h \in V_k(K). \end{aligned}$$

However, in general, the stability requirement is not satisfied. Indeed; consider $s^K = 0$ and $v_h \in V_k(K)$ a non-constant function such that $\Pi^\nabla v_h = 0$. Then $a_h^K(v_h, v_h) = 0$, but $a^K(v_h, v_h) = |v_h|_{1,K}^2 > 0$. Hence, we need s^K to be positive definite on $\ker \Pi^\nabla$. In fact, we will require it to satisfy

$$\tilde{c}_1 a^K(v_h, v_h) \leq s^K(v_h, v_h) \leq \tilde{c}_2 a^K(v_h, v_h) \quad \forall v_h \in \ker \Pi^\nabla|_{V_k(K)}, \tag{2.25}$$

where $\tilde{c}_1$ and $\tilde{c}_2$ are two positive constants, independent of K and h_K. This leads to the following theorem:

Theorem 2.39. *Let s^K be a symmetric bilinear form satisfying (2.25). Then, the bilinear form*

$$a_h^K(u, v) = a^K\left(\Pi^\nabla u, \Pi^\nabla v\right) + s^K\left(u - \Pi^\nabla u, v - \Pi^\nabla v\right) \tag{2.26}$$

satisfies k-consistency (2.8) *and stability* (2.9).

See [17, 33] for a proof. Before we proceed, we note the following:

Remark 2.40. We have that $\ker \Pi^\nabla = \operatorname{Im}\left(\mathrm{Id} - \Pi^\nabla\right)$, where Id is the identity map. Indeed; if $u \in \ker \Pi^\nabla$, then $u = u - \Pi^\nabla u$. Hence, $u \in \operatorname{Im}\left(\mathrm{Id} - \Pi^\nabla\right)$, so that $\ker \Pi^\nabla \subset \operatorname{Im}\left(\mathrm{Id} - \Pi^\nabla\right)$. On the other hand, since Π^∇ is a projection, we have that $\Pi^\nabla\left(\mathrm{Id} - \Pi^\nabla\right) = 0$, and it follows that $\operatorname{Im}\left(\mathrm{Id} - \Pi^\nabla\right) \subset \ker \Pi^\nabla$. ◦

In order to complete the construction of a_h^K, we define the following set of basis functions for $V_k(K)$:

Definition 2.41. Let K be a polygon in $\mathbb{R}^2$, or a polyhedron in $\mathbb{R}^3$. The canonical basis for $V_k(K)$ is defined as the unique set of functions $\{\phi^i\}_{i=1}^{N^K}$ such that

$$\chi^i(\phi^j) = \delta_{ij},$$

where χ^i are the functionals defined in Definition 2.23, and δ_{ij} is the Kronecker delta. ◦

Remark 2.42. We have the following interpolation identity:

$$v_h = \sum_{i=1}^{N^K} \chi^i(v_h)\phi^i \quad \forall v_h \in V_k(K). \tag{2.27}$$

To see this, let $\xi_1, \ldots, \xi_{N^K}$ be the real coefficients such that $v_h = \sum_{j=1}^{N^K} \xi_j \phi^j$. Since the functionals χ^i are linear, we get

$$\chi^i(v_h) = \chi^i\left(\sum_{j=1}^{N^K} \xi_j \phi^j\right) = \sum_{j=1}^{N^K} \xi_j \chi^i(\phi^j) = \xi_i,$$

and (2.27) follows. We will use this interpolation identity extensively in the following chapters. ◦

Using the canonical basis, it is clear that the discrete problem (2.7) for a cell K can be formulated in the following way:

For $i = 1, \ldots, N^K$, find

$$(\xi_1, \ldots \xi_{N^K})^T \in \mathbb{R}^{N^K} \text{ such that } a_h^K\left(\phi^i, \sum_{j=1}^{N^K} \xi_j \phi^j\right) = \langle f_h, \phi^i \rangle_K.$$

This can be written compactly in matrix form as

$$\boldsymbol{A}^K \boldsymbol{\xi} = \boldsymbol{b}^K,$$

where the local stiffness matrix is given by

$$\boldsymbol{A}_{i,j}^K = a_h^K(\phi^i, \phi^j) = a^K\left(\Pi^\nabla \phi^i, \Pi^\nabla \phi^j\right) + s^K\left(\phi^i - \Pi^\nabla \phi^i, \phi^j - \Pi^\nabla \phi^j\right),$$

while the local load vector is given by $\boldsymbol{b}_i^K = \langle f_h, \phi^i \rangle_K$, and $\boldsymbol{\xi}$ is the vector $(\xi_1, \ldots, \xi_{N^K})^T$. Following Remark 2.42, we see that the values $\xi_1, \ldots, \xi_{N^K}$ we are approximating in the virtual element method are the degrees of freedom $\chi^1(u_h), \ldots, \chi^{N^K}(u_h)$ of the solution u_h to the discrete problem (2.7).

2.6 Construction of the right-hand side

The only thing missing in order to have a complete description of VEM is the right-hand side of (2.7). We will assume that $f \in H^k(\Omega)$.

Definition 2.43. We define the element $f_h \in V_h'$ from Assumption 2.4 to be such that $f_h|_K = \Pi_{k-1}^K f$. ◦

That is, we approximate f by a piecewise polynomial that, for each cell K, is the L^2-orthogonal projection onto $\mathbb{P}_{k-1}(K)$. Consequently, for a function $v_h \in V_k(K)$, we have

$$\langle f_h, v_h \rangle_K = \left(\Pi_{k-1} f, v_h\right)_{0,K},$$

which we are able to compute using the degrees of freedom $\mathcal{P}^K$, along with the fact that

$$\left(\Pi_{k-1} f, v_h\right)_{0,K} = \left(\Pi_{k-1} f, \Pi_k^\nabla v_h\right)_{0,K}.$$

Moreover, we have from Definition 2.9 of the L^2 projection that

$$\left(f - \Pi_{k-1} f, \Pi_0 v_h\right)_{0,K} = 0.$$

Hence, we can write

$$\left(f - \Pi_{k-1} f, v_h\right)_{0,K} = \left(f - \Pi_{k-1} f, v_h - \Pi_0 v_h\right)_{0,K}.$$

Using the theory of approximation of functions in Sobolev spaces (see for instance [13, 5]), we can show that

$$\begin{aligned}\left(f - \Pi_{k-1} f, v_h - \Pi_0 v_h\right)_{0,K} &\leq \|f - \Pi_{k-1} f\|_{0,K} \|v_h - \Pi_0 v_h\|_{0,K} \\ &\leq C h_K^k |f|_{k,K} h_K |v_h|_{1,K},\end{aligned}$$

where C is a positive constant, independent on h_K and K. Thus, we have that

$$\begin{aligned}\langle f - f_h, v_h \rangle &\leq C \sum_{K \in \mathcal{T}_h} h_K^{k+1} |f|_{k,K} |v_h|_{1,K} \\ &\leq C h^{k+1} \left(\sum_{K \in \mathcal{T}_h} |f|_{k,K}^2 \right)^{1/2} \left(\sum_{K \in \mathcal{T}_h} |v_h|_{1,K}^2 \right)^{1/2} \\ &= C h^{k+1} |f|_{k,\Omega} |v_h|_{1,\Omega},\end{aligned}$$

from which we conclude that

$$\|f - f_h\|_{V_h'} \leq Ch^{k+1}\|f\|_{k,\Omega}.$$

2.7 L^2 error estimate

We now have the following theorem:

Theorem 2.44. *Let u be the solution to the weak formulation (2.3), and u_h be the solution to the discrete problem (2.7), with $f_h = \Pi_{k-1}f$ for $k \geq 1$. Assume further that Ω is convex, that the right-hand side f belongs to $H^k(\Omega)$, and that the exact solution u belongs to $H^{k+1}(\Omega)$. Then, the following estimate holds:*

$$\|u - u_h\|_{0,\Omega} + h\|u - u_h\|_{1,\Omega} \leq Ch^{k+1}|u|_{k+1,\Omega},$$

where C is a constant independent of h.

The proof is based on a duality argument, and is given in [3], wherein it is also argued that for $k \geq 3$, we can take $f_h = \Pi_{k-2}f$ and still have optimal estimates. Note that with this choice, we are able to compute the right-hand side using only the degrees of freedom $\mathcal{P}^K$.

2.8 Final Remarks

We note the following: For any function $v_h \in V_k(K)$, the L^2 projection $\Pi_{k-2}v_h$ is computable from the degrees of freedom $\mathcal{P}^K$. Hence, as long as we are in $\mathbb{R}^2$, if we instead choose $f_h = \Pi_{k-2}f$ for $k \geq 2$, we never actually have to use fact that $\left(v_h - \Pi_k^\nabla v_h, m\right)_{0,K} = 0$ for all $m \in \mathcal{M}_{k-1}^*(K) \cup \mathcal{M}_k^*(K)$. As a matter of fact, if we are satisfied with a convergence rate of k for $k = 2$, we could have done just fine with the local virtual element space $U_k(K)$, as mentioned in Remark 2.18. This was the original local virtual element space, proposed in [33]. The virtual element method presented here is also called the *modified VEM* [3].

The reason for using $V_k(K)$ instead of $U_k(K)$ is that it provides an explicit knowledge of the L^2 projection Π. For our model problem in three dimensions, this has the obvious advantage that it gives an easy way of computing the last term of (2.23) exactly. Indeed; In order to calculate this term using $U_k(K)$, one would have to add k additional degrees of freedom on each face. This is the approach in higher order mimetic finite differences [35]. In two dimensions, if the elliptic operator has only the principal part (that is, only the highest order derivative term), it is sufficient to compute Π^∇ in order to construct the discretization. However, in many cases, explicit knowledge of the L^2 projection allows for much cheaper discretizations. Moreover, in some cases, (like nonlinear problems) knowledge of this projection might substantially improve the quality of the method [3]. In fact, as long as we are in $\mathbb{R}^2$ and we are not computing the L^2 projection, there is no practical difference between choosing $U_k(K)$ or $V_k(K)$, since they share the same degrees of freedom.

Finally, we recall our model problem (2.1). In general, the permeability $\boldsymbol{K}$ is not isotropic, and can vary from between the cells of the grid. The elliptic operator of interest is then on the form $\nabla \cdot \boldsymbol{K}\nabla$. Thus, in order to solve (2.1) for a general permeability, we must consider the local stiffness bilinear form $a^K(u,v) = (\boldsymbol{K}\nabla u, \nabla v)_{0,K}$, and define the projection operator $\Pi^{\boldsymbol{K}\nabla}$ accordingly.

Chapter 3

A Family of Bilinear Forms

With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.

John von Neumann

As we saw in the previous chapter, we are free to choose the stability term as long as the local bilinear form satisfies k-consistency and stability. In this chapter, we present a method for constructing the stability term, and show how it can be chosen to make a_h^K equal to the exact bilinear form a^K. Finally, we use this to see how the stability term can be chosen to make VEM equivalent to other methods.

This chapter consists mainly of original work.

3.1 Constructing the stability term

For two functions $u, v \in H^1(K)$, we recall that our local stiffness bilinear form is

$$a_h^K(u,v) = a^K\left(\Pi^\nabla u, \Pi^\nabla v\right) + s^K\left(u - \Pi^\nabla u, v - \Pi^\nabla v\right).$$

In Section 2.5, we saw that in order to have a stiffness bilinear form satisfying both k-consistency and stability, we need the stability term s^K to be a symmetric bilinear form satisfying

$$\tilde{c}_1 a^K(v_h, v_h) \leq s^K(v_h, v_h) \leq \tilde{c}_2 a^K(v_h, v_h) \quad \forall v_h \in \ker \Pi^\nabla\big|_{V_k(K)}, \tag{3.1}$$

where $\tilde{c}_1$ and $\tilde{c}_2$ are two positive constants, independent of K and h_K. For the moment, we will relax this requirement so that $\tilde{c}_1$ and $\tilde{c}_2$ are two positive constants, which may depend on K and h_K. This will still ensure that the virtual element method produces a solution that is close to the exact solution in the sense that Theorem 2.6 still holds, with the constant C depending on K and h_K. However, we no longer have the same convergence properties. Throughout this chapter, we will sometimes abuse notation and write Π^∇, meaning $\Pi^\nabla\big|_{V_k(K)}$.

We denote the vector of degrees of freedom of a function $v \in H^1(K)$ by

$$\hat{\boldsymbol{v}} = \left(\chi^1(v), \dots, \chi^{N^K}(v)\right)^T. \tag{3.2}$$

Let u_h and v_h be two functions in $V_k(K)$. From Remark 2.42, we know that we can express these in the canonical VEM basis as $u_h = \sum_i \chi^i(u_h)\phi^i$ and $v_h = \sum_i \chi^i(v_h)\phi^i$. Hence, we can write our stability term as

$$s^K(u_h, v_h) = \sum_{i,j=1}^{N^K} \chi^i(u_h) s^K(\phi^i, \phi^j) \chi^j(v_h).$$

Defining the $N^K \times N^K$ matrix $\boldsymbol{S}$ such that $\boldsymbol{S}_{i,j} = s^K(\phi^i, \phi^j)$, we have the matrix representation of s^K:

$$s^K(u_h, v_h) = \hat{\boldsymbol{u}}_h^T \boldsymbol{S} \hat{\boldsymbol{v}}_h. \tag{3.3}$$

The requirement of s^K to be symmetric positive definite on $\ker \Pi^\nabla$ means that $\boldsymbol{S}$ must be a symmetric and positive definite matrix for all vectors $\hat{\boldsymbol{v}}_h$ such that $v_h \in \ker \Pi^\nabla$. We will simply refer to this as $\boldsymbol{S}$ being symmetric positive definite on $\ker \Pi^\nabla$. Moreover, we need $\boldsymbol{S}$ to be such that (3.1) is satisfied.

At first sight, choosing the stability term involves choosing the $N^K(N^K + 1)/2$ elements on and above the diagonal of $\boldsymbol{S}$. However, we will only use s^K on $\operatorname{Im}\left(\mathrm{Id} - \Pi^\nabla\right)$, which we know from Remark 2.40 equals $\ker \Pi^\nabla$. Hence, as we will see, we do not have to determine that many parameters. To this end, let $\{\psi^i\}_i$ be a basis for $\ker \Pi^\nabla$ in $V_k(K)$. We note that the restriction of Π^∇ to $V_k(K)$ is a projection from the N^K-dimensional function space $V_k(K)$ to the n_k-dimensional function space $\mathbb{P}_k(K)$, so that the dimension of $\ker \Pi^\nabla$ is $n_{\ker} \coloneqq N^K - n_k$. We now have the following result:

Lemma 3.1. *Let $\{\psi^i\}_i$ be a basis for $\ker \Pi^\nabla$, and let $\boldsymbol{Q} = [\hat{\boldsymbol{\psi}}^1, \dots, \hat{\boldsymbol{\psi}}^{n_{\ker}}]$ be its matrix representation. Similarly, let $\{p^i\}_i$ be a basis for $\left(\ker \Pi^\nabla\right)^\perp$, and let $\boldsymbol{R} = [\hat{\boldsymbol{p}}^1, \dots, \hat{\boldsymbol{p}}^{N^K - n_{\ker}}]$, so that $\boldsymbol{Q}^T\boldsymbol{R} = 0$. Moreover, let $\boldsymbol{S}$ be an $N^K \times N^K$ symmetric matrix, positive definite on $\ker \Pi^\nabla$, and let $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma}'$ and $\boldsymbol{\Sigma}''$ be the matrices defined by*

$$[\boldsymbol{Q}, \boldsymbol{R}] \begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{\Sigma}' \\ \boldsymbol{\Sigma}' & \boldsymbol{\Sigma}'' \end{bmatrix} \begin{bmatrix} \boldsymbol{Q}^T \\ \boldsymbol{R}^T \end{bmatrix} = \boldsymbol{S}. \tag{3.4}$$

Then, we have that

$$\hat{\boldsymbol{u}}_h^T \boldsymbol{S} \hat{\boldsymbol{v}}_h = \hat{\boldsymbol{u}}_h^T \boldsymbol{Q} \boldsymbol{\Sigma} \boldsymbol{Q}^T \hat{\boldsymbol{v}}_h \quad \forall u_h, v_h \in \ker \Pi^\nabla.$$

Note that the matrices $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma}'$ and $\boldsymbol{\Sigma}''$ are uniquely defined by (3.4), since $[\boldsymbol{Q}, \boldsymbol{R}]$ is an invertible matrix.

Proof. Let u_h and v_h be two functions in $\ker \Pi^\nabla$. We express them in the basis $\{\psi^i\}_i$ as

$$u_h = \sum_{i=1}^{n_{\ker}} \xi_i \psi^i, \quad v_h = \sum_{i=1}^{n_{\ker}} \eta_i \psi^i,$$

where $\xi_i, \eta_i \in \mathbb{R}$ for all i. From the linearity of χ^j, we have that $\hat{\boldsymbol{u}}_h = \sum_i \xi_i \hat{\boldsymbol{\psi}}^i$. We can write this in matrix form as $\hat{\boldsymbol{u}}_h = \boldsymbol{Q}\boldsymbol{\xi}$, where $\boldsymbol{\xi} = (\xi_1, \dots, \xi_{n_{\text{ker}}})^T$. Analogously, we write $\hat{\boldsymbol{v}}_h = \boldsymbol{Q}\boldsymbol{\eta}$. We now have that

$$\hat{\boldsymbol{u}}_h^T \boldsymbol{S} \hat{\boldsymbol{v}}_h = \boldsymbol{\xi}^T \boldsymbol{Q}^T \boldsymbol{S} \boldsymbol{Q} \boldsymbol{\eta}.$$

Inserting the expression (3.4) for $\boldsymbol{S}$ yields

$$\hat{\boldsymbol{u}}_h^T \boldsymbol{S} \hat{\boldsymbol{v}}_h = \boldsymbol{\xi}^T \boldsymbol{Q}^T \left(\boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T + \boldsymbol{R}\boldsymbol{\Sigma}'\boldsymbol{Q}^T + \boldsymbol{Q}\boldsymbol{\Sigma}'\boldsymbol{R}^T + \boldsymbol{R}\boldsymbol{\Sigma}''\boldsymbol{R}^T \right) \boldsymbol{Q}\boldsymbol{\eta}.$$

After multiplying out the parentheses, all terms involving $\boldsymbol{R}$ will vanish due to the orthogonality $\boldsymbol{Q}^T\boldsymbol{R} = 0$, and we are left with

$$\hat{\boldsymbol{u}}_h^T \boldsymbol{S} \hat{\boldsymbol{v}}_h = \boldsymbol{\xi}^T \boldsymbol{Q}^T \boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T \boldsymbol{Q}\boldsymbol{\eta} = \hat{\boldsymbol{u}}_h^T \boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T \hat{\boldsymbol{v}}_h.$$

□

Note that $\boldsymbol{\Sigma}$ must be symmetric positive definite: First of all, we have that

$$\left(\boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T\right)^T = \boldsymbol{Q}\boldsymbol{\Sigma}^T\boldsymbol{Q}^T.$$

Since $\boldsymbol{S}$ is symmetric, we must have that $\boldsymbol{\Sigma}^T = \boldsymbol{\Sigma}$. Moreover, for a function $v_h \in \ker \Pi^\nabla$, with $\hat{\boldsymbol{v}}_h = \boldsymbol{Q}\boldsymbol{\xi}$, we have that

$$s^K(v_h, v_h) = \boldsymbol{\xi}^T \boldsymbol{Q}^T \boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T \boldsymbol{Q}\boldsymbol{\xi}.$$

Now, $(\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{\xi})^T = \boldsymbol{\xi}^T\boldsymbol{Q}^T\boldsymbol{Q}$, so that $s^K(v_h, v_h)$ is on the form $\boldsymbol{y}^T\boldsymbol{\Sigma}\boldsymbol{y}$. Proposition 2.16 and Proposition 2.32 tells us that any function in $V_k(K)$ is uniquely determined from its degrees of freedom. Hence, $\boldsymbol{Q}$ must have full rank, and it follows that $\boldsymbol{Q}^T\boldsymbol{Q}$ is invertible. Thus, for a given vector $\boldsymbol{\xi} \in \mathbb{R}^{n_{\text{ker}}}$, there exists a vector $\boldsymbol{y} \in \mathbb{R}^{n_{\text{ker}}}$ such that $\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{\xi} = \boldsymbol{y}$. Since $\boldsymbol{S}$ is positive definite on $\ker \Pi^\nabla$, this implies that $\boldsymbol{y}^T\boldsymbol{\Sigma}\boldsymbol{y} > 0$ for all vectors $\boldsymbol{y} \neq 0$.

With this result, since we are only using s^K on $\ker \Pi^\nabla$, we can now define a whole *family* of consistent and stable bilinear forms:

Theorem 3.2. *Let K be a polygon or polyhedron, and let $u, v \in H^1(K)$. The family of consistent and stable local bilinear forms a_h^K consists of those that can be written on the form*

$$a_h^K(u, v) = a^K\left(\Pi^\nabla u, \Pi^\nabla v\right) + s^K\left(u - \Pi^\nabla u, v - \Pi^\nabla v\right),$$

where $s^K(u, v) = \hat{\boldsymbol{u}}^T\boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T\hat{\boldsymbol{v}}$. The matrix $\boldsymbol{Q} = [\hat{\boldsymbol{\psi}}^1, \dots, \hat{\boldsymbol{\psi}}^{n_{\text{ker}}}]$ is the matrix representation of $\ker \Pi^\nabla$, and $\boldsymbol{\Sigma}$ is an $n_{\text{ker}} \times n_{\text{ker}}$ symmetric positive definite matrix.

Proof. From Section 2.5, we know that a_h^K satisfies k-consistency (2.8) for any choice of s^K. To see that the stability requirement (3.1) is satisfied, consider a function $v_h \in \ker \Pi^\nabla$, and define $\|v_h\|_{s^K}^2 \coloneqq s^K(v_h, v_h)$. Since s^K is bilinear and symmetric positive definite on $\ker \Pi^\nabla$, $\|\cdot\|_{s^K}$ is a norm on $\ker \Pi^\nabla$. Next, we recall that $a^K(v_h, v_h) = |v_h|_{1,K}^2$, and that $|v_h|_{1,K} > 0$ for all non-constant functions v_h. Moreover, since $\Pi_k^\nabla p = p$ for all $p \in \mathbb{P}_k(K)$, it is clear that no

constant function besides $v_h \equiv 0$ is in $\ker \Pi^\nabla$. We also know that a^K is bilinear and symmetric, and it follows that $|\cdot|_{1,K}$ is also a norm on $\ker \Pi^\nabla$. Now, we notice that the dimension $n_{\ker} = N^K - n_k$ of $\ker \Pi^\nabla$ is finite. Since all norms are equivalent in a finite-dimensional vector space, we thus know that there exists two positive constants $\tilde{c}_1$ and $\tilde{c}_2$ such that

$$\sqrt{\tilde{c}_1}|v_h|_{1,K} \leq \|v_h\|_{s^K} \leq \sqrt{\tilde{c}_2}|v_h|_{1,K} \quad \forall v_h \in \ker \Pi^\nabla.$$

We conclude that (3.1) is satisfied, with the relaxed requirement that $\tilde{c}_1$ and $\tilde{c}_2$ can depend on K and h_K. □

Remark 3.3. Since $\mathbf{\Sigma}$ can be any $n_{\ker} \times n_{\ker}$ symmetric positive definite matrix, we are free to choose $n_{\ker}(n_{\ker}+1)/2$ coefficients, so long as they constitute a positive definite matrix. ◦

We now want to see how s^K must be modified in order for the constants $\tilde{c}_1$ and $\tilde{c}_2$ to be independent on K and h_K. Before we investigate what this means in practice, we need a result saying how the projection operator Π^∇ behaves under transformations in Euclidean space. We start with the following definition:

Definition 3.4. Let K_r and K be two polygons in $\mathbb{R}^2$, or polyhedra in $\mathbb{R}^3$. We say that K is similar to K_r if there exists a mapping G from K_r to K, and a positive constant γ, such that for any two points $\boldsymbol{x}, \boldsymbol{y} \in K_r$,

$$|G(\boldsymbol{x}) - G(\boldsymbol{y})| = \gamma|\boldsymbol{x} - \boldsymbol{y}|.$$

We refer to the mapping G as a similarity mapping. ◦

Remark 3.5. Given two similar elements $K_r, K \subset \mathbb{R}^d$, it can be shown that the similarity mapping G from K_r to K is affine, that is:

$$G(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b},$$

where $\boldsymbol{A}$ is a matrix in $\mathbb{R}^{d\times d}$, and $\boldsymbol{b}$ is a vector in $\mathbb{R}^d$ (see for example [10]). Moreover, for two points $\boldsymbol{x}, \boldsymbol{y} \in K_r$, we have that

$$|G(\boldsymbol{x}) - G(\boldsymbol{y})|^2 = |\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{y})|^2 = \gamma^2|\boldsymbol{x} - \boldsymbol{y}|^2,$$

which we can write as

$$(\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{A}^T \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T \gamma^2 (\boldsymbol{x} - \boldsymbol{y}).$$

Hence, $\boldsymbol{A}^T\boldsymbol{A} = \gamma^2\boldsymbol{I}$, and $\boldsymbol{A}$ is orthogonal. ◦

For an affine mapping G from K_r to K, we have that

$$|K| = \int_K \mathrm{d}\boldsymbol{x} = |\det(DG)| \int_{K_r} \mathrm{d}\boldsymbol{x}_r = |\det(DG)||K_r|,$$

where $\boldsymbol{x}_r = (x_1^r, \ldots, x_d^r)$ are the coordinates on K_r, and DG is the Jacobian matrix, so that $(DG)_{i,j} = \partial G_i/\partial x_j^r$. Since G is affine, we have that $DG = \boldsymbol{A}$,

where $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ is a constant matrix. However, we will stick to the notation DG. Note that we have used that DG is constant, and moved it outside the integral. We see that in order for $|K|$ to be non-zero, we must have $\det(DG) \neq 0$. In other words, G must be invertible. We will always consider elements with nonzero area or volume, so that if the mapping from an element K_r to K is affine, it is also invertible.

We will also consider the restriction of a mapping G to an edge E_r of K_r. In this case, we write $G|_{E_r}$, which is then understood to be a mapping from E_r to its corresponding edge E of K. That is, $G|_{E_r}$ is a mapping from $\mathbb{R}$ to $\mathbb{R}$. Hence, if G is affine, $DG|_{E_r}$ is a constant. Moreover, we see that if G is a similarity mapping, so is $G|_{E_r}$, and it follows that $|DG|_{E_r}|$ is equal for all edges E_r of K_r. The same applies for a face F_r of K_r: $G|_{F_r}$ is understood to be the mapping from $\mathbb{R}^2$ to $\mathbb{R}^2$, mapping F_r to its corresponding face F of K. For a similarity mapping G we will now have that $|\det(DG|_{F_r})|$ is equal for all faces of K_r. In order to have a consistent notation, we will write $DG|_{E_r} = \det(DG|_{E_r})$.

We now have the following result:

Proposition 3.6. *Let K_r and K be two similar polygons in $\mathbb{R}^2$, or two similar polyhedra in $\mathbb{R}^3$, and let G be the similarity mapping from K_r to K. For a function $v \in H^1(K_r)$, we then have that*

$$\left(\Pi_k^{\nabla,K_r} v\right) \circ G^{-1} = \Pi_k^{\nabla,K}(v \circ G^{-1}).$$

In other words, we the following diagram commutes:

$$\begin{array}{ccc} H^1(K_r) & \xrightarrow{\circ\, G^{-1}} & H^1(K) \\ \Big\downarrow{\scriptstyle \Pi_k^{\nabla,K_r}} & & \Big\downarrow{\scriptstyle \Pi_k^{\nabla,K}} \\ \mathbb{P}_k(K_r) & \xrightarrow[\circ\, G^{-1}]{} & \mathbb{P}_k(K) \end{array}$$

Proof. Let $p \in \mathbb{P}_k(K)$. We have that

$$\begin{aligned} &a^K\left(v \circ G^{-1} - \Pi_k^{\nabla,K}(v \circ G^{-1}), p\right) \\ &= \int_K \nabla\left(v \circ G^{-1} - \Pi_k^{\nabla,K}(v \circ G^{-1})\right) \cdot \nabla p \, \mathrm{d}\boldsymbol{x} \\ &= \int_{K_r} |\det(DG)| \nabla_r \left(v - \Pi_k^{\nabla,K}(v \circ G^{-1}) \circ G\right) (DG)^{-1} \cdot \nabla_r (p \circ G)(DG)^{-1} \, \mathrm{d}\boldsymbol{x}_r, \end{aligned}$$

where $\boldsymbol{x}_r = (x_1^r, \ldots, x_d^r)$ are the coordinates on K_r, and ∇_r is the gradient in row vector form with respect to these coordinates. Moreover, DG is the Jacobian

matrix. Since G is affine, DG must be constant. The affineness also gives that $p \circ G \in \mathbb{P}_k(K_r)$, and we write this compactly as $p_r \coloneqq p \circ G$. We now have that

$$a^K\left(v \circ G^{-1} - \Pi_k^{\nabla,K}(v \circ G^{-1}), p\right) = |\det(DG)| \int_{K_r} \nabla_r \left(v - \Pi_k^{\nabla,K}(v \circ G^{-1}) \circ G\right) (DG)^{-1}(DG)^{-T}(\nabla_r p_r)^T \,\mathrm{d}\boldsymbol{x}_r.$$

We know from Remark 3.5 that $(DG)^{-1}(DG)^{-T} = \gamma^{-2}\boldsymbol{I}$, where γ is a constant. This gives

$$\begin{aligned} &a^K\left(v \circ G^{-1} - \Pi_k^{\nabla,K}(v \circ G^{-1}), p\right) \\ &\quad = \frac{1}{\gamma^2}|\det(DG)| \int_{K_r} \nabla_r \left(v - \Pi_k^{\nabla,K}(v \circ G^{-1}) \circ G\right) (\nabla_r p_r)^T \,\mathrm{d}\boldsymbol{x}_r \\ &\quad = \frac{1}{\gamma^2}|\det(DG)| a^{K_r}\left(v - \Pi_k^{\nabla,K}(v \circ G^{-1}) \circ G, p_r\right). \end{aligned}$$

Next, for $k = 1$ and $K \subset \mathbb{R}^2$, we have

$$\begin{aligned} &\int_{\partial K} \left(v \circ G^{-1} - \Pi_1^{\nabla,K}(v \circ G^{-1})\right) \mathrm{d}s \\ &\qquad = \sum_{E_r \subset \partial K_r} \int_{E_r} |\det(DG|_{E_r})| \left(v - \Pi_1^{\nabla,K}(v \circ G^{-1}) \circ G\right) \mathrm{d}s_r, \end{aligned}$$

where E_r is an edge of K_r, and $G|_{E_r}$ is the restriction of the mapping G to E_r. Naturally, $DG|_{E_r}$ is constant. Moreover, since K_r and K are similar, we know that $|\det(DG|_{E_r})|$ is equal for all edges E_r of K_r. Thus,

$$\begin{aligned} &\int_{\partial K} \left(v \circ G^{-1} - \Pi_1^{\nabla,K}(v \circ G^{-1})\right) \mathrm{d}s \\ &\qquad = |\det(DG|_{E_r})| \int_{\partial K_r} \left(v - \Pi_1^{\nabla,K}(v \circ G^{-1}) \circ G\right) \mathrm{d}s_r. \end{aligned}$$

Recalling Definition 2.10 and Definition 2.27 of Π^∇ in two and three dimensions, we see that the definitions of $\Pi_1^{\nabla,K}(v \circ G^{-1}) \circ G$ and $\Pi_1^{\nabla,K_r} v$ are equivalent, and conclude that

$$\Pi_1^{\nabla,K}(v \circ G^{-1}) = \left(\Pi_1^{\nabla,K_r} v\right) \circ G^{-1}. \tag{3.5}$$

In $\mathbb{R}^3$, we have that

$$\begin{aligned} &\int_{\partial K} \left(\Pi_1^{\nabla,\partial K}(v \circ G^{-1}) - \Pi_1^{\nabla,K}(v \circ G^{-1})\right) \mathrm{d}s \\ &\quad = \sum_{F_r \subset \partial K_r} \int_{F_r} |\det(DG|_{F_r})| \left(\Pi_1^{\nabla,\partial K}(v \circ G^{-1}) \circ G - \Pi_1^{\nabla,K}(v \circ G^{-1}) \circ G\right) \mathrm{d}s_r, \end{aligned}$$

where F_r is a face of K_r. Using what we just found for $k = 1$ in $\mathbb{R}^2$, we know that $\Pi_1^{\nabla,\partial K}(v \circ G^{-1}) \circ G = \Pi_1^{\nabla,\partial K_r} v$. Moreover, from the similarity of K_r and K, we know that $|\det(DG|_{F_r})|$ is equal for all faces F_r. Thus, arguing as in $\mathbb{R}^2$, we know that (3.5) holds in $\mathbb{R}^3$ as well.

Finally, for $k \geq 2$, we have that

$$\int_K \left(v \circ G^{-1} - \Pi_k^{\nabla,K}(v \circ G^{-1}) \right) \mathrm{d}\boldsymbol{x} = |\det(DG)| \int_{K_r} \left(v - \Pi_k^{\nabla,K}(v \circ G^{-1}) \circ G \right) \mathrm{d}\boldsymbol{x}_r.$$

Arguing as for $k = 1$, it follows that

$$\Pi_k^{\nabla,K}(v \circ G^{-1}) = \left(\Pi_k^{\nabla,K_r} v \right) \circ G^{-1}, \quad k \geq 2.$$

This concludes the proof. □

Note that this result says nothing about the case when K_r and K are not similar. However, in some special cases, the result is valid for $k = 1$ even if this is not the case. We will come back to this point later.

Remark 3.7. Looking at the proof of Proposition 3.6, one might be tempted to go for an even more general result, valid for a broader range of mappings G: We only need the mapping G to be such that $\det(DG)$ and $(DG)^T DG$ are constant for all $\boldsymbol{x}$. This is indeed the case if DG is a unitary matrix, since then, we have that $(DG)^T DG = \boldsymbol{I}$ and $\det(DG) = 1$ for all $\boldsymbol{x}$. However, it can be shown that a mapping G with DG unitary is an isometry (see for example [10]). Moreover, an isometry preserves distances, so that it is simply a similarity mapping with $\gamma = 1$. In other words, these mappings are already covered. ∘

With this result, we can prove the following useful property of our VEM basis functions:

Proposition 3.8. *Let $K_r \subset \mathbb{R}^d$ be a polygon or polyhedron, and let and K be an element similar to K_r. If the mapping G from K_r to K can be written on the form*

$$G(\boldsymbol{x}) = \gamma \boldsymbol{x} + \boldsymbol{b}, \tag{3.6}$$

where γ is a constant, and $\boldsymbol{b}$ is a vector in $\mathbb{R}^d$, the basis $\{\phi_r^i \circ G^{-1}\}_i$ is the unique canonical basis for $V_k(K)$.

Proof. It is sufficient to show that all functions $\phi_r^i \circ G^{-1} \in V_k(K)$, and that the degrees of freedom coincides with the degrees of freedom of the canonical basis $\{\phi^i\}_i$ of $V_k(K)$. Then, we know from Proposition 2.16 or Proposition 2.32 that $\{\phi_r^i \circ G^{-1}\}_i$ is the unique basis $\{\phi^i\}_i$.

We start with the case of K being a polygon in $\mathbb{R}^2$. Since G is affine, we can conclude $\phi^i \circ G^{-1} \in H^1(K)$, and that $\Delta(\phi_r^i \circ G^{-1}) \in \mathbb{P}_k(K)$. Moreover, since all basis functions $\phi_r^i|_E \in \mathbb{P}_k(E_r)$ for each edge E_r of K_r, it follows that

$(\phi_r^i \circ G^{-1})|_{\partial K} \in B_k(\partial K)$. Next, we need to check the condition

$$\left(\phi_r^i \circ G^{-1} - \mathit{\Pi}^{\nabla,K}(\phi_r^i \circ G^{-1}), m\right)_{0,K} = 0 \quad \forall m \in \mathcal{M}_{k-1}^* \cup \mathcal{M}_k^*(K). \tag{3.7}$$

Proposition 3.6 tells us that

$$\begin{aligned}
&\left(\phi_r^i \circ G^{-1} - \mathit{\Pi}^{\nabla,K}(\phi_r^i \circ G^{-1}), m\right)_{0,K} \\
&\quad = |\det(DG)| \int_{K_r} \left(\phi_r^i - \mathit{\Pi}^{\nabla,K}(\phi_r^i \circ G^{-1}) \circ G\right) m \circ G \,\mathrm{d}\boldsymbol{x}_r \\
&\quad = |\det(DG)| \int_{K_r} \left(\phi_r^i - \mathit{\Pi}^{\nabla,K_r}\phi_r^i\right) m \circ G \,\mathrm{d}\boldsymbol{x}_r = 0,
\end{aligned}$$

where we have used the fact that $m \circ G$ is a linear combination of the monomials $\mathcal{M}_{k-1}^*(K_r) \cup \mathcal{M}_k^*(K_r)$. Hence, we conclude that $\phi_r^i \circ G^{-1} \in V_k(K)$.

In $\mathbb{R}^3$, we can use what we found in $\mathbb{R}^2$ to conclude that $(\phi_r^i \circ G^{-1})|_{\partial K} \in V_k(\partial K)$. Furthermore, since G is affine, we see that $\phi^i \circ G^{-1} \in H^1(K)$, $\Delta(\phi_r^i \circ G^{-1}) \in \mathbb{P}_k(K)$, and that (3.7) is satisfied, so that $\phi_r^i \circ G^{-1} \in \mathbb{V}_k(K)$.

Both in $\mathbb{R}^2$ and $\mathbb{R}^3$, it is clear that $\phi_r^i \circ G^{-1}$ takes on the value 1 at all the vertices of K, so its degrees of freedom $\mathcal{V}^K$ coincides with $\mathcal{V}^K$ of ϕ^i. Moreover, since G is on the form (3.6), we have that $\boldsymbol{x}_K = \gamma \boldsymbol{x}_{K_r} + \boldsymbol{b}$, $h_K = \gamma h_{K_r}$, and $|K| = \gamma^d |K_r|$. Thus, for the degrees of freedom $\mathcal{P}^K$, we have that

$$\begin{aligned}
|K|^{-1}\left(\phi_r^i \circ G^{-1}, m^{\boldsymbol{\alpha}}\right)_{0,K} &= \frac{1}{|K|}\int_K \phi_r^i\left(\frac{\boldsymbol{x}-\boldsymbol{b}}{\gamma}\right)\left(\frac{\boldsymbol{x}-\boldsymbol{x}_K}{h_K}\right)^{\boldsymbol{\alpha}} \mathrm{d}\boldsymbol{x} \\
&= \frac{1}{\gamma^d |K_r|}\int_{K_r} \phi_r^i(\boldsymbol{x})\left(\frac{\gamma\boldsymbol{x}+\boldsymbol{b}-(\gamma\boldsymbol{x}_{K_r}+\boldsymbol{b})}{\gamma h_{K_r}}\right)^{\boldsymbol{\alpha}} \mathrm{d}(\gamma\boldsymbol{x}) \\
&= \frac{1}{\gamma^d |K_r|}\gamma^d \int_{K_r} \phi_r^i(\boldsymbol{x})\left(\frac{\boldsymbol{x}-\boldsymbol{x}_{K_r}}{h_{K_r}}\right)^{\boldsymbol{\alpha}} \mathrm{d}\boldsymbol{x} = 1,
\end{aligned}$$

so that the degrees of freedom $\mathcal{P}^K$ coincide as well. In the same manner, we can check that this is also true for $\mathcal{E}^K$, and in $\mathbb{R}^3$, for $\mathcal{F}^K$. This gives that all degrees of freedom of $\phi_r^i \circ G^{-1}$ and ϕ^i coincide, and we conclude that $\{\phi_r^i \circ G^{-1}\}_i$ is the unique canonical basis $\{\phi^i\}_i$ for $V_k(K)$. □

As explained in [34], we can say that the degrees of freedom scales as one. We can now see how s^K must scale in order to satisfy (3.1) with $\tilde{c}_1$ and $\tilde{c}_2$ independent on K and h_K: Let K_r and K be two similar elements in $\mathbb{R}^d$, with diameters $h_{K_r} = 1$ and h_K, respectively, and let them be oriented such that the similarity mapping taking K_r to K is $G : \boldsymbol{x} \mapsto \gamma\boldsymbol{x}$. It then follows that $\gamma = h_K$. We know from Proposition 3.8 that the ith basis function of $V_k(K)$ is $\phi^i(\boldsymbol{x}) = \phi_r^i(\boldsymbol{x}/h_K)$. Moreover, denoting the coordinates on K_r and K by $\boldsymbol{x}_r = (x_1^r, \ldots, x_d^r)^T$ and $\boldsymbol{x} = (x_1, \ldots, x_d)^T$, respectively, we have that

$$\frac{\partial \phi^i}{\partial x_j} = \frac{\partial \phi_r^i}{\partial x_j^r}\frac{\partial x_j^r}{\partial x_j} = \frac{1}{h_K}\frac{\partial \phi_r^i}{\partial x_j^r},$$

so that $\nabla\phi^i = \frac{1}{h_K}\nabla_r\phi_r^i$. This yields

$$a^K\left(\phi^i,\phi^i\right) = \int_K \left|\nabla\phi^i\right|^2 \mathrm{d}\boldsymbol{x} = \int_{K_r} \frac{1}{h_K^2}|\nabla_r\phi_r^i|^2 h_K^d \,\mathrm{d}\boldsymbol{x}_r = h_K^{d-2}a^{K_r}\left(\phi_r^i,\phi_r^i\right).$$

In other words, we see that a^K scales as h_K^{d-2}. More generally, if Assumption 2.19 is satisfied in $\mathbb{R}^2$, or Assumption 2.35 is satisfied in $\mathbb{R}^3$, we must ensure that s^K scale as h_K^{d-2} in order to obtain the optimal convergence properties mentioned in Theorem 2.44. See for [33, 3] for details.

We now ask the following question: Is it possible to choose the parameter matrix $\boldsymbol{\Sigma}$ such that $a_h^K(u_h,v_h) = a^K(u_h,v_h)$ for all $u_h, v_h \in V_k(K)$? Before we proceed, we introduce the useful concept of *generalized eigenvalues*:

Definition 3.9. Let $\Omega \subset \mathbb{R}^d$, with $d = 2$ or 3. We say that λ is a generalized eigenvalue of a^K if there exists a function $u \in H^1(\Omega)$, with $u \neq 0$, such that

$$a^K(u,v) = \lambda(u,v)_{0,K} \quad \forall v \in H^1(\Omega). \tag{3.8}$$

The function u is then an eigenfunction of a^K. ∘

If we let Ω be a polygon or polyhedron K, and consider functions $u_h, v_h \in V_k(K)$, we can use the interpolation identity (2.27) to obtain the matrix form of (3.8):

$$\boldsymbol{A}^K\hat{\boldsymbol{u}}_h = \lambda\boldsymbol{L}^K\hat{\boldsymbol{u}}_h,$$

where $\boldsymbol{A}_{i,j}^K = a^K(\phi^i,\phi^j)$, and $\boldsymbol{L}_{i,j}^K = (\phi^i,\phi^j)_{0,K}$. It can be shown that since $\boldsymbol{A}^K$ and $\boldsymbol{L}^K$ are symmetric and positive definite, there exists a basis of generalized real eigenvalues $\{\lambda_i\}_i$, with generalized eigenfunctions $\{\psi^i\}_i$. Moreover, we can choose this basis to be L^2-orthogonal [27, Chapter 15.3].

We now have the following important result, telling us how choose s^K in order to obtain the exact bilinear form a^K:

Theorem 3.10. *Let $\{\psi^i\}_i$ be an L^2-orthonormal basis for $\ker\Pi^\nabla$, with generalized eigenvalues $\{\lambda_i\}_i$, so that*

$$a^K(\psi^i,\psi^i) = \lambda_i,$$

and let $\boldsymbol{Q} = [\hat{\boldsymbol{\psi}}^1,\ldots,\hat{\boldsymbol{\psi}}^{n_{\ker}}]$ be its matrix representation. Let $u, v \in H^1(K)$, and let the stability term s^K be given by

$$s^K(u,v) = \hat{\boldsymbol{u}}^T\boldsymbol{Q}\boldsymbol{P}^{-1}\boldsymbol{\Lambda}\boldsymbol{P}^{-1}\boldsymbol{Q}^T\hat{\boldsymbol{v}}, \tag{3.9}$$

where $\boldsymbol{P} = \boldsymbol{Q}^T\boldsymbol{Q}$, and $\boldsymbol{\Lambda}$ is the diagonal matrix with the generalized eigenvalues $\{\lambda_i\}_i$ on its diagonal. Then, we have that

$$a_h^K(u_h,v_h) = a^K(u_h,v_h) \quad \forall u_h, v_h \in V_k(K).$$

Proof. Let u_h and v_h be two functions in $\ker \Pi^\nabla$. We write these as $\sum_i \xi_i \psi^i$ and $\sum_i \eta_i \psi^i$, respectively, where $\xi_i, \eta_i \in \mathbb{R}$ for all i. Then, we have

$$a^K(u_h, v_h) = a^K\left(\sum_{i=1}^{n_{\ker}} \xi_i \psi^i, \sum_{j=1}^{n_{\ker}} \eta_j \psi^j\right) = \sum_{i,j=1}^{n_{\ker}} \xi_i a^K(\psi^i, \psi^j)\eta_j.$$

Due to the orthonormality of the basis $\{\psi^i\}_i$, this simplifies to

$$a^K(u_h, v_h) = \sum_{i=1}^{n_{\ker}} \xi_i \lambda_i \eta_i = \boldsymbol{\xi}^T \boldsymbol{\Lambda} \boldsymbol{\eta},$$

where

$$\boldsymbol{\xi} = (\xi_1, \ldots, \xi_{n_{\ker}})^T, \quad \boldsymbol{\eta} = (\eta_1, \ldots, \eta_{n_{\ker}})^T.$$

Recalling that we can write $\hat{\boldsymbol{u}}_h = \boldsymbol{Q}\boldsymbol{\xi}$ and $\hat{\boldsymbol{v}}_h = \boldsymbol{Q}\boldsymbol{\eta}$, we also have that

$$s^K(u_h, v_h) = \boldsymbol{\xi}^T \boldsymbol{Q}^T \boldsymbol{Q} \boldsymbol{P}^{-1} \boldsymbol{\Lambda} \boldsymbol{P}^{-1} \boldsymbol{Q}^T \boldsymbol{Q} \boldsymbol{\eta} = \boldsymbol{\xi}^T \boldsymbol{\Lambda} \boldsymbol{\eta},$$

so that $s^K(u_h, v_h) = a^K(u_h, v_h)$ for all functions $u_h, v_h \in \ker \Pi^\nabla$.

Now, let u_h and v_h denote two functions in $V_k(K)$. Since $v - \Pi^\nabla v \in \ker \Pi^\nabla$ for any $v \in H^1(K)$, we have that

$$\begin{aligned} a_h^K(u_h, v_h) &= a^K\left(\Pi^\nabla u_h, \Pi^\nabla v_h\right) + s^K\left(u_h - \Pi^\nabla u_h, v_h - \Pi^\nabla v_h\right) \\ &= a^K\left(\Pi^\nabla u_h, \Pi^\nabla v_h\right) + a^K\left(u_h - \Pi^\nabla u_h, v_h - \Pi^\nabla v_h\right). \end{aligned}$$

Recalling the identity (2.24):

$$a^K(u, v) = a^K\left(\Pi^\nabla u, \Pi^\nabla v\right) + a^K\left(u - \Pi^\nabla u, v - \Pi^\nabla v\right) \quad \forall u, v \in H^1(K),$$

it is clear that

$$a^K(u_h, v_h) = a_h^K(u_h, v_h) \quad \forall u_h, v_h \in V_k(K),$$

which concludes the proof. □

We see that if we choose s^K to be on the form (3.9), the approximated bilinear form a_h^K is equal to the exact bilinear form a^K. Hence, we will refer to this choice of stability term as the *exact* stability term. We now take a moment to investigate two examples carefully.

Example 3.11. We consider a first order VEM for the rectangle

$$K = [-h_x, h_x] \times [-h_y, h_y] \subset \mathbb{R}^2,$$

shown in Figure 3.1.

We know that the restriction of Π_1^∇ to $V_1(K)$ is a projection from $V_1(K)$ of dimension $N^K = 4$, to $\mathbb{P}_1(K)$ of dimension $n_1 = 3$, so that $\ker \Pi_1^\nabla$ has dimension 1. We can check that xy is in $V_k(K)$, and that $\Pi_1^\nabla xy = 0$. We follow the procedure described in Theorem 3.10, and normalize the basis with respect to

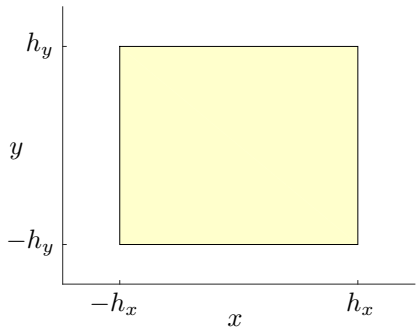


Figure 3.1: The rectangle $[-h_x, h_x] \times [-h_y, h_y]$.

$L^2(K)$ to obtain the L^2-orthonormal basis:

$$\{\psi^1\} = \left\{ \sqrt{\frac{9}{4h_x^3 h_y^3}} xy \right\}. \tag{3.10}$$

Its eigenvalue is then

$$\lambda_1 = a^K\left(\psi^1, \psi^1\right) = 3\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right).$$

The degrees of freedom are the values at the vertices, and we order them in counter-clockwise direction, staring at $(-h_x, -h_y)^T$. We obtain

$$\boldsymbol{Q} = \sqrt{\frac{9}{4h_x h_y}} \tilde{\boldsymbol{q}}, \tag{3.11}$$

where $\tilde{\boldsymbol{q}} = (1, -1, 1, -1)^T$. Note that in order to conform with the notation of Theorem 3.10, we have slightly abused our notation standard of denoting vectors in lower-case bold types. The exact stability term is now

$$\boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T = \frac{1}{12}\tilde{\boldsymbol{q}}\left(\epsilon + \epsilon^{-1}\right)\tilde{\boldsymbol{q}}^T, \tag{3.12}$$

where $\epsilon = h_y/h_x$ expresses the aspect ratio. Notice that all terms involving h_x and h_y are gathered in the term $(\epsilon + \epsilon^{-1})$. We clearly see that the exact stability term depends on the aspect ratio. ◦

Example 3.12. Next, we consider the cube

$$K = [-h_x, h_x] \times [-h_y, h_y] \times [-h_z, h_z] \subset \mathbb{R}^3,$$

shown in Figure 3.2.

In this case, Π_1^∇ is a projection from $V_1(K)$ of dimension 8 to $\mathbb{P}_1(K)$ of dimension 4, so that $\dim\ker \Pi_1^\nabla = 4$. A basis for $\ker \Pi^\nabla$ is given by $\{xy, xz, yz, xyz\}$.

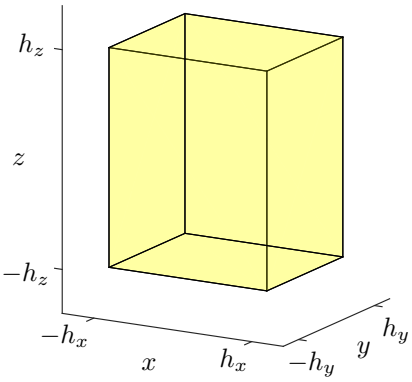


Figure 3.2: The cube $[-h_x, h_x] \times [-h_y, h_y] \times [-h_z, h_z]$.

We can check that these functions are also a^K-orthogonal. We follow the same procedure as in the previous example, and obtain the L^2-orthonormal basis

$$\{\psi^1, \psi^2, \psi^3, \psi^4\} = \left\{ \sqrt{\frac{9}{8h_x^3 h_y^3 h_z}} xy, \sqrt{\frac{9}{8h_x^3 h_y h_z^3}} xz, \sqrt{\frac{9}{8h_x h_y^3 h_z^3}} yz, \sqrt{\frac{27}{8h_x^3 h_y^3 h_z^3}} xyz \right\}.$$

The eigenvalues are

$$\lambda_1 = 3\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right), \quad \lambda_2 = 3\left(\frac{1}{h_x^2} + \frac{1}{h_z^2}\right), \quad \lambda_3 = 3\left(\frac{1}{h_y^2} + \frac{1}{h_z^2}\right),$$
$$\lambda_4 = 3\left(\frac{1}{h_x^2} + \frac{1}{h_y^2} + \frac{1}{h_z^2}\right).$$

We order the vertices as in Example 3.11 for $z = -h_z$ and $z = h_z$, and obtain

$$\boldsymbol{Q} = \sqrt{\frac{9}{8h_x h_y h_z}} \tilde{\boldsymbol{Q}}, \tag{3.13}$$

where

$$\tilde{\boldsymbol{Q}} = \begin{bmatrix} 1 & 1 & 1 & -\sqrt{3} \\ -1 & -1 & 1 & \sqrt{3} \\ 1 & -1 & -1 & -\sqrt{3} \\ -1 & 1 & -1 & \sqrt{3} \\ 1 & -1 & -1 & \sqrt{3} \\ -1 & 1 & -1 & -\sqrt{3} \\ 1 & 1 & 1 & \sqrt{3} \\ -1 & -1 & 1 & -\sqrt{3} \end{bmatrix}.$$

The exact stability term now reads

$$\boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T = \frac{1}{24}\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{\Sigma}}\tilde{\boldsymbol{Q}}^T,$$

where $\tilde{\boldsymbol{\Sigma}}$ is the matrix

$$\begin{bmatrix} h_z\left(\epsilon_{yx}+\epsilon_{yx}^{-1}\right) & 0 & 0 & 0 \\ 0 & h_y\left(\epsilon_{xz}+\epsilon_{xz}^{-1}\right) & 0 & 0 \\ 0 & 0 & h_x\left(\epsilon_{zy}+\epsilon_{zy}^{-1}\right) & 0 \\ 0 & 0 & 0 & \frac{1}{9}\left(h_z\epsilon_{yx}+h_y\epsilon_{xz}+h_x\epsilon_{zy}\right) \end{bmatrix},$$

and

$$\epsilon_{yx} = \frac{h_y}{h_x}, \quad \epsilon_{xz} = \frac{h_x}{h_z}, \quad \epsilon_{zy} = \frac{h_z}{h_y}.$$

Again, note that all terms involving h_x, h_y and h_z are gathered in the matrix $\tilde{\boldsymbol{\Sigma}}$. We clearly see how the aspect ratios are reflected in stability term. It should also be mentioned that in [33, 3, 34], the proposed stability term is on the form $\boldsymbol{S} = h_K\boldsymbol{I}$. But as we have shown, it is impossible to obtain the exact bilinear if we use a stability term on the form $\alpha h_K\boldsymbol{I}$, with $\alpha > 0$. ∘

3.2 Equivalence with the Finite Element Method

The finite element method (FEM) is in many aspects equal to the virtual element method. However, in a finite element method, we have explicit knowledge of our basis functions. This requires a heavy machinery of basis functions, numerical integration formulas and mappings from a reference element to the elements of the grid. Moreover, the classical finite element method can only be defined for certain cell geometries. We will not go into detail here, and the interested reader is referred to Appendix A and [5, 29].

We still consider the Poisson equation

$$-\Delta u = f, \quad \boldsymbol{x} \in \Omega, \tag{3.14}$$

where Ω is a subset of $\mathbb{R}^2$ or $\mathbb{R}^3$. We will also assume that suitable boundary conditions are given. As in VEM, for a given order $k \geq 1$, we seek to solve a

discrete version of the weak formulation of (3.14):

$$\text{Find} \quad u_h \in Q_h \quad \text{such that} \quad a(u_h, v_h) = (f, v_h)_{0,\Omega} \quad \forall v_h \in Q_h,$$

where Q_h is the global finite element function space. Note that since we have explicit knowledge of the FEM basis functions, we do not approximate the stiffness bilinear form a, but calculate it exactly. The load term $(f, \cdot)_{0,\Omega}$ must be approximated to the right order.

As in VEM, we consider a decomposition $\mathcal{T}_h$ of Ω into non-overlapping cells, and split the bilinear form as

$$a(u, v) = \sum_{K \in \mathcal{T}_h} a^K(u, v).$$

On each element K, we define a set of local basis functions $\{\varphi^i\}_i$, and obtain the local stiffness matrix $\boldsymbol{A}^{K,\text{FEM}}$, where $\boldsymbol{A}^{K,\text{FEM}}_{i,j} = a^K(\varphi^i, \varphi^j)$. Theorem 3.10 tells us how to construct the exact stability term s^K, which yields $a_h^K(u_h, v_h) = a^K(u_h, v_h)$ for all functions u_h, v_h in the virtual element function space $V_k(K)$. Hence, with this choice of s^K, if the local set of basis functions for FEM are the same as for VEM, the two methods will have the same local stiffness matrix, and we can consider them to be equivalent.

Due to the need for explicit definitions of the FEM basis functions, we will have to define the basis functions differently for each kind of geometry. We consider five different cell geometries: The triangle, quadrilateral, tetrahedron, hexahedron and triangular prism. With each of these cell geometries, we associate a reference element: The reference triangle K_t, square K_s, tetrahedron K_T, cube K_C, and prism K_P

3.2.1 FEM for triangles and tetrahedra

The reference triangle

We denote by K_t the reference triangle in $\mathbb{R}^2$, shown in Figure 3.3. The local function space for a k-th order FEM for the reference triangle is denoted $Q_k(K_t)$, and is defined in Definition A.2. We observe that we have 3 vertex functions, $k-1$ edge functions for each of the 3 edges, and $(k-1)(k-2)/2$ bubble functions. Hence, the local FEM space $Q_k(K_t)$ has dimension

$$N^{K_t,\text{FEM}} = 3k + (k-1)(k-2)/2.$$

We also recall the dimension of the local VEM space $V_k(K_t)$:

$$N^{K_t} = 3k + k(k-1)/2.$$

Thus, we have that for $k = 1$, $N^{K_t} = N^{K_t,\text{FEM}}$, while for $k \geq 2$, we have that $N^{K_t} > N^{K_t,\text{FEM}}$. Hence, for triangles, we can only analyse the equivalence between first order FEM and VEM. In a sense, for $k \geq 2$, FEM is less expensive than VEM for this cell geometry, since it uses fewer degrees of freedom.

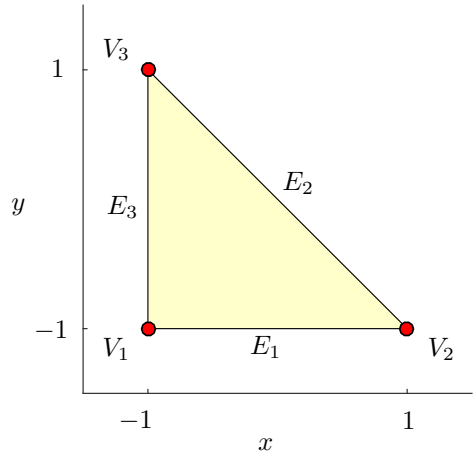


Figure 3.3: The reference triangle K_t.

The reference tetrahedron

Next, consider the reference tetrahedron K_T in $\mathbb{R}^3$, shown in Figure 3.4. We identify each face of K_T by its vertices, and choose the following numbering:

$$F_1 = V_1V_2V_4, \quad F_2 = V_2V_3V_4, \quad F_3 = V_1V_3V_4, \quad F_4 = V_1V_2V_3.$$

For a given order k, the local FEM space for K_T is denoted $\mathbb{Q}_k(K_T)$, and is defined in Definition A.3. In this case, we have 4 vertex functions, $k-1$ edge functions for each of the 6 edges, $(k-1)(k-2)/2$ face functions for each of the 4 faces, and $(k-3)(k-2)(k-1)/6$ bubble functions, so that the local FEM space $\mathbb{Q}_k(K_T)$ has dimension

$$N^{K_T,\text{FEM}} = 4 + 6(k-1) + 4(k-1)(k-2)/2 + (k-3)(k-2)(k-1)/6.$$

The local VEM space $\mathbb{V}_k(K_T)$ has dimension

$$N^{K_T} = 4 + 6(k-1) + 4k(k-1)/2 + k(k^2-1)/6.$$

Again, we have that for $k = 1$, $N^{K_T} = N^{K_T,\text{FEM}}$, while for $k \geq 2$, we have that $N^{K_T} > N^{K_T,\text{FEM}}$. Hence, we can only analyse equivalence between FEM and VEM of first order for the tetrahedron as well. We note that for $k \geq 2$, FEM is cheaper in terms of degrees of freedom than VEM for K_T.

The general case

Using the function spaces $Q_k(K_t)$ and $\mathbb{Q}_k(K_T)$, we can construct the local FEM spaces for any triangle in $\mathbb{R}^2$, or any hexahedron in $\mathbb{R}^3$. To do so, we introduce the following mapping:

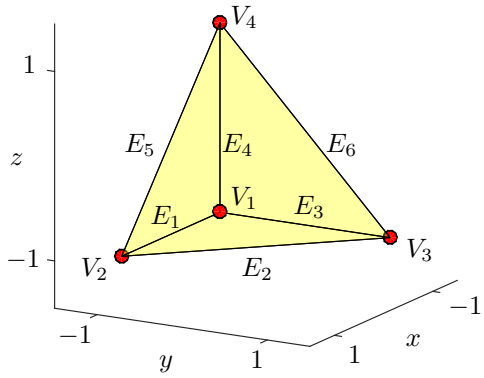


Figure 3.4: The reference tetrahedron K_T.

Definition 3.13. Let K_r be the reference triangle K_t in $\mathbb{R}^2$, or the reference tetrahedron K_T in $\mathbb{R}^3$, with coordinates $\boldsymbol{x}_r$, and denote the FEM vertex basis functions on K_r by $\{\varphi_r^{V_i}\}_i$. Let K be any triangle in $\mathbb{R}^2$ or any tetrahedron in $\mathbb{R}^3$, with vertex coordinates $\{\boldsymbol{x}^i\}_i$. We define the mapping G from K_r to K by

$$G(\boldsymbol{x}_r) = \sum\nolimits_i \boldsymbol{x}^i \varphi_r^{V_i}(\boldsymbol{x}_r).$$

∘

Remark 3.14. Note that since the basis functions of $Q_k(K_t)$ and $\mathbb{Q}_k(K_T)$ are linear, the mapping G is affine. ∘

Using this mapping, we construct our basis functions for triangles and tetrahedra:

Definition 3.15. Let K be a triangle $\mathbb{R}^2$ or a tetrahedron $\mathbb{R}^3$. Let G be the mapping from the reference element K_r to K, defined in Definition 3.13, and denote the ith basisfunction of $Q_k(K_r)$ by $\varphi_r^{V_i}$. The finite element basis functions for $Q_k(K)$ are

$$\varphi^{V_i} = \varphi_r^{V_i} \circ G^{-1}.$$

∘

Equivalence between FEM and VEM for triangles and tetrahedra

We are now ready to analyse equivalence between FEM and VEM for triangles and tetrahedra. Whenever no confusion can arise, we denote the local VEM and FEM spaces for an element K by $V_k(K)$ and $Q_k(K)$, respectively, both in two and three dimensions. The key is to notice the equivalence between the two function spaces:

Proposition 3.16. *Let K be a triangle in $\mathbb{R}^2$, or a tetrahedron in $\mathbb{R}^3$. Then, $V_1(K) = Q_1(K)$, and their bases $\{\phi^i\}_i$ and $\{\varphi^{V_i}\}_i$ are equal.*

Proof. By the definition of the local FEM spaces, we have that $\chi^j(\varphi^{V_i}) = \delta_{ij}$. Hence, from Proposition 2.16 and Proposition 2.32, we know that if all FEM basis functions φ^{V_i} is in $V_1(K)$ then the FEM basis is the unique VEM basis.

We start by recalling Definition 2.13 and Definition 2.29 of the local VEM spaces, and state the properties of a function v_h $V_1(K)$ or $\mathbb{V}_1(K)$:

$$v_h \in H^1(K), \tag{3.15a}$$

$$\Delta v_h|_K \in \mathbb{P}_1(K), \tag{3.15b}$$

$$v_h|_{\partial K} \in \begin{cases} B_1(K) & \text{for } v_h \in V_1(K), \\ V_1(\partial K) & \text{for } v_h \in \mathbb{V}_1(K), \end{cases} \tag{3.15c}$$

$$\left(v_h - \Pi_1^\nabla v_h, m\right)_{0,K} = 0 \quad \forall m \in \mathcal{M}_1(K). \tag{3.15d}$$

Let K be an arbitrary triangle in $\mathbb{R}^2$, and denote the basis of $Q_1(K)$ by $\{\varphi^{V_i}\}_i$. Let K_t be the reference triangle in Figure 3.3. The basis for $Q_1(K_t)$ is the the set of linear vertex interpolants $\{\varphi_t^{V_i}\}_i$. Let G be the mapping from Definition 3.13, taking K_t to K. We have from Remark 3.14 that G is affine. Since $\varphi^{V_i} = \varphi_t^{V_i} \circ G^{-1}$, we then know that all basis functions φ^{V_i} are linear, and conclude that they satisfy (3.15a)-(3.15c). Finally, we need to check that all basis functions φ^{V_i} satisfies (3.15d). But this is trivially true, since $\Pi_1^\nabla p = p$ for all polynomials $p \in \mathbb{P}_1(K)$, and all basis functions $\varphi^{V_i} \in \mathbb{P}_1(K)$. We can conclude that $Q_1(K) \subset V_1(K)$, and since they have equal dimension, $V_1(K) = Q_1(K)$. Moreover, since the degrees of freedom of their basis functions coincide, it is clear that $\{\phi^i\}_i = \{\varphi^{V_i}\}_i$.

Next, we consider an arbitrary tetrahedron K. We denote the basis $\mathbb{Q}_1(K)$ by $\{\varphi^{V_i}\}_i$. Let K_T be the reference tetrahedron in Figure 3.4. The basis for $\mathbb{Q}_1(K_T)$ is now the set $\{\varphi_T^{V_i}\}_i$ of linear vertex interpolants. We follow the 2D procedure, and conclude that (3.15a), (3.15b) and (3.15d) are satisfied. Moreover, the restriction of the basis functions to a face F of K is the FEM space $Q_1(F)$. Since all faces F are triangular, we can use what we just found in 2D to conclude that $\varphi^{V_i}|_{\partial K} \in V_1(\partial K)$, so that (3.15c) is also satisfied. Again, since the two function spaces have equal dimensions, and the degrees of freedom of their basis functions coincide, we conclude that $\mathbb{V}_1(K) = \mathbb{Q}_1(K)$, and $\{\phi^i\}_i = \{\varphi^{V_i}\}_i$. □

The main lesson to take away from this result is the following: The local stiffness matrix we are approximating in a first order virtual element method for triangles or tetrahedra is, in fact, the local stiffness matrix of the finite element method. We immediately have the following result:

Theorem 3.17. *Let K be a triangle in $\mathbb{R}^2$, or a tetrahedron in $\mathbb{R}^3$. Then, for K, the local stiffness matrix of the virtual element method and the finite element method are equal.*

Proof. From Proposition 3.16, we know that $V_1(K)$ and $Q_1(K)$ share the same basis. Moreover, since $Q_1(K)$ consists entirely of linear functions, we know that $\Pi_1^{\nabla,K} v_h = v_h$ for all $v_h \in V_1(K)$. It then follows that

$$\begin{aligned} a_h^K(u_h, v_h) &= a^K\left(\Pi^\nabla u_h, \Pi^\nabla v_h\right) + s^K\left(u_h - \Pi^\nabla u_h, v_h - \Pi^\nabla v_h\right) \\ &= a^K(u_h, v_h) \quad \forall u_h, v_h \in V_1(K), \end{aligned}$$

which concludes the proof. □

3.2.2 FEM for quadrilaterals and hexahedra

Next, we will analyse the equivalence between FEM and VEM for quadrilaterals and hexahedra.

The reference square

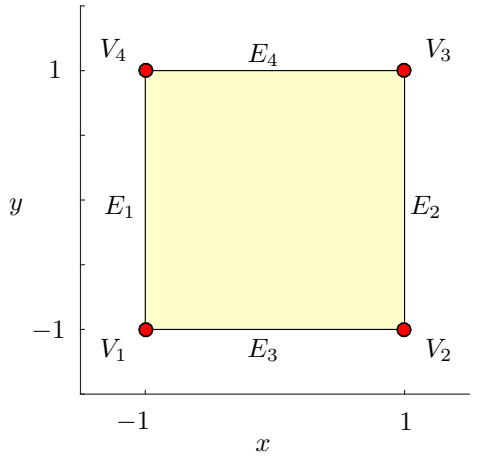


Figure 3.5: The reference square K_s.

We consider the reference square K_s, shown in Figure 3.5. The local finite element function space for a kth order FEM for K_s is denoted $Q_k(K_s)$, and is defined in Definition A.4. We observe that we have 4 vertex functions, $k-1$ edge functions for each of the 4 edges, and $(k-1)^2$ bubble functions. Hence, $Q_k(K_s)$ has dimension

$$N^{K_s,\text{FEM}} = 4k + (k-1)^2.$$

We also recall the dimension of the local VEM space:

$$N^{K_s} = 4k + k(k-1)/2.$$

In other words, for $k = 1$ and $k = 2$, $N^{K_s} = N^{K_s,\text{FEM}}$, while for $k \geq 3$, we have that $N^{K_s} < N^{K_s,\text{FEM}}$. Because of this, it only makes sense to analyse equivalence between VEM and FEM of first and second order. In this case, we see that for $k \geq 3$, VEM is cheaper than FEM in terms of degrees of freedom.

The reference cube

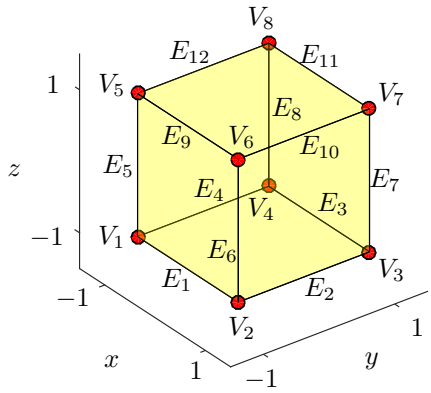


Figure 3.6: The reference cube K_C.

Next, we consider the reference cube K_C, shown in Figure 3.6. As for the reference tetrahedron, we identify the faces of K_C by their vertices, and choose the following numbering:

$$F_1 = V_1V_4V_5V_8, \quad F_2 = V_2V_3V_6V_7, \quad F_3 = V_1V_2V_5V_6,$$
$$F_4 = V_3V_4V_7V_8, \quad F_5 = V_1V_2V_3V_4, \quad F_6 = V_5V_6V_7V_8.$$

The local finite element space for a kth order FEM for K_C is denoted $\mathbb{Q}_k(K_C)$, and is defined in Definition A.5. We observe that we have 8 vertex functions, $k-1$ edge functions for each of the 12 edges, $(k-1)^2$ face functions for each of the 6 faces, and $(k-1)^3$ bubble functions. Hence, the local FEM space $\mathbb{Q}_k(K_C)$ has dimension

$$N^{K_C,\text{FEM}} = 8 + 12(k-1) + 6(k-1)^2 + (k-1)^3.$$

We also recall the dimension of the local VEM space $\mathbb{V}_k(K_C)$:

$$N^{K_C} = 8 + 12(k-1) + 6k(k-1)/2 + k(k^2-1)/6.$$

Again, we have that for $k = 1$ and $k = 2$, $N^{K_C} = N^{K_C,\text{FEM}}$, while for $k \geq 3$, we have that $N^{K_C} < N^{K_C,\text{FEM}}$. Hence, we can only analyse the equivalence

between FEM and VEM for first and second order for the cube as well. As for K_s, we observe that for $k \geq 3$, VEM uses fewer degrees of freedom than FEM.

The general case

Using the function spaces $Q_k(K_s)$ and $\mathbb{Q}_k(K_C)$, we can now build the local FEM spaces for any convex quadrilateral or hexahedron. To do so, we define the mapping G just as in Definition 3.13, but with K_t and K_T replaced by K_s and K_C. Furthermore, the construction of basis functions for convex quadrilaterals and hexahedra are now defined just as in Definition 3.15, but with *convex quadrilateral* and *convex hexahedron* in place of *triangle* and *tetrahedron*.

Equivalence between FEM and VEM for quadrilaterals and hexahedra

Again, when no confusion can arise, we will refer to the local VEM and FEM spaces for an element K by $V_k(K)$ and $Q_k(K)$, both in two and three dimensions.

When $k = 1$, we have from Definition A.4 that $Q_1(K_s)$ consists the polynomials $\{1, x, y, xy\}$, and from Definition A.5 that $\mathbb{Q}_1(K_C)$ consists of the polynomials $\{1, x, y, z, xy, xz, yz, xyz\}$. Using this, we can show the following result:

Lemma 3.18. *Let K_r be the reference square in Figure 3.5, or the reference cube in Figure 3.6. Then, $V_1(K_r) = Q_1(K_r)$.*

Proof. Since the two function spaces have the same dimension, it is sufficient to show that $Q_1(K_r) \subset V_1(K_r)$.

We follow the same procedure as in the proof of Proposition 3.16, and start with K_s. First of all, since $Q_1(K_s)$ is made up of the polynomials $\{1, x, y, xy\}$, it is clear that all basis functions $\varphi_s^{V_i}$ satisfies (3.15a)-(3.15c). We must also ensure that they satisfy (3.15d). That is, since $\Pi^\nabla p = p$ for all polynomials $p \in \mathbb{P}_1(K_s)$, it is sufficient to check that the relation is satisfied for the nonlinear terms for $Q_1(K_s)$. But the only nonlinear term in $Q_1(K_s)$ is xy, and we already know from Example 3.11 that $\Pi^\nabla xy = 0$. Moreover, due to the symmetry of K_s, the integral of xy multiplied by any monomial $m \in \mathcal{M}_1(K_s)$ over K_s is zero, so that (3.15d) is satisfied. Thus, it is clear that $Q_1(K_s) \subset V_1(K_s)$, and we conclude that $V_1(K_s) = Q_1(K_s)$. We also note that since the degrees of freedom of their basis functions coincide, we know from Proposition 2.16 that they share the same basis.

Next, we consider K_C. Since $\mathbb{Q}_1(K_C)$ consists of the polynomials

$$\{1, x, y, xy, xz, yz, xyz\},$$

it is clear that all basis function $\varphi_C^{V_i}$ satisfies (3.15a) and (3.15b). Moreover, the restriction of the basis functions to a face F_C of K_C is nothing but the basis of $Q_1(F_C) = Q_1(K_s)$. Hence, the above ensures that all basis functions satisfies (3.15c). Finally, we saw in Example 3.12 that the nonlinear terms xy, xz, yz and xyz are in $\ker \Pi^\nabla$. Again, the symmetry of K_C ensures that the integrals of these polynomials multiplied by any $m \in \mathcal{M}_1(K_C)$ over K_C are zero. Hence, the basis

functions also satisfies (3.15d). Again, we have that $\mathbb{Q}_1(K_C) \subset \mathbb{V}_1(K_C)$, and we conclude that $\mathbb{V}_1(K_C) = \mathbb{Q}_1(K_C)$. □

We now have that for $k = 1$, the FEM basis functions and VEM basis functions coincide for K_s and K_C. But what about other quadrilaterals and hexahedra? As we have seen, Definition 3.13 tells us how to define the FEM space for convex quadrilaterals and hexahedra through the mapping G. Before we proceed, we need the following special result:

Lemma 3.19. *Let K_r and K be the reference square K_s and a quadrilateral in $\mathbb{R}^2$, or the reference cube K_C and a hexahedron in $\mathbb{R}^3$, such that the mapping G from Definition 3.13 is affine, and let v_h be a function in $V_1(K_r)$. Then,*

$$\left(\Pi_1^{\nabla,K_r} v_h\right) \circ G^{-1} = \Pi_1^{\nabla,K}(v_h \circ G^{-1}). \tag{3.16}$$

Proof. We start by noting that for a cell K, any function $v \in H^1(K)$ can be split into one part in $\operatorname{Im} \Pi^\nabla$ and one part in $\ker \Pi^\nabla$ as

$$\Pi^\nabla v + \left(v - \Pi^\nabla v\right).$$

Since G is affine, it maps linear functions to linear functions. We know that $\operatorname{Im} \Pi_1^{\nabla,K_r} = \mathbb{P}_1(K_r)$, and it follows that

$$G^{-1}\left(\operatorname{Im} \Pi_1^{\nabla,K_r}\right) = \operatorname{Im} \Pi_1^{\nabla,K}.$$

Thus, in order to prove the lemma, it is sufficient to check that for a function $v_h \in \ker \Pi_1^{\nabla,K_r}$, we have that $v_h \circ G^{-1} \in \ker \Pi_1^{\nabla,K}$.

Let $v_h \in \ker \Pi_1^{\nabla,K_r}$, and $p \in \mathbb{P}_1(K)$. We recall the proof of Proposition 3.6, and calculate

$$a^K\left(v_h \circ G^{-1}, p\right) = \int_{K_r} |\det(DG)| \nabla_r v_h (DG)^{-1} (DG)^{-T} (\nabla_r p_r)^T \, \mathrm{d}\boldsymbol{x}_r,$$

where $p_r = p \circ G$. Since G is affine, we know that DG is constant. Moreover, we know that for a constant matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ and a differentiable, scalar function v, we have that $(\nabla v)\boldsymbol{A} = \nabla(\boldsymbol{A}v)$, so that we can write

$$(DG)^{-1}(DG)^{-T}(\nabla_r p_r)^T = \left(\nabla_r \left((DG)^{-1}(DG)^{-T} p_r\right)\right)^T.$$

Since p_r is linear, this is a constant vector, and we can write it as $(\nabla_r q)^T$ for a function $q \in \mathbb{P}_1(K_r)$. It then follows that

$$a^K\left(v_h \circ G^{-1}, p\right) = |\det(DG)| a^{K_r}(v_h, q) = 0,$$

where the last equality follows from the fact that $v_h \in \ker \Pi_1^{\nabla,K_r}$.

In $\mathbb{R}^2$, we have that

$$\begin{aligned}\int_{\partial K} v_h \circ G^{-1}\,\mathrm{d}s &= \sum_{E_s \subset \partial K_s} |\det(DG|_{E_s})| \int_{E_s} v_h \circ G^{-1} \circ G\,\mathrm{d}s_s \\ &= \sum_{E_s \subset \partial K_s} |\det(DG|_{E_s})| \int_{E_s} v_h\,\mathrm{d}s_s, \end{aligned} \tag{3.17}$$

where E_s is an edge of K_s, and $G|_{E_s}$ is the restriction of the mapping G to E_s. We know from Example 3.11 that xy is a basis for $\ker \Pi_1^{\nabla,K_s}$. Clearly, the integral of xy over each edge E_s of K_s is zero. Thus, we have that (3.17) is zero. It follows that $v_h \circ G^{-1}$ must be in $\ker \Pi_1^{\nabla,K}$, and we conclude that

$$\left(\Pi_1^{\nabla,K_s} v_h\right) \circ G^{-1} = \Pi_1^{\nabla,K}(v_h \circ G^{-1}) \quad \forall v_h \in V_1(K_s).$$

In $\mathbb{R}^3$, we have that

$$\begin{aligned}\int_{\partial K} \Pi_1^{\nabla,\partial K}\left(v_h \circ G^{-1}\right)\mathrm{d}s &= \sum_{F_C \subset \partial K_C} |\det(DG|_{F_C})| \int_{F_C} \Pi_1^{\nabla,F}\left(v_h \circ G^{-1}\right) \circ G\,\mathrm{d}s_C \\ &= \sum_{F_C \subset \partial K_C} |\det(DG|_{F_C})| \int_{F_C} \Pi_1^{\nabla,F_C} v_h\,\mathrm{d}s_C, \end{aligned} \tag{3.18}$$

where F is a face of K, F_C is a face of K_C, and $G|_{F_C}$ is the restriction of G to F_C. The last equality follows from what we found in $\mathbb{R}^2$. Now, we know from Example 3.12 that $\{xy, xz, yz, xyz\}$ constitutes a basis for $\ker \Pi_1^{\nabla,K_C}$. We look at the restriction of these polynomials to the first face F_1 of K_C (see Figure 3.6). These are

$$\{-y, -z, yz, -yz\}. \tag{3.19}$$

We know that $\Pi_1^{\nabla,F_1} p = p$ for all linear functions p, and the integral of the linear terms in (3.19) over F_1 are clearly zero. Moreover, we recognize the nonlinear terms $\pm yz$ as the basis of $\ker \Pi^{\nabla,F_1}$, so that these integrals are also zero. Arguing similarly for the remaining faces, we thus have that (3.18) is zero, and it follow that

$$\left(\Pi_1^{\nabla,K_C} v_h\right) \circ G^{-1} = \Pi_1^{\nabla,K}(v_h \circ G^{-1}) \quad \forall v_h \in \mathbb{V}_1(K_C).$$

This concludes the proof. □

Using this lemma, we can now prove the following equivalence result for quadrilaterals and hexahedra:

Proposition 3.20. *Let K be a quadrilateral in $\mathbb{R}^2$, or a hexahedron in $\mathbb{R}^3$, for which the mapping G in Definition 3.13 from their respective reference elements K_s or K_C to K is affine. Then, $V_1(K) = Q_1(K)$, and their bases $\{\phi^i\}_i$ and $\{\varphi^{V_i}\}_i$ are equal.*

Proof. We start in $\mathbb{R}^2$, and denote the basis functions $Q_1(K_s)$ and $Q_1(K)$ by $\{\varphi_s^{V_i}\}_i$ and $\{\varphi^{V_i}\}_i$, respectively. The function space $Q_1(K_s)$ consists of the polynomials $\{1, x, y, xy\}$. Since G is affine, and $p \circ G^{-1} \in \mathbb{P}_k(K)$ for all $p \in \mathbb{P}_k(K)$,

we can conclude that all basis functions $\varphi^{V_i} = \varphi_s^{V_i} \circ G^{-1}$ satisfies (3.15a) and (3.15b). Moreover, since the basis functions $\varphi_s^{V_i}$ are piecewise linear on ∂K_s, all basis functions φ^{V_i} are piecewise linear on ∂K, so that they satisfy (3.15c). Finally, we have that

$$\begin{aligned}\left(\varphi^{V_i} - \Pi_1^{\nabla,K}\varphi^{V_i}, m\right)_{0,K} &= \left(\varphi_s^{V_i} \circ G^{-1} - \Pi_1^{\nabla,K}(\varphi_s^{V_i} \circ G^{-1}), m\right)_{0,K} \\ &= |\det(DG)| \left(\varphi_s^{V_i} - \Pi_1^{\nabla,K}(\varphi_s^{V_i} \circ G^{-1}) \circ G, m \circ G\right)_{0,K_s},\end{aligned}$$

where $m \in \mathcal{M}_1(K)$, so that $m \circ G \in \mathbb{P}_1(K_s)$. Lemma 3.18 tells us that $\varphi_s^{V_i} \in V_1(K_s)$, and using this, Lemma 3.19 tells us that

$$\Pi_1^{\nabla,K}\left(\varphi_s^{V_i} \circ G^{-1}\right) \circ G = \Pi_1^{\nabla,K_s}\varphi_s^{V_i}.$$

It then follows that

$$\begin{aligned}&\left(\varphi^{V_i} - \Pi_1^{\nabla,K}\varphi^{V_i}, m\right)_{0,K} \\ &\quad = |\det(DG)| \left(\varphi_s^{V_i} - \Pi_1^{\nabla,K_s}\varphi_s^{V_i}, m \circ G\right)_{0,K_s} = 0 \quad \forall m \in \mathcal{M}_1(K),\end{aligned}$$

from which it is clear that all basis functions satisfies (3.15d). We conclude that $Q_1(K) \subset V_1(K)$, and since they have equal dimension, $V_1(K) = Q_1(K)$. Moreover, we know from Definition A.4 that $\chi^j(\varphi^{V_i}) = \chi^j(\phi^i) = \delta_{ij}$, and use Proposition 2.16 to conclude that $\{\phi^i\}_i = \{\varphi^{V_i}\}_i$.

In $\mathbb{R}^3$, since the restriction of $\mathbb{Q}_1(K)$ to one of the quadrilateral faces F of K is the function space $Q_1(F)$, we know from what we found in $\mathbb{R}^2$ that $\varphi^{V_i}|_{\partial K} \in V_1(\partial K)$. Moreover, we know that $\mathbb{Q}_1(K_C)$ consists of the polynomials $\{1, x, y, xy, xz, yz, xyz\}$. Thus, arguing as in $\mathbb{R}^2$, we have that all basis functions φ^{V_i} satisfies (3.15a)-(3.15c). Using Lemma 3.18, and Lemma 3.19, we know that

$$\left(\varphi^{V_i} - \Pi^{\nabla,K}\varphi^{V_i}, m\right)_{0,K} = 0 \quad \forall m \in \mathcal{M}_1(K),$$

so that all basis functions satisfy (3.15d) as well. Again, since the two function spaces have equal dimension, we conclude that $\mathbb{V}_1(K) = \mathbb{Q}_1(K)$. Moreover, Definition A.5 tells us that $\chi^j(\varphi^{V_i}) = \chi^j(\phi^i) = \delta_{ij}$, and we use Proposition 2.32 to conclude that $\{\phi^i\}_i = \{\varphi^{V_i}\}_i$. □

As for triangles and tetrahedra, we see that in a first order VEM for a quadrilateral or hexahedron for which the mapping G from their respective reference elements K_s or K_C is affine, the local stiffness matrix we are approximating is, in fact, the local stiffness matrix of the finite element method. The hard work now pays off, and we immediately have the following result:

Theorem 3.21. *Let K be a quadrilateral in $\mathbb{R}^2$, or a hexahedron in $\mathbb{R}^3$, for which the mapping G in Definition 3.13 from their respective reference element K_s or K_C to K is affine. Then, for K, the local stiffness matrices of the first order virtual element method, with the exact stability term (3.9), and the first order finite element method, are equal.*

Proof. This follows directly from Proposition 3.20 and Theorem 3.10, since the former tells us that $V_1(K)$ and $Q_1(K)$ are equal and share the same basis, and the latter tells us that with the given choice of s^K, $a_h^K(u_h, v_h) = a^K(u_h, v_h)$ for all functions $u_h, v_h \in V_1(K)$. □

Note that the result is, in general, not valid convex quadrilaterals and hexahedra when the mapping G is not affine. We consider a counter-example:

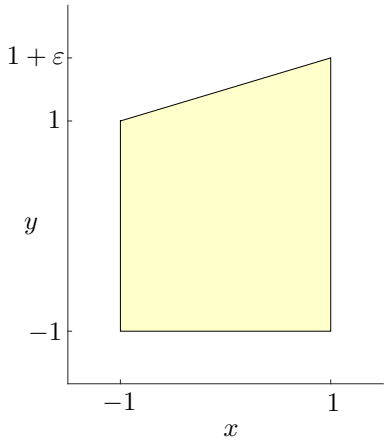


Figure 3.7: The distorted reference square K_ε.

Example 3.22. Let K_ε be the distorted reference square, with vertices

$$(-1,-1)^T, \quad (1,-1)^T, \quad (1,1+\varepsilon)^T, \quad (-1,1)^T,$$

shown in Figure 3.7. We find that

$$G(x,y) = \left(x, y + \varepsilon\varphi_s^{V_3}(x,y)\right)^T, \quad G^{-1}(x,y) = \left(x, \frac{4y - \varepsilon(1+x)}{4+\varepsilon(1+x)}\right)^T.$$

Definition A.4 and Definition 3.13 gives that the FEM basis functions are

$$\varphi^{V_i}(x,y) = l_{d_x}(x) l_{d_y}\left(\frac{4y-\varepsilon(1+x)}{4+\varepsilon(1+x)}\right),$$

where l_{d_x} and l_{d_y} are the Lobatto shape functions from Definition A.1, with (d_x, d_y) defined as in Definition A.4. We consider the first vertex function φ^{V_1}, and calculate its Laplacian:

$$\Delta\varphi^{V_1} = \frac{\varepsilon(\varepsilon-4y)(\varepsilon+2)}{(4+\varepsilon(1+x))^3},$$

which is not in $\mathbb{P}_1(K_\varepsilon)$. Hence, we conclude that $Q_1(K_\varepsilon) \not\subset V_1(K_\varepsilon)$.

In $\mathbb{R}^3$, consider a hexahedron K where, for example, two opposite faces are similar to K_ε. This would mean that $\mathbb{Q}_1(K)|_{K_\varepsilon} = Q_1(K_\varepsilon) \not\subset V_1(K_\varepsilon)$, so that $\mathbb{Q}_1(K) \not\subset \mathbb{V}_1(K)$. ◦

As mentioned, the dimensions of the local VEM space and the local FEM space are equal also for $k = 2$. In this case, the methods differs in the sense that they only share their first degrees of freedom, namely the values at the vertices. However, it might be possible to show that the two function spaces consists of the same set of polynomials, such that one can construct the local VEM space using the local FEM space, but we will not do this here.

3.2.3 FEM for triangular prisms

We now consider the final piece of the 3D puzzle; the triangular prisms. These are commonly used in 3D grids to connect hexahedra and tetrahedra.

The reference prism

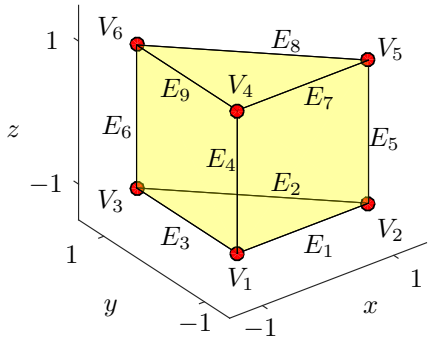


Figure 3.8: The reference prism K_P.

We consider the reference prism K_P in $\mathbb{R}^3$, shown in Figure 3.8. As usual, we identify a face by its vertices, and choose

$$F_1 = V_1V_2V_4V_5, \quad F_2 = V_2V_3V_5V_6, \quad F_3 = V_1V_3V_4V_6,$$
$$F_4 = V_1V_2V_3, \quad F_5 = V_4V_5V_6.$$

The local FEM space for a k-th order FEM is denoted $\mathbb{Q}_k(K_P)$, and is defined in Definition A.6. For the prism, we have 6 vertex functions, $k-1$ edge functions for each of the 9 edges, $(k-2)(k-1)/2$ face functions for each of the 2 triangular faces, $(k-1)^2$ face functions for each of the 3 rectangular faces, and $(k-2)(k-1)^2/2$ bubble functions. Hence, the local FEM space $\mathbb{Q}_k(K_P)$ has dimension

$$N^{K_P,\text{FEM}} = 6 + 9(k-1) + 2(k-2)(k-1)/2 + 3(k-1)^2 + (k-2)(k-1)^2/2.$$

The dimension of the local VEM space $\mathbb{V}_k(K_P)$ is

$$N^{K_P} = 6 + 9(k-1) + 5k(k-1)/2 + k(k^2-1)/6.$$

We observe that for $k = 1$, $N^{K_P} = N^{K_P,\text{FEM}}$, while for $k \geq 2$, we have that $N^{K_P} > N^{K_P,\text{FEM}}$. Hence, we can only analyse the equivalence between FEM and VEM for triangular prisms when $k = 1$. As for triangles and tetrahedra, we note that FEM uses fewer degrees of freedom than VEM for $k \geq 2$.

The general case

As in the previous cases, the mapping G from K_P to any triangular prism K is as in Definition 3.13, with K_P in place of K_t and K_T. Moreover, the basis functions for $\mathbb{Q}_k(K)$ are constructed as in Definition 3.15, with *triangular prism* in place of *triangle* and *tetrahedron*.

Equivalence between FEM and VEM for prisms

When $k = 1$, we have from Definition A.6 that $\mathbb{Q}_1(K_P)$ consists of the polynomials $\{1, x, y, z, xz, yz\}$. We now have the following result:

Lemma 3.23. *Let K_P be the reference prism. Then, $\mathbb{V}_1(K_P) = \mathbb{Q}_1(K_P)$.*

Proof. We follow the same procedure as in the previous proofs.

Since $\mathbb{Q}_1(K_P)$ consists of the polynomials $\{1, x, y, z, xz, yz\}$, it is clear that all basis functions $\varphi_P^{V_i}$ satisfies (3.15a) and (3.15b). Moreover, the restriction of $\mathbb{Q}_1(K_P)$ to a rectangular face F_r of K_P is $Q_1(F_r)$, which we know from Proposition 3.20 equals $V_1(F_r)$. Similarly, we know from Proposition 3.16 that the restriction to a triangular face F_t is $Q_1(F_t) = V_1(F_t)$. Thus, we have that $\varphi_P^{V_i}\big|_{\partial K_P} \in V_1(\partial K_P)$, so that all basis functions satisfy (3.15c) as well. Finally, we recall that the nonlinear terms of $\mathbb{Q}_1(K_P)$ must satisfy (3.15d). We calculate

$$\Pi_1^{\nabla,K_P} xz = \Pi_1^{\nabla,K_P} yz = -\frac{1}{3}z.$$

We can check that the integrals of $xz + z/3$ and $yz + z/3$ multiplied by any $m \in \mathcal{M}_1(K_P)$ over K_P are zero, and it follows that all basis functions satisfy (3.15d). It is thus clear that $\mathbb{Q}_1(K_P) \subset \mathbb{V}_1(K_P)$, and since they have equal dimension, we conclude that $\mathbb{V}_1(K_P) = \mathbb{Q}_1(K_P)$. □

As for quadrilaterals and hexahedra, we need the following special result in order to proceed:

Lemma 3.24. *Let K be a triangular prism such that the mapping G from K_P to K in Definition 3.13 is affine, and let v_h be a function in $\mathbb{V}_1(K_P)$. Then,*

$$\left(\Pi_1^{\nabla,K_P} v_h\right) \circ G^{-1} = \Pi_1^{\nabla,K}(v_h \circ G^{-1}).$$

Proof. We follow the same procedure as in the proof of Lemma 3.19, and recall that we only need to consider $\ker \Pi_1^{\nabla,K_P}$. First, we need a basis for $\ker \Pi_1^{\nabla,K_P}$, and note that the restriction of Π_1^{∇,K_P} to $\mathbb{V}_1(K_P)$ is a projection from a 6-dimensional function space to a 4-dimensional function space, so that $\dim \ker \Pi_1^{\nabla,K_P} = 2$. We use the functions $\varphi_P^{V_2}$ and $\varphi_P^{V_3}$ to obtain a basis for $\ker \Pi_1^{\nabla,K_P}$:

$$\begin{aligned}\{\psi^1,\psi^2\} &= \left\{-12\left(\varphi_P^{V_2} - \Pi_1^{\nabla,K_P}\varphi_P^{V_2}\right), -12\left(\varphi_P^{V_3} - \Pi_1^{\nabla,K_P}\varphi_P^{V_3}\right)\right\}\\ &= \{4z + xz, 4z + yz\}.\end{aligned}$$

For a function $v_h \in \ker \Pi_1^{\nabla,K_P}$, it follows in the same manner as in the proof of Lemma 3.19 that

$$a^K\left(v_h \circ G^{-1}, p\right) = 0 \quad \forall p \in \mathbb{P}_1(K).$$

Moreover we have that

$$\int_{\partial K} \Pi_1^{\nabla,\partial K}\left(v_h \circ G^{-1}\right) \mathrm{d}s = \sum_{F_P \subset \partial K_P} |\det(DG|_{F_P})| \int_{F_P} \Pi_1^{\nabla,F}\left(v_h \circ G^{-1}\right) \circ G \,\mathrm{d}s_P.$$

where F is a face of K, F_P is a face of K_P, and $G|_{F_P}$ is the restriction of G to F_P. We note that for a rectangular face F_r of K_P, since $\mathbb{V}_1(K_P) = \mathbb{Q}_1(K_P)$ consists of the polynomials $\{1, x, y, z, xz, yz\}$, we have that $v_h|_{F_r} \in V_1(F_r)$. Hence, denoting by F a quadrilateral face of K, we have from Lemma 3.19 that

$$\Pi_1^{\nabla,F}(v_h \circ G^{-1}) \circ G = \Pi_1^{\nabla,F_r} v_h.$$

Moreover, the restriction of v_h to a triangular face F_t of K_P is linear. Hence, we have that

$$\int_{\partial K} \Pi_1^{\nabla,\partial K}\left(v_h \circ G^{-1}\right) \mathrm{d}s = \sum_{F_P \subset \partial K_P} |\det(DG|_{F_P})| \int_{F_P} \Pi_1^{\nabla,F_P} v_h \,\mathrm{d}s_P. \tag{3.20}$$

Now, since it is sufficient to consider the basis of $\ker \Pi_1^{\nabla,K_P}$, we let v_h be the basis function ψ^1. The restriction of ψ^1 to each face of K_P is

$$\left\{\psi^1|_{F_i}\right\}_i = \{4z + xz, 4z + xz, 3z, -(x+4), x+4\}.$$

For the rectangular faces F_1, F_2 and F_3, we see that the integrals of the linear terms vanish, while we recall from Example 3.11 that $xz \in \ker \Pi_1^{\nabla,F_i}$ for $i = 1$ and 2. Moreover, for the triangular faces F_4 and F_5, we have have that the integrals

$$-\int_{F_4} (x+4) \,\mathrm{d}s_P \quad \text{and} \quad \int_{F_5} (x+4) \,\mathrm{d}s_P$$

cancels. Hence, it is clear that (3.20), with ψ^1 in place of v_h, is zero, and we have that $\psi^1 \circ G^{-1} \in \ker \Pi^{\nabla,K}$. Arguing similarly, for ψ^2, we get that if $v_h \in$

$\ker \Pi_1^{\nabla,K_P}$, then $v_h \circ G^{-1} \in \ker \Pi_1^{\nabla,K}$. As in Lemma 3.19, we conclude that

$$\left(\Pi_1^{\nabla,K_P} v_h\right) \circ G^{-1} = \Pi_1^{\nabla,K}(v_h \circ G^{-1}) \quad \forall v_h \in \mathbb{V}_1(K_P).$$

□

Combining this with the previous results of this chapter, we can prove the final equivalence:

Proposition 3.25. *Let K be a triangular prism such that the mapping G from K_P to K in Definition 3.13 is affine. Then, $\mathbb{V}_1(K) = \mathbb{Q}_1(K)$, and their bases $\{\phi^i\}_i$ and $\{\varphi^{V_i}\}_i$ are equal.*

Proof. We denote the basis of $\mathbb{Q}_1(K_P)$ and $\mathbb{Q}_1(K)$ by $\{\varphi_P^{V_i}\}_i$ and $\{\varphi^{V_i}\}_i$, respectively. We know from Definition A.6 that $\chi^j(\varphi^{V_i}) = \delta_{ij}$. Since $\mathbb{Q}_1(K_P)$ consists of the polynomials $\{1, x, y, z, xz, yz\}$, and since G is affine, we know that all basis functions φ^{V_i} satisfies (3.15a) and (3.15b). Moreover, the restriction of $\mathbb{Q}_1(K)$ to a rectangular face F_r is $Q_1(F_r)$, which we know from Proposition 3.20 equals $V_1(F_r)$. Similarly, we know from Proposition 3.16 that the restriction to a triangular face F_t is $Q_1(F_t) = V_1(F_t)$. Hence, we have that $\varphi^{V_i}\big|_{\partial K} \in V_1(\partial K)$ for all basis functions, so that (3.15c) is satisfied. Finally, Lemma 3.23 tells us that $\mathbb{V}_1(K_P) = \mathbb{Q}_1(K_P)$. Using this, we know from Lemma 3.24 that

$$\begin{aligned}&\left(\varphi^{V_i} - \Pi_1^{\nabla,K}\varphi^{V_i}, m\right)\\ &\quad = |\det(DG)|\left(\varphi_P^{V_i} - \Pi_1^{\nabla,K_P}\varphi_P^{V_i}, m \circ G\right) = 0 \quad \forall m \in \mathcal{M}_1(K),\end{aligned}$$

so that all basis functions satisfies (3.15d) as well. It is now clear that $\mathbb{Q}_1(K) \subset \mathbb{V}_1(K)$, and since they have equal dimension, we conclude that $\mathbb{V}_1(K) = \mathbb{Q}_1(K)$. Moreover, since the degrees of freedom of their basis functions coincide, we know from Proposition 2.32 that $\{\phi^i\}_i = \{\varphi^{V_i}\}_i$. □

As for the other cell geometries, we have the following result:

Theorem 3.26. *Let K be a triangular prism in $\mathbb{R}^3$ such that the mapping G in Definition 3.13 from K_P to K is affine. Then, for K, the local stiffness matrices of the first order virtual element method, with the exact stability term (3.9), and the first order finite element method, are equal.*

The proof is equivalent to that of Theorem 3.21, but with Proposition 3.25 in place of Proposition 3.20.

We note that there might exist triangular prisms K for which the mapping G is not affine, but such that FEM and VEM have equal local stiffness matrices. However, following Example 3.22, this is not true in general, since we could have a case where two faces are similar to K_ε, and, in turn, that $\mathbb{Q}_1(K)|_{K_\varepsilon} = Q_1(K_\varepsilon) \neq V_1(K_\varepsilon)$.

3.2.4 Some remarks

We take a moment to sum up what we have just shown. Given a first order FEM, and a first order VEM with the exact stability term (3.9), we know that their local stiffness matrices are

(i) equal for all triangles and tetrahedra,

(ii) equal for all quadrilaterals and hexahedra K for which the mapping G from the reference element K_r to K is affine,

(iii) equal for all triangular prisms K for which the mapping G from K_P to K is affine.

As usual, G is the mapping from Definition 3.13. Note that for triangles and tetrahedra, we know from Remark 3.14 that G is always affine.

One might also wonder exactly how large the group of quadrilaterals, hexahedra and triangular prisms with equal first order FEM and VEM stiffness matrices is. The answer lies in the property that parallel lines stay parallel after an affine mapping. Thus, the quadrilateral resulting from an affine mapping from K_s must necessarily have its opposite edges parallel. In other words, it is a *parallelogram*. Similarly, the hexahedron resulting from an affine mapping from K_C is a *parallelepiped*, and the triangular prism resulting from an affine mapping from K_P is a triangular prism with *parallelogram* sides. Examples of such cell geometries are shown in Figure 3.9.

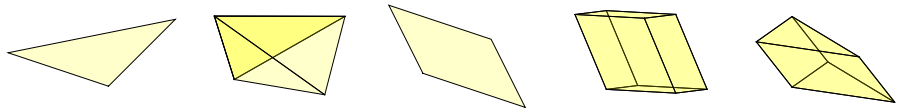


Figure 3.9: Examples of cell geometries for which FEM and VEM can have equal local stiffness matrices.

3.3 Equivalence with Finite Difference methods

Finite difference methods are based on Taylor series expansion, and due to its simplicity, it is used in numerous applications. An example within subsurface modelling is solving visco-acoustic wave propagation problems [26].

In the finite difference methods we analyse, we disregard the grid structure we use in VEM, and care only about the vertices of the grid. This typically results in a stencil, which is a geometric arrangement of the vertices in a certain part of the grid. We consider a rectangular or cubical domain, covered by a grid $\mathcal{T}_h$ of equally shaped rectangular or cubical elements. For a first order virtual element method, since we only use the vertex values as degrees of freedom, we might as well view the resulting system of linear equations as a result of a finite

difference stencil. With this mindset, we will now investigate what choices of the stability term s^K that makes VEM equivalent to various finite difference methods.

Before we proceed, we ease the notation with the following definition:

Definition 3.27. Let Ω denote the rectangular domain $[-H_x, H_x] \times [-H_y, H_y]$ in $\mathbb{R}^2$, or the cubical domain $[-H_x, H_x] \times [-H_y, H_y] \times [-H_z, H_z]$ in $\mathbb{R}^3$. We define the regular Cartesian grid over Ω to be the grid of adjacent, non-overlapping rectangles of equal dimensions $2h_x \times 2h_y$ in $\mathbb{R}^2$, or non-overlapping cubes of equal dimension $2h_x \times 2h_y \times 2h_z$ in $\mathbb{R}^3$. We number the vertices by the multiindex (i, j) in $\mathbb{R}^2$, and (i, j, k) in $\mathbb{R}^3$, which indicates their position along the coordinate axes. A regular Cartesian grid in $\mathbb{R}^2$ is denoted by $\mathcal{T}_{h_x,h_y}$, and in $\mathbb{R}^3$ by $\mathcal{T}_{h_x,h_y,h_z}$. ∘

Note that this gives a convenient mapping between the vertices and their coordinates: Defining $n_\xi \coloneqq H_\xi/h_\xi$, we have the following relation in $\mathbb{R}^2$:

$$(i, j) \leftrightarrow \boldsymbol{x}_{i,j} = (-H_x + 2ih_x, -H_y + 2jh_y)^T, \quad 0 \le i \le n_x, \quad 0 \le j \le n_y.$$

Further, we will denote the value of a function $u : \Omega \to \mathbb{R}$ at vertex (i, j) by $u_{i,j} \coloneqq u(\boldsymbol{x}_{i,j})$. The same applies in $\mathbb{R}^3$.

It is important to understand the difference between finite difference methods and VEM: The former is a direct discretization of the differential equation using Taylor series expansion, while the latter is a discretization of the weak formulation of the differential equation.

Again, we consider the Poisson equation

$$-\Delta u = f, \quad \boldsymbol{x} \in \Omega, \tag{3.21}$$

where Ω is a rectangular domain in $\mathbb{R}^2$, or a cubical domain in $\mathbb{R}^3$, and $f \in L^2(\Omega)$. We will also assume that suitable boundary conditions are given.

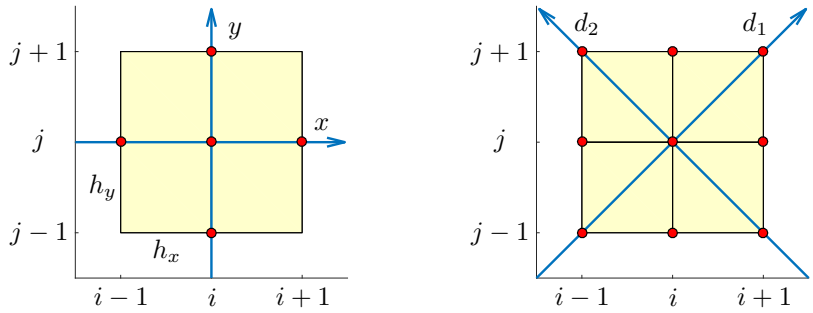
3.3.1 Two dimensions

We consider the two finite difference stencils in Definition B.1. The stencils are shown in Figure 3.10.

Obviously, any weighted average of these two stencils will also result in a discretization of the Laplace operator. Using these discretizations in equation (3.21), and multiplying both sides by $4h_xh_y$, the right-hand side reads $4h_xh_yf_{i,j}$. Further, we recall the VEM approximation of the right-hand side:

$$\langle f_h, \phi^i \rangle = (\Pi_0 f, \phi^i)_{0,\Omega}.$$

If f is constant, we have that $\Pi_0 f = f$. For a rectangular cell K, we know from Proposition 3.20 that $V_k(K) = Q_k(K)$. Using this, we find that the right-hand side of the VEM formulation of (3.21) equals $4h_xh_yf$ for all basis functions with compact support in Ω. That is, with f constant, the two methods yields the same right-hand side.



(a) Stencil 1. (b) Stencil 2.

Figure 3.10: The two 2D finite difference stencils, along with their coordinate axes.

Before we proceed, we recall that the global stiffness bilinear form can be split as

$$a_h(u_h, v_h) = \sum_{K \in \mathcal{T}_{h_x,h_y}} a_h^K(u_h, v_h) \quad \forall u_h, v_h \in V_h.$$

In the same manner, the global stiffness matrix $\boldsymbol{A}$, which is defined by $\boldsymbol{A}_{i,j} = a_h(\phi^i, \phi^j)$, is built by assembling the local stiffness matrices $\boldsymbol{A}^K$. We will come back to how this is done in Chapter 4. For now, it is sufficient to notice that for two basis functions ϕ^i, ϕ^j only the cells K on which both ϕ^i and ϕ^j are nonzero will have $a^K(\phi^i, \phi^j) \neq 0$, and contribute to $a_h(\phi^i, \phi^j)$.

Theorem 3.28. *Let $\Omega = [-H_x, H_x] \times [-H_y, H_y] \subset \mathbb{R}^2$, and $\mathcal{T}_{h_x,h_y}$ be a regular Cartesian grid over Ω. Let $\boldsymbol{F}_c$ and $\boldsymbol{F}_d$ be the finite difference matrices for the discrete Laplacian, as defined in Definition B.1. Further, let $\boldsymbol{A}$ be the global stiffness matrix obtained from the first order virtual element method on $\mathcal{T}_{h_x,h_y}$, with local stability terms s^K given by*

$$s^K(u, v) = \hat{\boldsymbol{u}}^T \boldsymbol{Q} \boldsymbol{P}^{-1} \tilde{\boldsymbol{\Lambda}} \boldsymbol{P}^{-1} \boldsymbol{Q}^T \hat{\boldsymbol{v}},$$

where the vector $\boldsymbol{Q}$ is given by (3.11), $\boldsymbol{P} = \boldsymbol{Q}^T \boldsymbol{Q}$, and $\tilde{\boldsymbol{\Lambda}}$ is the scaled eigenvalue matrix

$$\tilde{\boldsymbol{\Lambda}} = 3w\lambda_1 = 9w \left(\frac{1}{h_x^2} + \frac{1}{h_y^2} \right), \quad 0 < w \leq 1.$$

Then, we have that

$$w\boldsymbol{F}_c + (1 - w)\boldsymbol{F}_d = \boldsymbol{A}. \tag{3.22}$$

For the Poisson equation (3.21), if f is a constant function, the methods are equivalent.

Notice the similarity between this stability term and the exact stability term (3.9).

Proof. We number the vertices in the same way as in Figure 3.10, and start by writing the finite difference approximation of (3.21) using Stencil 1:

$$\epsilon(-u_{i-1,j} + 2u_{i,j} - u_{i+1,j}) + \epsilon^{-1}(-u_{i,j-1} + 2u_{i,j} - u_{i,j+1}) = 4h_x h_y f_{i,j}. \quad (3.23)$$

Using Stencil 2, we get the equation

$$\begin{aligned} -\frac{1}{4}(\epsilon + \epsilon^{-1})(u_{i-1,j-1} + u_{i-1,j+1} - 4u_{i,j} + u_{i+1,j-1} + u_{i+1,j+1}) \\ +\frac{1}{2}(\epsilon - \epsilon^{-1})(u_{i,j-1} - u_{i-1,j} - u_{i+1,j} + u_{i,j+1}) = 4h_x h_y f_{i,j}. \end{aligned} \quad (3.24)$$

We temporarily rename the VEM basis functions by the number of the vertex on which they are equal to one. That is, if $\phi^i(\boldsymbol{x}_{k,l}) = 1$, then $\phi^i = \phi^{k,l}$. Using Proposition 3.20, we know our basis functions explicitly. We calculate the consistency terms:

$$\begin{aligned} a^K\left(\Pi^\nabla \phi^{i,j}, \Pi^\nabla \phi^{i,j}\right) &= \epsilon + \epsilon^{-1}, \\ a^K\left(\Pi^\nabla \phi^{i,j}, \Pi^\nabla \phi^{i\pm1,j}\right) &= \frac{1}{2}(-\epsilon + \epsilon^{-1}), \\ a^K\left(\Pi^\nabla \phi^{i,j}, \Pi^\nabla \phi^{i,j\pm1}\right) &= \frac{1}{2}(\epsilon - \epsilon^{-1}), \\ a^K\left(\Pi^\nabla \phi^{i,j}, \Pi^\nabla \phi^{i\pm1,j\pm1}\right) &= -\frac{1}{4}(\epsilon + \epsilon^{-1}). \end{aligned}$$

Next, we know from (3.11) that

$$\boldsymbol{Q} = \sqrt{\frac{9}{4h_x h_y}} \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}.$$

Hence, the stability terms reads

$$\begin{aligned} s^K\left(\phi^{i,j} - \Pi^\nabla \phi^{i,j}, \phi^{i,j} - \Pi^\nabla \phi^{i,j}\right) &= w(\epsilon + \epsilon^{-1}), \\ s^K\left(\phi^{i,j} - \Pi^\nabla \phi^{i,j}, \phi^{i\pm1,j} - \Pi^\nabla \phi^{i\pm1,j}\right) &= -\frac{1}{2}w(\epsilon + \epsilon^{-1}), \\ s^K\left(\phi^{i,j} - \Pi^\nabla \phi^{i,j}, \phi^{i,j\pm1} - \Pi^\nabla \phi^{i,j\pm1}\right) &= -\frac{1}{2}w(\epsilon + \epsilon^{-1}), \\ s^K\left(\phi^{i,j} - \Pi^\nabla \phi^{i,j}, \phi^{i\pm1,j\pm1} - \Pi^\nabla \phi^{i\pm1,j\pm1}\right) &= \frac{1}{4}w(\epsilon + \epsilon^{-1}). \end{aligned}$$

Summing the consistency and stability terms, we see that we the VEM left-hand side reads

$$\begin{aligned}(1+w)\big((\epsilon+\epsilon^{-1})u_{i,j} - \frac{1}{2}\epsilon(u_{i-1,j}+u_{i+1,j}) - \frac{1}{2}\epsilon^{-1}(u_{i,j-1}+u_{i,j+1})\big)\\ +\frac{1}{2}(1-w)\bigg(-\frac{1}{2}(\epsilon+\epsilon^{-1})(u_{i-1,j-1}+u_{i+1,j-1}+u_{i-1,j+1}+u_{i+1,j+1})\\ +\epsilon(u_{i,j-1}+u_{i,j+1})+\epsilon^{-1}(u_{i-1,j}+u_{i+1,j})\bigg).\end{aligned}$$

Using that $(1+w) = 2(w+(1-w)/2)$, we can write this as

$$\begin{aligned}-w\big(\epsilon(u_{i-1,j}-2u_{i,j}+u_{i+1,j})+\epsilon^{-1}(u_{i,j-1}-2u_{i,j}+u_{i,j+1})\big)\\ -(1-w)\bigg(\frac{1}{4}(\epsilon+\epsilon^{-1})(u_{i-1,j-1}+u_{i+1,j-1}-4u_{i,j}+u_{i-1,j+1}+u_{i+1,j+1})\\ -\frac{1}{2}(\epsilon-\epsilon^{-1})(u_{i,j-1}-u_{i-1,j}-u_{i+1,j}+u_{i,j+1})\bigg).\end{aligned}$$

Now, we clearly recognize the first terms as equation (3.23) multiplied by w, while we recognize the second term as equation (3.24) multiplied by $(1-w)$. It follows that equation (3.22) holds.

For the Poisson equation (3.21), if f is a constant function, we see that the right-hand side of the two methods are equal as well, and the two methods are equivalent. □

Since we can scale the stability term s^K by any positive constant, any $w > 0$ gives a perfectly valid virtual element method. However, from a finite difference point of view, only the choices of w with $0 \leq w \leq 1$ makes sense. We also note that the choice $w = 0$ corresponds to $s^K \equiv 0$, resulting in an unstable virtual element method. Hence, Stencil 2 is *not* stable, and must be combined with Stencil 1. Thus, we have found an alternative way of proving convergence of the finite difference scheme.

Finally, we observe that the exact stability term is obtained with $w = 1/3$. Hence, using Theorem 3.21 we can conclude that with $w = 1/3$, the FEM global stiffness matrix for (3.21) on $\mathcal{T}_{h_x,h_y}$ is equal to

$$\frac{1}{3}\boldsymbol{F}_c + \frac{2}{3}\boldsymbol{F}_d.$$

3.3.2 Three Dimensions

Again, we consider the Poisson equation (3.21), with $\Omega \subset \mathbb{R}^3$ being a cubical domain. We will consider three different stencils for the Laplace operator, which arise from eight different coordinate systems, defined in Definition B.2. These stencils are inspired by the ones presented in [26], which are used to solve visco-acoustic wave propagation problems. The stencils using the coordinate systems (x,y,z) and (x,η_x,ζ_x) are shown in Figure 3.11a and Figure 3.11b, while the average stencil using the coordinate systems (d_1,d_2,d_3), (d_1,d_2,d_4), (d_1,d_3,d_4)

and (d_2, d_3, d_4) is shown in Figure 3.11c. Note that the z-axis points downwards, as this is the standard in subsurface modelling.

Again, we have that if f is a constant, the right-hand side of these finite difference methods coincide with the right-hand side of the first order virtual element method. Indeed; multiplying both sides of the equation obtained from the finite difference discretizations of equation (3.21) by $8h_xh_yh_z$, gives the right-hand side $8h_xh_yh_zf$. Since f is a constant function, we also have, using Proposition 3.20, that

$$\langle f_h, \phi^i \rangle = (\Pi_0 f, \phi^i)_{0,\Omega} = 8h_xh_yh_zf$$

for all basis functions ϕ^i with compact support inside Ω.

We now have the following theorem:

Theorem 3.29. *Let $\Omega = [-H_x, H_x]\times[-H_y, H_y]\times[-H_z, H_z] \subset \mathbb{R}^3$, and let $\mathcal{T}_{h_x,h_y,h_z}$ be a regular Cartesian grid over Ω. Let $\boldsymbol{F}_c$, $\boldsymbol{F}_{\boldsymbol{x}}$ and $\boldsymbol{F}_d$ be the finite difference matrices for the discrete Laplacian, as defined in Definition B.2. Further, let $\boldsymbol{A}$ be the global stiffness matrix obtained from the first order virtual element method on $\mathcal{T}_{h_x,h_y,h_z}$, with local stability terms s^K given by*

$$s^K(u, v) = \hat{\boldsymbol{u}}^T \boldsymbol{Q}\boldsymbol{P}^{-1}\tilde{\boldsymbol{\Lambda}}\boldsymbol{P}^{-1}\boldsymbol{Q}^T\hat{\boldsymbol{v}}, \tag{3.25}$$

where the matrix $\boldsymbol{Q}$ is given in (3.13), $\boldsymbol{P} = \boldsymbol{Q}^T\boldsymbol{Q}$, and $\tilde{\boldsymbol{\Lambda}}$ is the scaled eigenvalue matrix, with diagonal entries

$$\begin{aligned}
\tilde{\lambda}_1 &= 3(3w_1 + 2w_2)\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right),\\
\tilde{\lambda}_2 &= 3(3w_1 + 2w_2)\left(\frac{1}{h_x^2} + \frac{1}{h_z^2}\right),\\
\tilde{\lambda}_3 &= 3(3w_1 + 2w_2)\left(\frac{1}{h_y^2} + \frac{1}{h_z^2}\right),\\
\tilde{\lambda}_4 &= 3(9w_1 + 3w_2 + 9w_3)\left(\frac{1}{h_x^2} + \frac{1}{h_y^2} + \frac{1}{h_z^2}\right),
\end{aligned}$$

with $0 \leq w_1, w_2, w_3 \leq 1$, $w_1 + w_2 + w_3 = 1$. Then, we have that

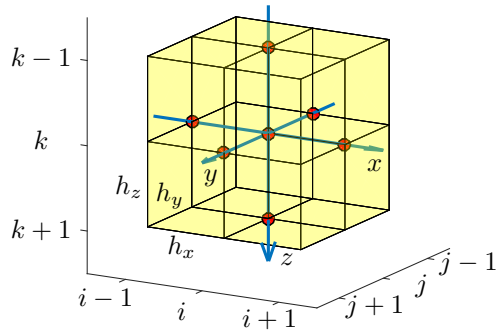
$$w_1\boldsymbol{F}_c + w_2\boldsymbol{F}_{\boldsymbol{x}} + w_3\boldsymbol{F}_d = \boldsymbol{A}. \tag{3.26}$$

For the Poisson equation (3.21), if f is a constant function, the two methods are equivalent.

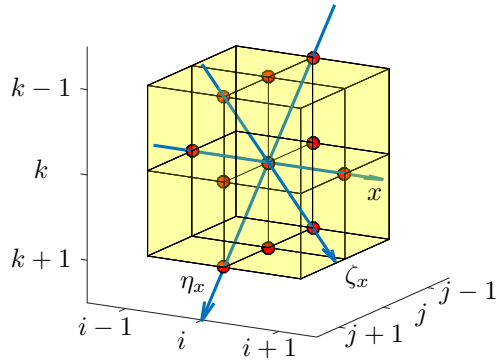
Again, we notice the similarity between (3.25) and the exact stability term.

Proof. We start by writing out the equations obtained using the three different stencils. For Stencil 1, we get the equation

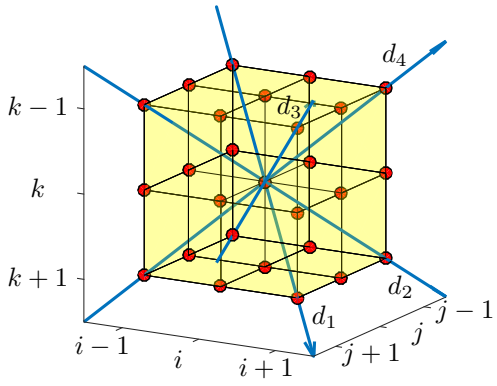
$$\begin{aligned}
-2h_z\epsilon_{yx}(u_{i-1,j,k} - 2u_{i,j,k} + u_{i+1,j,k}) - 2h_x\epsilon_{zy}(u_{i,j-1,k} - 2u_{i,j,k} + u_{i,j+1,k})\\
- 2h_y\epsilon_{xz}(u_{i,j,k-1} - 2u_{i,j,k} + u_{i,j,k+1}) = 8h_xh_yh_zf_{i,j,k}.
\end{aligned} \tag{3.27}$$



(a) Stencil 1.



(b) Stencil 2.



(c) Stencil 3.

Figure 3.11: The three 3D finite difference stencils, along with their coordinate axes.

Stencil 2 yields

$$
\begin{aligned}
&-\frac{1}{3}h_x\Big(2\epsilon_{zy}(u_{i,j-1,k}-2u_{i,j,k}+u_{i,j+1,k})\\
&\qquad+\frac{1}{2}(\epsilon_{zy}+\epsilon_{zy}^{-1})(u_{i,j-1,k-1}+u_{i,j-1,k+1}-4u_{i,j,k}+u_{i,j+1,k-1}+u_{i,j+1,k+1})\\
&\qquad-(\epsilon_{zy}-\epsilon_{zy}^{-1})(u_{i,j,k-1}-u_{i,j-1,k}-u_{i,j+1,k}+u_{i,j,k+1})\Big)\\
&-\frac{1}{3}h_y\Big(2\epsilon_{xz}(u_{i,j,k-1}-2u_{i,j,k}+u_{i,j,k+1})\\
&\qquad+\frac{1}{2}(\epsilon_{xz}+\epsilon_{xz}^{-1})(u_{i-1,j,k-1}+u_{i-1,j,k+1}-4u_{i,j,k}+u_{i+1,j,k-1}+u_{i+1,j,k+1})\\
&\qquad-(\epsilon_{xz}-\epsilon_{xz}^{-1})(-u_{i,j,k-1}+u_{i-1,j,k}+u_{i+1,j,k}-u_{i,j,k+1})\Big)\\
&-\frac{1}{3}h_z\Big(2\epsilon_{yx}(u_{i-1,j,k}-2u_{i,j,k}+u_{i+1,j,k})\\
&\qquad+\frac{1}{2}(\epsilon_{yx}+\epsilon_{yx}^{-1})(u_{i-1,j-1,k}+u_{i-1,j+1,k}-4u_{i,j,k}+u_{i+1,j-1,k}+u_{i+1,j+1,k})\\
&\qquad-(\epsilon_{yx}-\epsilon_{yx}^{-1})(u_{i,j-1,k}-u_{i-1,j,k}-u_{i+1,j,k}+u_{i,j+1,k})\Big)=8h_xh_yh_zf_{i,j,k}.
\end{aligned}
\tag{3.28}
$$

Finally, Stencil 3 yields

$$
\begin{aligned}
&-\frac{1}{4}(h_x\epsilon_{zy}+h_y\epsilon_{xz}+h_z\epsilon_{yx})\\
&\quad(u_{i-1,j-1,k-1}+u_{i-1,j+1,k-1}+u_{i-1,j-1,k+1}+u_{i-1,j+1,k+1}-8u_{i,j,k}\\
&\qquad+u_{i+1,j-1,k-1}+u_{i+1,j+1,k-1}+u_{i+1,j-1,k+1}+u_{i+1,j+1,k+1})\\
&+\frac{1}{2}h_x\epsilon_{zy}(u_{i-1,j,k-1}+u_{i-1,j,k+1}-2u_{i,j-1,k}-2u_{i,j+1,k}+u_{i+1,j,k-1}+u_{i+1,j,k+1})\\
&+\frac{1}{2}h_y\epsilon_{xz}(u_{i-1,j-1,k}+u_{i-1,j+1,k}-2u_{i,j,k-1}-2u_{i,j,k+1}+u_{i+1,j-1,k}+u_{i+1,j+1,k})\\
&+\frac{1}{2}h_z\epsilon_{yx}(u_{i,j-1,k-1}+u_{i,j-1,k+1}-2u_{i-1,j,k}-2u_{i+1,j,k}+u_{i,j+1,k-1}+u_{i,j+1,k+1})\\
&\qquad=8h_xh_yh_zf_{i,j,k}.
\end{aligned}
\tag{3.29}
$$

As in the proof of Theorem 3.28, we temporarily rename the VEM basis functions so that if $\phi^i(\boldsymbol{x}_{k,l,m})=1$, then $\phi^i=\phi^{k,l,m}$. We calculate the consistency terms:

$$
\begin{aligned}
a^K(\Pi^\nabla\phi^{i,j,k},\Pi^\nabla\phi^{i,j,k}) &= h_x\epsilon_{zy}+h_y\epsilon_{xz}+h_z\epsilon_{yx},\\
a^K(\Pi^\nabla\phi^{i,j,k},\Pi^\nabla\phi^{i\pm1,j,k}) &= \frac{1}{2}(h_x\epsilon_{zy}+h_y\epsilon_{xz}-h_z\epsilon_{yx}),\\
a^K(\Pi^\nabla\phi^{i,j,k},\Pi^\nabla\phi^{i,j\pm1,k}) &= \frac{1}{2}(-h_x\epsilon_{zy}+h_y\epsilon_{xz}+h_z\epsilon_{yx}),
\end{aligned}
$$

$$
\begin{aligned}
a^K(\Pi^\nabla\phi^{i,j,k}, \Pi^\nabla\phi^{i,j,k\pm1}) &= \frac{1}{2}(h_x\epsilon_{zy} - h_y\epsilon_{xz} + h_z\epsilon_{yx}),\\
a^K(\Pi^\nabla\phi^{i,j,k}, \Pi^\nabla\phi^{i\pm1,j\pm1,k}) &= \frac{1}{4}(-h_x\epsilon_{zy} + h_y\epsilon_{xz} - h_z\epsilon_{yx}),\\
a^K(\Pi^\nabla\phi^{i,j,k}, \Pi^\nabla\phi^{i\pm1,j,k\pm1}) &= \frac{1}{4}(h_x\epsilon_{zy} - h_y\epsilon_{xz} - h_z\epsilon_{yx}),\\
a^K(\Pi^\nabla\phi^{i,j,k}, \Pi^\nabla\phi^{i,j\pm1,k\pm1}) &= \frac{1}{4}(-h_x\epsilon_{zy} - h_y\epsilon_{xz} + h_z\epsilon_{yx}),\\
a^K(\Pi^\nabla\phi^{i,j,k}, \Pi^\nabla\phi^{i\pm1,j\pm1,k\pm1}) &= -\frac{1}{8}(h_x\epsilon_{zy} + h_y\epsilon_{xz} + h_z\epsilon_{yx}).
\end{aligned}
$$

Next, we know from (3.13) that

$$
\boldsymbol{Q} = \sqrt{\frac{9}{8h_xh_yh_z}}\begin{bmatrix}
1 & 1 & 1 & -\sqrt{3}\\
-1 & -1 & 1 & \sqrt{3}\\
1 & -1 & -1 & -\sqrt{3}\\
-1 & 1 & -1 & \sqrt{3}\\
1 & -1 & -1 & \sqrt{3}\\
-1 & 1 & -1 & -\sqrt{3}\\
1 & 1 & 1 & \sqrt{3}\\
-1 & -1 & 1 & -\sqrt{3}
\end{bmatrix}.
$$

For simplicity, we write

$$
s^K_{l,m,n} = s^K\Big(\phi^{i,j,k} - \Pi^\nabla\phi^{i,j,k}, \phi^{l,m,n} - \Pi^\nabla\phi^{l,m,n}\Big),
$$

and obtain the stability terms

$$
\begin{aligned}
s^K_{i,j,k} &= \left(3w_1 + \frac{5}{3}w_2 + w_3\right)(h_x\epsilon_{zy} + h_y\epsilon_{xz} + h_z\epsilon_{yx})\\
s^K_{i\pm1,j,k} &= -\frac{1}{2}w_1(h_x\epsilon_{zy} + h_y\epsilon_{xz} + 3h_z\epsilon_{yx}) - \frac{1}{6}w_2(h_x\epsilon_{zy} + h_y\epsilon_{xz} + 5h_z\epsilon_{yx})\\
&\quad -\frac{1}{2}w_3(h_x\epsilon_{zy} + h_y\epsilon_{xz} + h_z\epsilon_{yx}),\\
s^K_{i,j\pm1,k} &= -\frac{1}{2}w_1(3h_x\epsilon_{zy} + h_y\epsilon_{xz} + h_z\epsilon_{yx}) - \frac{1}{6}w_2(5h_x\epsilon_{zy} + h_y\epsilon_{xz} + h_z\epsilon_{yx})\\
&\quad -\frac{1}{2}w_3(h_x\epsilon_{zy} + h_y\epsilon_{xz} + h_z\epsilon_{yx}),\\
s^K_{i,j,k\pm1} &= -\frac{1}{2}w_1(h_x\epsilon_{zy} + 3h_y\epsilon_{xz} + h_z\epsilon_{yx}) - \frac{1}{6}w_2(h_x\epsilon_{zy} + 5h_y\epsilon_{xz} + h_z\epsilon_{yx})\\
&\quad -\frac{1}{2}w_3(h_x\epsilon_{zy} + h_y\epsilon_{xz} + h_z\epsilon_{yx}),\\
s^K_{i\pm1,j\pm1,k} &= \frac{1}{4}w_1(h_x\epsilon_{zy} - h_y\epsilon_{xz} + h_z\epsilon_{yx}) + \frac{1}{12}w_2(h_x\epsilon_{zy} - 3h_y\epsilon_{xz} + h_z\epsilon_{yx})\\
&\quad +\frac{1}{4}w_3(h_x\epsilon_{zy} + h_y\epsilon_{xz} + h_z\epsilon_{yx}),
\end{aligned}
$$

$$
\begin{aligned}
s^K_{i\pm1,j,k\pm1} = \;& \frac{1}{4}w_1(-h_x\epsilon_{zy} + h_y\epsilon_{xz} + h_z\epsilon_{yx}) \\
&+ \frac{1}{12}w_2(-3h_x\epsilon_{zy} + h_y\epsilon_{xz} + h_z\epsilon_{yx}) + \frac{1}{4}w_3(h_x\epsilon_{zy} + h_y\epsilon_{xz} + h_z\epsilon_{yx}), \\
s^K_{i,j\pm1,k\pm1} = \;& \frac{1}{4}w_1(h_x\epsilon_{zy} + h_y\epsilon_{xz} - h_z\epsilon_{yx}) + \frac{1}{12}w_2(h_x\epsilon_{zy} + h_y\epsilon_{xz} - 3h_z\epsilon_{yx}) \\
&+ \frac{1}{4}w_3(h_x\epsilon_{zy} + h_y\epsilon_{xz} + h_z\epsilon_{yx}), \\
s^K_{i\pm1,j\pm1,k\pm1} = \;& \frac{1}{8}(w_1 + w_2 - w_3)(h_x\epsilon_{zy} + h_y\epsilon_{xz} + h_z\epsilon_{yx}).
\end{aligned}
$$

Now, setting $w_1 = 1$ and summing the consistency and stability terms yields the VEM left-hand side

$$
\begin{aligned}
2h_z\epsilon_{yx}(-u_{i-1,j,k} + 2u_{i,j,k} - u_{i+1,j,k}) + 2h_x\epsilon_{zy}(-u_{i,j-1,k} + 2u_{i,j,k} - u_{i,j+1,k}) \\
+ 2h_y\epsilon_{xz}(-u_{i,j,k-1} + 2u_{i,j,k} - u_{i,j,k+1}),
\end{aligned}
$$

which is the same as the left-hand side of (3.27).

Next, we set $w_2 = 1$ and sum:

$$
\begin{aligned}
&\frac{8}{3}(h_x\epsilon_{zy} + h_y\epsilon_{xz} + h_z\epsilon_{yx})u_{i,j,k} \\
&+ \frac{1}{3}(h_x\epsilon_{zy} + h_y\epsilon_{xz} - 4h_z\epsilon_{yx})(u_{i-1,j,k} + u_{i+1,j,k}) \\
&+ \frac{1}{3}(-4h_x\epsilon_{zy} + h_y\epsilon_{xz} + h_z\epsilon_{yx})(u_{i,j-1,k} + u_{i,j+1,k}) \\
&+ \frac{1}{3}(h_x\epsilon_{zy} - 4h_y\epsilon_{xz} + h_z\epsilon_{yx})(u_{i,j,k-1} + u_{i,j,k+1}) \\
&- \frac{1}{6}(h_x\epsilon_{zy} + h_z\epsilon_{yx})(u_{i-1,j-1,k} + u_{i-1,j+1,k} + u_{i+1,j-1,k} + u_{i+1,j+1,k}) \\
&- \frac{1}{6}(h_y\epsilon_{xz} + h_z\epsilon_{yx})(u_{i-1,j,k-1} + u_{i-1,j,k+1} + u_{i+1,j,k-1} + u_{i+1,j,k+1}) \\
&- \frac{1}{6}(h_x\epsilon_{zy} + h_y\epsilon_{xz})(u_{i,j-1,k-1} + u_{i,j-1,k+1} + u_{i,j+1,k-1} + u_{i,j+1,k+1}).
\end{aligned}
\tag{3.30}
$$

We start by noticing that $h_x\epsilon_{zy} = \frac{h_xh_z}{h_y} = h_z\epsilon_{yx}^{-1}$, so that we can write

$$
h_x\epsilon_{zy} = \alpha h_x\epsilon_{zy} + (1-\alpha)h_z\epsilon_{yx}^{-1}, \quad \alpha \in \mathbb{R}\,. \tag{3.31}
$$

Similar relations holds for $h_y\epsilon_{xz}$ and $h_z\epsilon_{yx}$. Summing the terms involving $u_{i,j,k}$ in equation (3.28) yields

$$
\frac{2}{3}\big(h_x(3\epsilon_{zy} + \epsilon_{zy}^{-1}) + h_y(3\epsilon_{xz} + \epsilon_{xz}^{-1}) + h_z(3\epsilon_{yx} + \epsilon_{yx}^{-1})\big).
$$

Using relation (3.31) with $\alpha = 3/4$ for all terms in (3.30) involving $u_{i,j,k}$, we obtain

$$
\frac{8}{3}(h_x\epsilon_{zy} + h_y\epsilon_{xz} + h_z\epsilon_{yx}) = \frac{2}{3}\big(h_x(3\epsilon_{zy} + \epsilon_{zy}^{-1}) + h_y(3\epsilon_{xz} + \epsilon_{xz}^{-1}) + h_z(3\epsilon_{yx} + \epsilon_{yx}^{-1})\big),
$$

and we see that the terms multiplied by $u_{i,j,k}$ are the same in both expressions. Next, summing the terms in (3.28) involving $u_{i\pm1,j,k}$ yields

$$\frac{1}{3}\big(h_y(\epsilon_{xz}-\epsilon_{xz}^{-1})-h_z(3\epsilon_{yx}-\epsilon_{yx}^{-1})\big)$$

while summing the terms (3.30) involving $u_{i\pm1,j,k}$ yields

$$\frac{1}{3}\big(h_x\epsilon_{zy}+h_y\epsilon_{xz}-4h_z\epsilon_{yx}\big).$$

Now, using relation (3.31) with $\alpha=0$ for the first term, and $\alpha=3/4$ for the last term, we obtain

$$\frac{1}{3}\big(h_z\epsilon_{yx}^{-1}+h_y\epsilon_{xz}-3h_z\epsilon_{yx}-h_y\epsilon_{xz}^{-1}\big)=\frac{1}{3}\big(h_y(\epsilon_{xz}-\epsilon_{xz}^{-1})-h_z(3\epsilon_{yx}-\epsilon_{yx}^{-1})\big),$$

from which we see that the terms involving $u_{i\pm1,j,k}$ are also the same in both expressions. The same applies to the terms involving $u_{i,j\pm1,k}$ and $u_{i,jk\pm1}$. Moreover, we immediately see that the terms involving $u_{i\pm1,j\pm1,k}$, $u_{i\pm1,j,k\pm1}$ and $u_{i,j\pm1,k\pm1}$ are the same in both expressions. We thus have that equation (3.30) equals the left-hand side of (3.28).

Finally, setting $w_3=1$ yields

$$\begin{aligned}
&2(h_x\epsilon_{zy}+h_y\epsilon_{xz}+h_z\epsilon_{yx})u_{i,j,k}-h_z\epsilon_{yx}(u_{i-1,j,k}+u_{i+1,j,k})\\
&\quad-h_x\epsilon_{zy}(u_{i,j-1,k}+u_{i,j+1,k})-h_y\epsilon_{xz}(u_{i,j,k-1}+u_{i,j,k-1})\\
&+\frac{1}{2}h_y\epsilon_{xz}(u_{i-1,j-1,k}+u_{i-1,j+1,k}+u_{i+1,j-1,k}+u_{i+1,j+1,k})\\
&+\frac{1}{2}h_x\epsilon_{zy}(u_{i-1,j,k-1}+u_{i-1,j,k+1}+u_{i+1,j,k-1}+u_{i+1,j,k+1})\\
&+\frac{1}{2}h_z\epsilon_{yx}(u_{i,j-1,k-1}+u_{i,j-1,k+1}+u_{i,j+1,k-1}+u_{i,j+1,k+1})\\
&-\frac{1}{4}(h_x\epsilon_{zy}+h_y\epsilon_{xz}+h_z\epsilon_{yx})(u_{i-1,j-1,k-1}+u_{i-1,j+1,k-1}+u_{i-1,j-1,k+1}\\
&\quad+u_{i-1,j+1,k+1}+u_{i+1,j-1,k-1}+u_{i+1,j+1,k-1}+u_{i+1,j-1,k+1}+u_{i+1,j+1,k+1}),
\end{aligned}$$

which we see equals the left-hand side of equation (3.29).

Now, we have that Stencil i can be obtained from VEM by setting $w_i=1$, and the other weights equal to zero. Moreover, we note that stability term s^K is linear in the scaled eigenvalue matrix $\tilde{\boldsymbol{\Lambda}}$ in the following sense: Let s_1^K and s_2^K be two different stability terms, differing in their scaled eigenvalue matrices $\tilde{\boldsymbol{\Lambda}}_1$ and $\tilde{\boldsymbol{\Lambda}}_2$, which are obtained from different choices of weights (w_1,w_2,w_3). Then

$$\begin{aligned}
s_1^K(u,v)+s_2^K(u,v)&=\hat{\boldsymbol{u}}^T\boldsymbol{Q}\boldsymbol{P}^{-1}\tilde{\boldsymbol{\Lambda}}_1\boldsymbol{P}^{-1}\boldsymbol{Q}^T\hat{\boldsymbol{v}}+\hat{\boldsymbol{u}}^T\boldsymbol{Q}\boldsymbol{P}^{-1}\tilde{\boldsymbol{\Lambda}}_2\boldsymbol{P}^{-1}\boldsymbol{Q}^T\hat{\boldsymbol{v}}\\
&=\hat{\boldsymbol{u}}^T\boldsymbol{Q}\boldsymbol{P}^{-1}\big(\tilde{\boldsymbol{\Lambda}}_1+\tilde{\boldsymbol{\Lambda}}_2\big)\boldsymbol{P}^{-1}\boldsymbol{Q}^T\hat{\boldsymbol{v}}.
\end{aligned}$$

From this, we conclude that any weighted average of the finite difference Stencils 1, 2 and 3 can be obtained from a first order virtual element method, with

stability term given by (3.25). It follows that (3.26) holds.

For the Poisson equation (3.21), if f is a constant function, we see that both methods produces the same right-hand side as well, and are thus equivalent. □

Note that any combination of the weights w_1, w_2 and w_3 such that the diagonal elements of $\tilde{\mathbf{\Lambda}}$ are strictly positive gives a valid virtual element method. However, only the choices with $0 \leq w_1, w_2, w_3 \leq 1$ such that $w_1 + w_2 + w_3 = 1$ makes sense from a finite difference point of view. We also observe that we cannot have both w_1 and w_2 equal to zero, since this gives a scaled eigenvalue matrix $\tilde{\mathbf{\Lambda}}$ which is not positive definite, so that s^K fails to be positive definite on $\ker \Pi^\nabla$, and the method is not stable. In other words, Stencil 3 is not stable, and must be combined with the other two. Again, this gives an alternative way of proving convergence of the finite difference scheme.

Finally, we observe that any choice of weights such that

$$3w_1 + 2w_2 = 1, \quad 9w_1 + 3w_2 + 9w_3 = 1, \tag{3.32}$$

will give the exact stability term. From Theorem 3.21, we can then conclude that the FEM global stiffness matrix for (3.21) on $\mathcal{T}_{h_x,h_y,h_z}$ is equal to (3.26), with the weights satisfying (3.32).

3.4 Final Remarks

The results presented here are interesting for several reasons. First of all, it narrows down the number of parameters involved in choosing the stability term. Moreover, once we have established equivalence between VEM and another method, we can conclude that the virtual element method possesses all the properties of that method. On the other hand, we can also derive useful information of the other method from the properties of VEM, and it gives a natural extension of the method to grids with more general cell geometries.

Regarding the procedure for obtaining the exact stability term in Theorem 3.10, it should be mentioned that the process of determining a basis for $\ker \Pi^\nabla$ is essentially the same as explicitly determining the VEM basis functions. Indeed; for a function $v_h \in V_k(K)$, we have that

$$v_h = \Pi^\nabla v_h + \left(v_h - \Pi^\nabla v_h\right).$$

As we mentioned in Chapter 2, we are able to compute $\Pi^\nabla v_h$ exactly. Moreover, it is clear that $v_h - \Pi^\nabla v_h \in \ker \Pi^\nabla$. Hence, if we determine a basis for $\ker \Pi^\nabla$, we are also able to compute the last term exactly. Thus, we can calculate v_h explicitly, and there is nothing virtual left in the virtual element method. Moreover, in many cases, it might be very hard to determine an L^2-orthonormal basis of $\ker \Pi^\nabla$. However, the theory presented here might be a good starting point for choosing the stability term for more complex cell geometries as well.

Chapter 4

Implementing VEM for Poisson Problems

All you really need to know for the moment is that the universe is a lot more complicated than you might think, even if you start from a position of thinking it's pretty damn complicated in the first place.

Douglas Adams, *Mostly Harmless*

We now have the necessary framework to implement the virtual element method. However, as the reader might experience, the transition from theory to a fully working implementation is not straightforward.

Section 4.2 to Section 4.5 describes in detail how to compute the local stiffness matrix for Poisson problems in two and three dimensions, how to compute the L^2 projection, and how to compute the local load vector. These sections are mainly based on what is found in [34], supplemented by remarks to emphasize important aspects of the virtual element method. Moreover, we provide a detailed proof of that the projection Π^∇ can be calculated for any function in the local virtual element space using its degrees of freedom. We also present a way of constructing the local load term in order to reduce the use of numerical integration. Finally, we describe the assembly procedure of the global stiffness matrix, and discuss how to implement inhomogeneous boundary conditions.

Before we continue, it is useful to recall Definition 2.13 and Definition 2.29 of the local virtual element spaces, and Definition 2.14 and Definition 2.30 of their degrees of freedom.

4.1 Numerical Integration

In Chapter 2, we rely on that we are able to calculate integrals of polynomials over polygons and polyhedra. As we will see, we will also have to calculate

integrals of the source term f. Hence, before we start describing the implementation process, a short discussion of numerical integration techniques is in order.

Let Ω be a subset of $\mathbb{R}^d$, and let v be a continuous function. We will consider numerical quadrature rules on the form

$$\int_\Omega v \, \mathrm{d}\boldsymbol{x} \approx |\Omega| \sum_{i=1}^{n} w_i v(\boldsymbol{x}^i),$$

where $w_i \in \mathbb{R}$ are quadrature weights, $\boldsymbol{x}^i \in \Omega$ are quadrature points, and n is the number of quadrature points. Notice that if v is a constant, a consistent quadrature rule necessarily needs to have $\sum_i w_i = 1$. If a quadrature rule is exact for polynomials of degree less than or equal to k, we say that the quadrature rule is of *precision* k. There exists a huge variety of numerical quadrature rules, and it is difficult to decide which is better for a particular purpose. See, for example [29, 30] for a description of quadrature rules used in finite element methods, and [37] for construction of efficient quadrature rules for more general polygons and polyhedra.

Remark 4.1. Recall from Section 2.5 that we have to calculate integrals of the type

$$(\partial_n p, \phi^i)_{0,\partial K}, \quad p \in \mathbb{P}_k(K).$$

According to Remark 2.15, we can choose $\mathcal{E}^K$ to be the values at $k-1$ points at each edge of K. As mentioned in [33, 34], a particularly suited quadrature rule is then the Lobatto quadrature rule [29, Chapter 4.1.3], since it uses the endpoints of the interval as quadrature points, along with $k-1$ more points. Thus, choosing the $k-1$ points of $\mathcal{E}^K$ to match the Lobatto quadrature points, one can use the degrees of freedom $\mathcal{V}^K$ and $\mathcal{E}^K$ to evaluate ϕ^i at the quadrature points. Moreover, a Lobatto quadrature rule using $k+1$ points is of precision $2k-1$. Letting E be an edge of K, we have that p and $\phi^i|_E$ are polynomials in $\mathbb{P}_k(E)$, so that that $\phi^i \partial_n p|_E \in \mathbb{P}_{2k-1}(E)$. In other words, the quadrature rule is exact. ◦

We will consider two techniques of numerical integration:

4.1.1 Using the Divergence Theorem

Consider a polynomial $\boldsymbol{x}^{\boldsymbol{\alpha}}$, defined on a domain $K \subset \mathbb{R}^d$. This can be written as

$$\boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d} = \frac{1}{\alpha_m + 1} \nabla \cdot (0, \ldots, 0, x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_m^{\alpha_m + 1} \cdots x_d^{\alpha_d}, 0, \ldots, 0).$$

Using the divergence theorem, we have that

$$\begin{aligned}\int_K \boldsymbol{x}^{\boldsymbol{\alpha}} \,\mathrm{d}\boldsymbol{x} &= \frac{1}{\alpha_m+1}\int_K \nabla\cdot(0,\dots,0,x_1^{\alpha_1}x_2^{\alpha_2}\cdots x_m^{\alpha_m+1}\cdots x_d^{\alpha_d},0,\dots,0)\,\mathrm{d}\boldsymbol{x}\\ &= \frac{1}{\alpha_m+1}\int_{\partial K}(x_1^{\alpha_1}x_2^{\alpha_2}\cdots x_m^{\alpha_m+1}\cdots x_d^{\alpha_d})n_m\,\mathrm{d}s,\end{aligned}$$

where n_m is the m-th component of the normal vector of ∂K. In $\mathbb{R}^2$, if K is a polygon, we can calculate the integral using a suitable quadrature rule on each edge of the polygon.

In $\mathbb{R}^3$, if K is a polyhedron, we can apply the trick once to transform the volume integral to a sum of surface integrals, and once more to reduce it to a sum of line integrals. However, when applying the trick for the second time, one must be careful when choosing the variable x_m: The trick is not valid if we are on a face perpendicular to the x_m-axis, since then, x_m is constant over the domain of integration.

4.1.2 Mapping from a reference element

We will also need to evaluate the integrals of more general functions over polygons and polyhedra. Since the anti-derivative of a function is not necessarily easy to calculate, we will need to apply a more flexible method than the one described above.

The idea is to construct a suitable quadrature rule for a reference element T_r, which is a triangle in $\mathbb{R}^2$, or a tetrahedron in $\mathbb{R}^3$. Then, we partition, or triangulate, our polygon or polyhedron into a set $\{T_i\}_i$ of non-overlapping triangles or tetrahedra. Denoting the polygon or polyhedron by K, we have that $\int_K v\,\mathrm{d}\boldsymbol{x} = \sum_i \int_{T_i} v\,\mathrm{d}\boldsymbol{x}$. If we let G be the affine mapping from T_r to T_i, we can use change of variables, along with the quadrature rule for T_r, to obtain

$$\begin{aligned}\int_{T_i} v\,\mathrm{d}\boldsymbol{x} &= |\det(DG)|\int_{T_r} v\circ G\,\mathrm{d}\boldsymbol{x}_r\\ &\approx |\det(DG)||T_r|\sum\nolimits_j w_j(v\circ G)(\boldsymbol{x}_r^j).\end{aligned}$$

As usual, DG is the Jacobian of G, and $\boldsymbol{x}_r$ are the coordinates on T_r.

4.2 Computing the 2D local stiffness matrix

We recall the local form of the weak formulation of the Poisson equation (2.3) in matrix form:

$$\boldsymbol{A}^K\boldsymbol{\xi} = \boldsymbol{b}^K.$$

The first step in assembling the virtual element method is to construct the local stiffness matrix $\boldsymbol{A}^K$. We consider a polygon K, with n vertices, like the polygon in Figure 4.1.

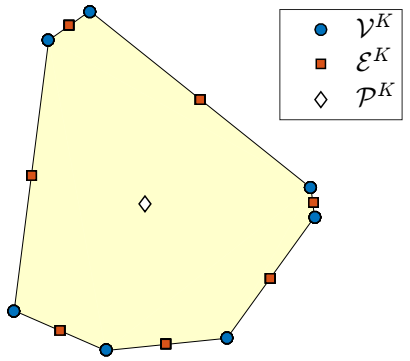


Figure 4.1: Example of a polygon K, with the degrees of freedom for $V_2(K)$ indicated.

We recall that the dimension of $\mathcal{M}_k(K)$ for a polygon is $n_k = (k+1)(k+2)/2$. Moreover, from the discussion in Section 2.5, we recall that $\boldsymbol{A}^K_{i,j} = a^K_h\left(\phi^i, \phi^j\right)$, and that

$$a^K_h\left(\phi^i, \phi^j\right) = a^K\left(\Pi^\nabla \phi^i, \Pi^\nabla \phi^j\right) + s^K\left(\phi^i - \Pi^\nabla \phi^i, \phi^j - \Pi^\nabla \phi^j\right). \tag{4.1}$$

Since $\Pi^\nabla \phi^i \in \mathbb{P}_k(K)$, we can express this in the monomial basis. This will give us the first term of (4.1), since we can easily compute $a^K\left(m^\alpha, m^\beta\right)$ for two monomials m^α and m^β. Moreover, since $\mathbb{P}_k(K) \subset V_k(K)$, we can always express $\Pi^\nabla \phi^i$ in the canonical basis $\{\phi^i\}_i$. As we will see, this gives us the second term. Hence, if we obtain matrix representations of Π^∇ in the monomial basis and the canonical basis, we have our matrix representation $\boldsymbol{A}^K$ of a^K_h.

We start by computing the matrix representation of the projection operator Π^∇ in the monomial basis. We know that $\mathcal{M}_k(K)$ is a basis for $\mathbb{P}_k(K)$, so that equation (2.12) in the definition of Π^∇ can be written as

$$a^K\left(v_h - \Pi^\nabla v_h, m^\alpha\right) = 0, \quad \alpha = 1, \dots, n_k.$$

Since $\Pi^\nabla v_h \in \mathbb{P}_k(K)$, we can write $\Pi^\nabla v_h = \sum_\beta \pi_\beta m^\beta$, where $\pi_\beta \in \mathbb{R}$ for all β. This yields

$$\sum_{\beta=1}^{n_k} \pi_\beta a^K(m^\beta, m^\alpha) = a^K(m^\alpha, v_h), \quad \alpha = 1, \dots, n_k.$$

In our case, $a^K(u, v) = (\nabla u, \nabla v)_{0,K}$. Hence, $\alpha = 1$ gives us a degenerate equation. The extra equation needed for a fully determined system can be obtained

from equation (2.13):

$$\sum_{\beta=1}^{n_k} \pi_\beta \int_{\partial K} m^\beta \, \mathrm{d}s = \int_{\partial K} v_h \, \mathrm{d}s \quad \text{for } k = 1,$$
$$\sum_{\beta=1}^{n_k} \pi_\beta \int_{K} m^\beta \, \mathrm{d}\boldsymbol{x} = \int_{K} v_h \, \mathrm{d}\boldsymbol{x} \quad \text{for } k \geq 2.$$

For simplicity, we write

$$\bar{g} = \begin{cases} \int_{\partial K} g \, \mathrm{d}s & \text{for } k = 1, \\ \int_K g \, \mathrm{d}\boldsymbol{x} & \text{for } k \geq 2. \end{cases}$$

This gives the system

$$\begin{bmatrix} \bar{m}^1 & \bar{m}^2 & \cdots & \bar{m}^{n_k} \\ 0 & a^K(m^2, m^2) & \cdots & a^K(m^2, m^{n_k}) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a^K(m^{n_k}, m^2) & \cdots & a^K(m^{n_k}, m^{n_k}) \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_{n_k} \end{bmatrix} = \begin{bmatrix} \bar{v}_h \\ a^K(m^1, v_h) \\ \vdots \\ a^K(m^{n_k}, v_h) \end{bmatrix},$$

which we write compactly as

$$\boldsymbol{M}\boldsymbol{\pi} = \boldsymbol{b}. \tag{4.2}$$

It is clear that $\boldsymbol{M}$ is computable, since we can evaluate integrals of polynomials over K. Further, from the discussion in Section 2.5, we know that we are able to compute $\boldsymbol{b}$ from the degrees of freedom. Indeed; we have that

$$a^K(m^\alpha, v_h) = -(\Delta m^\alpha, v_h)_{0,K} + (\partial_n m^\alpha, v_h)_{0,\partial K}.$$

Since $m^\alpha \in \mathbb{P}_k(K)$, we know that $\Delta m^\alpha \in \mathbb{P}_{k-2}(K)$, so that Δm^α can be expressed in terms of the first n_{k-2} monomials:

$$(\Delta m^\alpha, v_h)_{0,K} = \sum_{\beta=1}^{n_{k-2}} \xi_\beta (m^\beta, v_h)_{0,K} = |K| \sum_{\beta=1}^{n_{k-2}} \xi_\beta \chi^{(kn+\beta)}(v_h),$$

where $\xi_\beta \in \mathbb{R}$ for all β, and we recall that K has n edges. For the second term, we recall that $v_h|_E \in \mathbb{P}_k(E)$, so that we can calculate this using a suitable quadrature rule.

For each of the N^K basis functions ϕ^i, we write $\Pi^\nabla \phi^i = \sum_\alpha \pi_\alpha^i m^\alpha$. Further, we replace v_h with ϕ^i in (4.2), and denote the solution to this system by $\boldsymbol{\pi}^i =$

$\left(\pi_1^i, \dots, \pi_{n_k}^i\right)^T$. We write this compactly as $\boldsymbol{\pi}^i = \boldsymbol{M}^{-1}\boldsymbol{b}^i$. Defining

$$\boldsymbol{B} \coloneqq \left[\boldsymbol{b}^1, \dots, \boldsymbol{b}^{N^K}\right] = \begin{bmatrix} \bar{\phi}^1 & \cdots & \bar{\phi}^{N^K} \\ a^K\left(m^2, \phi^1\right) & \cdots & a^K\left(m^2, \phi^{N^K}\right) \\ \vdots & \ddots & \vdots \\ a^K\left(m^{n_k}, \phi^1\right) & \cdots & a^K\left(m^{n_k}, \phi^{N^K}\right) \end{bmatrix}, \tag{4.3}$$

we get the matrix representation $\boldsymbol{\Pi}_*^\nabla$ of the operator Π^∇ in the monomial basis $\mathcal{M}_k(K)$:

$$\boldsymbol{\Pi}_*^\nabla = \boldsymbol{M}^{-1}\boldsymbol{B}. \tag{4.4}$$

Remark 4.2. We can now see why the definition of Π_k^∇ differs for $k = 1$ and $k \geq 2$: For $k \geq 2$ the values $\bar{\phi}^i$ are given from the degrees of freedom $\mathcal{P}^K$. This is not the case for $k = 1$, since then, we only have the degrees of freedom $\mathcal{V}^K$. However, since we know that $\phi^i|_E \in \mathbb{P}_1(E)$ for each edge E of K, we are indeed able to calculate the integral if ϕ^i over ∂K. ◦

For a function $v_h \in V_k(K)$, we use the interpolation identity (2.27) and the linearity of Π^∇ to write

$$\Pi^\nabla v_h = \Pi^\nabla\left(\sum_{i=1}^{N^K} \chi^i(v_h)\phi^i\right) = \sum_{i=1}^{N^K} \chi^i(v_h)\Pi^\nabla\phi^i = \sum_{i=1}^{N^K}\sum_{\alpha=1}^{n_k} \chi^i(v_h)\pi_\alpha^i m^\alpha.$$

As in Chapter 3, we denote the vector of degrees of freedom of v_h by $\hat{\boldsymbol{v}}_h$. Further, we define the vector-valued function $\boldsymbol{m}^{n_k} = (m^1, \dots, m^{n_k})$. Then, we have that

$$\Pi^\nabla v_h = \boldsymbol{m}^{n_k}\boldsymbol{\Pi}_*^\nabla\hat{\boldsymbol{v}}_h.$$

It is now easy to justify the statement made in Remark 2.17 that $\Pi^\nabla v_h$ is computable for any function $v_h \in V_k(K)$ from the degrees of freedom. This is an essential component of the virtual element method, and we state it as a theorem:

Theorem 4.3. *For any function $v_h \in V_k(K)$, we have that $\Pi^\nabla v_h$ can be calculated exactly.*

Proof. We have already shown how to calculate the representation of Π^∇ in the monomial basis through (4.4), where $\boldsymbol{B}$ can be obtained using the degrees of freedom. Therefore, the only thing left to do is to show that the matrix $\boldsymbol{M}$ is, in fact, invertible, so that (4.4) makes sense. Let $\boldsymbol{v} = (v_2, \dots, v_{n_k})^T \in \mathbb{R}^{n_k-1}$ be a vector, and $\hat{\boldsymbol{M}} \in \mathbb{R}^{n_k-1\times n_k-1}$ be the matrix

$$\hat{\boldsymbol{M}}_{\alpha,\beta} = a^K\left(m^{\alpha+1}, m^{\beta+1}\right).$$

If $\hat{\boldsymbol{M}}\boldsymbol{v} = 0$, this means that $\boldsymbol{v}^T\hat{\boldsymbol{M}}\boldsymbol{v} = 0$. From the bilinearity of a^K, we have

$$\begin{aligned}\boldsymbol{v}^T\hat{\boldsymbol{M}}\boldsymbol{v} &= \sum_{\alpha,\beta=2}^{n_k} v_\alpha a^K\big(m^\alpha, m^\beta\big)v_\beta \\ &= a^K\left(\sum_{\alpha=2}^{n_k} v_\alpha m^\alpha, \sum_{\beta=2}^{n_k} v_\beta m^\beta\right) = \left|\sum_{\alpha=2}^{n_k} v_\alpha m^\alpha\right|_{1,K}^2.\end{aligned}$$

Now, we know that $|u|_{1,K} > 0$ for all non-constant functions $u \in H^1(K)$. Hence, if $\hat{\boldsymbol{M}}\boldsymbol{v} = 0$, then $\sum_\alpha v_\alpha m^\alpha$ must be constant. We express this in terms of the constant monomial m^1 as

$$Cm^1 = \sum_{\alpha=2}^{n_k} v_\alpha m^\alpha,$$

where C is constant. Since the monomials $\mathcal{M}_k(K)$ are linearly independent, it follows that $C = 0$, and $v_\alpha = 0$ for all α. We conclude that $\hat{\boldsymbol{M}}\boldsymbol{v} = 0$ implies that $\boldsymbol{v}$ is zero, so that $\hat{\boldsymbol{M}}$ is invertible.

Finally, we have that $\bar{m}^1 = |\partial K| > 0$ for $k = 1$ and $\bar{m}^1 = |K| > 0$ for $k \geq 2$. We consider the matrix $\boldsymbol{M}$, and expand the determinant through the first column:

$$\det(\boldsymbol{M}) = \bar{m}^1 \det(\hat{\boldsymbol{M}}) \neq 0.$$

We conclude that $\boldsymbol{M}$ is invertible, so that $\Pi^\nabla v_h$ can be calculated from the degrees of freedom for all $v_h \in V_k(K)$ by (4.4). □

As mentioned earlier, we will also need the matrix representation of Π^∇ in the canonical basis. Using the interpolation identity (2.27), we can write

$$\Pi^\nabla\phi^i = \sum_{j=1}^{N^K} \chi^j\left(\Pi^\nabla\phi^i\right)\phi^j, \quad i = 1, \ldots, N^K.$$

Expressing $\Pi^\nabla\phi^i$ in the monomial basis, and using the interpolation identity again, we have

$$\Pi^\nabla\phi^i = \sum_{\alpha=1}^{n_k} \pi_\alpha^i m^\alpha = \sum_{\alpha=1}^{n_k}\sum_{j=1}^{N^K} \pi_\alpha^i \chi^j(m^\alpha)\phi^j,$$

from which we conclude that $\chi^j\left(\Pi^\nabla\phi^i\right) = \sum_\alpha \pi_\alpha^i \chi^j(m^\alpha)$. We define the $N^K \times n_k$ matrix $\boldsymbol{D}$ such that $(\boldsymbol{D})_{i,\alpha} = \chi^i(m^\alpha)$, that is;

$$\boldsymbol{D} = \begin{bmatrix} \chi^1(m^1) & \cdots & \chi^1(m^{n_k}) \\ \vdots & \ddots & \vdots \\ \chi^{N^K}(m^1) & \cdots & \chi^{N^K}(m^{n_k}) \end{bmatrix},$$

and see that $\chi^j\left(\Pi^\nabla \phi^i\right) = (\boldsymbol{D}\boldsymbol{\Pi}_*^\nabla)_{j,i}$. Hence, the matrix representation of Π^∇ in the canonical basis $\{\phi^i\}_i$ is

$$\boldsymbol{\Pi}^\nabla = \boldsymbol{D}\boldsymbol{\Pi}_*^\nabla.$$

This time, writing $\boldsymbol{\phi}^{N^K} = \left(\phi^1, \dots, \phi^{N^K}\right)$, we have for a function $v_h \in V_k(K)$ that

$$\Pi^\nabla v_h = \sum_{i=1}^{N^K} \chi^i(v_h)\Pi^\nabla \phi^i = \boldsymbol{\phi}^{N^K}\boldsymbol{\Pi}^\nabla \hat{\boldsymbol{v}}_h.$$

Remark 4.4. Using again the interpolation identity (2.27), we note that

$$(\boldsymbol{B}\boldsymbol{D})_{1,\beta} = \sum\nolimits_i \bar{\phi}^i\chi^i(m^\beta) = \int_{\partial K}\sum\nolimits_i \phi^i\chi^i(m^\beta)\,\mathrm{d}s = \bar{m}^\beta \quad \text{for } k=1,$$
$$(\boldsymbol{B}\boldsymbol{D})_{1,\beta} = \sum\nolimits_i \bar{\phi}^i\chi^i(m^\beta) = \int_{K}\sum\nolimits_i \phi^i\chi^i(m^\beta)\,\mathrm{d}\boldsymbol{x} = \bar{m}^\beta \quad \text{for } k\geq 2,$$

so that $(\boldsymbol{B}\boldsymbol{D})_{1,\beta} = \boldsymbol{M}_{1,\beta}$. Moreover,

$$(\boldsymbol{B}\boldsymbol{D})_{\alpha,\beta} = \sum\nolimits_i a^K(m^\alpha,\phi^i)\chi^i(m^\beta) = a^K(m^\alpha,m^\beta) = \boldsymbol{M}_{\alpha,\beta}, \quad \alpha\geq 2.$$

That is, once we have $\boldsymbol{B}$ and $\boldsymbol{D}$, we already have $\boldsymbol{M} = \boldsymbol{B}\boldsymbol{D}$. ∘

Remark 4.5. Note that if we choose $\mathcal{E}^K$ to be the moments over the edges of K, constructing $\boldsymbol{D}$ involves calculating

$$|E|^{-1}\left(m^\alpha, \mu_E^\beta\right)_{0,E} \quad \forall \mu_E^\beta \in \mathcal{M}_k(E).$$

Hence, we need to construct a local coordinate system on each edge of K, and calculate the moments using these. This can easily be done by mapping each edge E to the unit interval.

We now have that

$$a^K\left(\Pi^\nabla\phi^i, \Pi^\nabla\phi^j\right) = \sum_{\alpha,\beta=1}^{n_k}\pi_\alpha^i a^K(m^\alpha,m^\beta)\pi_\beta^j = \left(\left(\boldsymbol{\Pi}_*^\nabla\right)^T\tilde{\boldsymbol{M}}\boldsymbol{\Pi}_*^\nabla\right)_{i,j},$$

where $\tilde{\boldsymbol{M}}$ coincides with $\boldsymbol{M}$ except for the first row, which is zero. Further, we recall from Theorem 3.2 that a valid stability term is on the form

$$s^K\left(u-\Pi^\nabla u, v-\Pi^\nabla v\right) = \left(\hat{\boldsymbol{u}} - \widehat{\boldsymbol{\Pi}^\nabla \boldsymbol{u}}\right)^T \boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T\left(\hat{\boldsymbol{v}} - \widehat{\boldsymbol{\Pi}^\nabla \boldsymbol{v}}\right),$$

where $\boldsymbol{\Sigma}$ is an $n_{\ker}\times n_{\ker}$ positive definite matrix, and $\boldsymbol{Q} = [\hat{\boldsymbol{\psi}}^1,\dots,\hat{\boldsymbol{\psi}}^{n_{\ker}}]$ is the matrix representation of $\ker\Pi^\nabla$. Moreover, these matrices are such that $\boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T$ scales as 1. Replacing u and v with ϕ^i and ϕ^j, we have that

$$s^K\left(\phi^i-\Pi^\nabla\phi^i, \phi^j-\Pi^\nabla\phi^j\right) = \left(\left(\boldsymbol{I}-\boldsymbol{\Pi}^\nabla\right)^T\boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T\left(\boldsymbol{I}-\boldsymbol{\Pi}^\nabla\right)\right)_{i,j}.$$

Recall from Remark 2.40 that $\ker \Pi^\nabla = \mathrm{Im}(\mathrm{Id} - \Pi^\nabla)$. Hence, once we have computed $\boldsymbol{\Pi}^\nabla$, we can simply construct $\boldsymbol{Q}$ by picking the linearly independent columns of $\boldsymbol{I} - \boldsymbol{\Pi}^\nabla$. Finally, we arrive at the following expression for the local stiffness matrix:

$$\boldsymbol{A}^K = \left(\boldsymbol{\Pi}_*^\nabla\right)^T \tilde{\boldsymbol{M}} \boldsymbol{\Pi}_*^\nabla + \left(\boldsymbol{I} - \boldsymbol{\Pi}^\nabla\right)^T \boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T \left(\boldsymbol{I} - \boldsymbol{\Pi}^\nabla\right).$$

Remark 4.6. We know from Lemma 3.1 that *any* $N^K \times N^K$ matrix $\boldsymbol{S}$ which is symmetric and positive definite on $\ker \Pi^\nabla$ can be written on the form $\boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T$. Hence, one can take the choice $\boldsymbol{S} = \boldsymbol{I}$, and avoid the process of determining $\boldsymbol{Q}$ from the linearly independent columns of $\boldsymbol{I} - \boldsymbol{\Pi}^\nabla$. This is the stability term presented in [33, 3, 34]. ◦

4.3 Computing the 3D local stiffness matrix

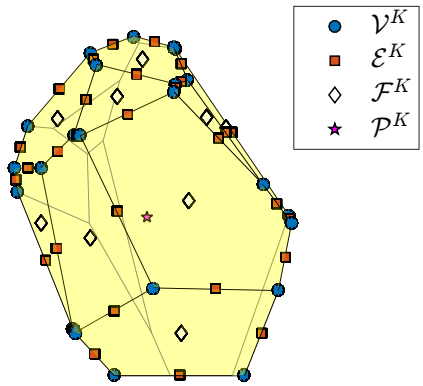


Figure 4.2: Example of a polyhedron K, with the degrees of freedom for $\mathbb{V}_2(K)$ indicated.

We start by recalling that the dimension of $\mathbb{P}_k(\mathbb{R}^3)$ is $n_k = (k+1)(k+2)(k+3)/6$.

Consider a polyhedron with n_V vertices, n_E edges and n_F faces. An example polyhedron is shown in Figure 4.2. Obviously, we can proceed as in two dimensions: Build the matrices $\boldsymbol{B}$ and $\boldsymbol{D}$, construct the matrix representations of our projection operator Π^∇, and calculate the local stiffness matrix.

The computation of the matrix $\boldsymbol{B}$ is now slightly different from the 2D case. First of all, we need the definition of $\bar{g}$ to be

$$\bar{g} = \begin{cases} \int_{\partial K} \Pi^{\nabla,\partial K} g \,\mathrm{d}s & \text{for } k = 1, \\ \int_K g \,\mathrm{d}\boldsymbol{x} & \text{for } k \geq 2. \end{cases}$$

Remark 4.7. It is now clear why we stated the definition of Π_1^∇ in $\mathbb{R}^3$ differently from how we stated it in $\mathbb{R}^2$: In order to calculate the matrix representations of Π_1^∇, we need to calculate $\bar{\phi}^i$ for all basis functions ϕ^i. If we used the 2D definition of Π_1^∇, this would involve calculating the integral of ϕ^i over the boundary of K. But we do not know the shape of the basis functions on ∂K . However, we are able to calculate their projections $\Pi_1^{\nabla,\partial K}\phi^i$. As a matter of fact, since $\phi^i|_{\partial K} \in V_1(\partial K)$, we have that

$$\int_F \Pi_1^{\nabla,F}\phi^i \,\mathrm{d}s = \int_F \phi^i \,\mathrm{d}s \quad \forall F \subset \partial K.$$

Hence, the definitions of Π_1^∇ in $\mathbb{R}^2$ and $\mathbb{R}^3$ are equal. ◦

Moreover, have that

$$\begin{aligned} a^K(m^\alpha, \phi^i) &= -(\Delta m^\alpha, \phi^i)_{0,K} + (\partial_n m^\alpha, \phi^i)_{0,\partial K} \\ &= -(\Delta m^\alpha, \phi^i)_{0,K} + \sum_{F\subset\partial K} (\partial_n m^\alpha, \phi^i)_{0,F}. \end{aligned} \tag{4.5}$$

Again, the first term can be computed directly from the degrees of freedom $\mathcal{P}^K$. Recalling the ordering of the monomials, it is clear that for any α such that $\alpha \leq n_k$, the monomial m^α belongs to $\mathbb{P}_k(K)$. Thus, for $\alpha \leq n_{k-1}$, we know that $\partial_n m^\alpha \in \mathbb{P}_{k-2}$, so that the last term can be computed directly from $\mathcal{F}^K$. However, as we have already pointed out in Section 2.5, we are not able to compute $a^K(m^\alpha, \phi^i)_{0,K}$ directly from the degrees of freedom whenever $\alpha > n_{k-1}$. The solution is to utilize the definition of the local VEM space $\mathbb{V}_k(K)$: Since $\phi^i|_{\partial K} \in V_k(\partial K)$, we know that

$$(\partial_n m^\alpha, \phi^i)_{0,F} = \left(\partial_n m^\alpha, \Pi^{\nabla,F}\phi^i\right)_{0,F}, \quad n_{k-1} < \alpha \leq n_k.$$

We have already seen how to compute the matrix representation of $\Pi^{\nabla,F}$ in two dimensions. Hence, we are able to compute the last term of (4.5).

Remark 4.8. It is important to notice that the projection operator $\Pi^{\nabla,F}$ is the projection onto the space of polynomials *on the face* F. Practically, this means that, for each face F of K, we need to construct a local coordinate system. Using this, we can calculate the projection $\Pi^{\nabla,F}\phi^i$ of each basis function onto the space of polynomials, and then evaluate the integrals $\left(\partial_n m^\alpha, \Pi^{\nabla,F}\phi^i\right)_{0,F}$ in this coordinate system. Moreover, similar to Remark 4.5, in order to construct $\boldsymbol{D}$, we have to calculate the moments over the edges and faces of K:

$$|E|^{-1}\left(m^\alpha, \mu_E^\beta\right)_{0,E} \quad \text{and} \quad |F|^{-1}\left(m^\alpha, \mu_F^\beta\right)_{0,F},$$

where $\mu_E^\beta \in \mathcal{M}_{k-2}(E)$ and $\mu_F^\beta \in \mathcal{M}_{k-2}(F)$. This too involves constructing local coordinate systems. A straight-forward construction procedure for the local coordinate system on a face F is to choose the first edge of F as the first coordinate axis, and construct the second coordinate axis by taking the cross product of this with the normal vector of F. ◦

We now know that we can compute $\Pi^\nabla v_h$ for any function $v_h \in \mathbb{V}_k(K)$ by

$$\Pi^\nabla v_h = \boldsymbol{m}^{n_k} \boldsymbol{\Pi}_*^\nabla \hat{\boldsymbol{v}}_h,$$

where $\boldsymbol{\Pi}_*^\nabla = (\boldsymbol{B}\boldsymbol{D})^{-1}\boldsymbol{B}$. Hence, we have the following theorem:

Theorem 4.9. *For any function $v_h \in \mathbb{V}_k(K)$, we have that $\Pi^\nabla v_h$ can be computed exactly.*

Since we have shown how to compute the matrices $\boldsymbol{B}$ and $\boldsymbol{D}$, and thereby $\boldsymbol{M} = \boldsymbol{B}\boldsymbol{D}$, the proof is equivalent to that of Theorem 4.3.

Once we have constructed the matrices $\boldsymbol{B}$ and $\boldsymbol{D}$, we can proceed as in two dimensions: We construct the matrices $\boldsymbol{\Pi}^\nabla$, $\boldsymbol{\Pi}_*^\nabla$ and $\boldsymbol{Q}$, choose a positive definite matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n_{\ker} \times n_{\ker}}$ such that $\boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T$ scales as h_K, and finally, compute the local stiffness matrix:

$$\boldsymbol{A}^K = \left(\boldsymbol{\Pi}_*^\nabla\right)^T \tilde{\boldsymbol{M}} \boldsymbol{\Pi}_*^\nabla + \left(\boldsymbol{I} - \boldsymbol{\Pi}^\nabla\right)^T \boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T \left(\boldsymbol{I} - \boldsymbol{\Pi}^\nabla\right).$$

As in $\mathbb{R}^2$, $\tilde{\boldsymbol{M}}$ coincides with the matrix $\boldsymbol{M}$ except for the first row, which is zero.

Remark 4.10. Similar to Remark 4.6, we can choose $\boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T = h_K \boldsymbol{I}$, and avoid constructing $\boldsymbol{Q}$. ◦

4.4 Computing the L^2 projection

We recall Definition 2.9 of the L^2 projection Π: For a function $v_h \in V_k(K)$, we have that

$$\left(\Pi v_h, m^\alpha\right)_{0,K} = \left(v_h, m^\alpha\right)_{0,K}, \quad \alpha = 1, \dots, n_k.$$

Expressing Πv_h in the monomial basis as $\Pi v_h = \sum_\beta \tilde{\pi}_\beta m^\beta$, we get

$$\sum_{\beta=1}^{n_k} \tilde{\pi}_\beta \left(m^\beta, m^\alpha\right)_{0,K} = \left(v_h, m^\alpha\right)_{0,K}, \quad \alpha = 1, \dots, n_k,$$

which we write in matrix form as

$$\begin{bmatrix} \left(m^1, m^1\right)_{0,K} & \cdots & \left(m^1, m^{n_k}\right)_{0,K} \\ \vdots & \ddots & \vdots \\ \left(m^{n_k}, m^1\right)_{0,K} & \cdots & \left(m^{n_k}, m^{n_k}\right)_{0,K} \end{bmatrix} \begin{bmatrix} \tilde{\pi}_1 \\ \vdots \\ \tilde{\pi}_{n_k} \end{bmatrix} = \begin{bmatrix} \left(v_h, m^1\right)_{0,K} \\ \vdots \\ \left(v_h, m^{n_k}\right)_{0,K} \end{bmatrix}. \tag{4.6}$$

Defining the matrix $\boldsymbol{H} \in \mathbb{R}^{n_k \times n_k}$ by

$$\boldsymbol{H}_{\alpha,\beta} = \left(m^\alpha, m^\beta\right)_{0,K}, \tag{4.7}$$

we can write this compactly as $\boldsymbol{H}\tilde{\boldsymbol{\pi}} = \boldsymbol{c}$. In the same manner as for Π^∇, we replace v_h in (4.6) by each of the basis functions ϕ^i in turn, and define the $n_k \times$

N^K matrix $\boldsymbol{C}$ such that

$$\boldsymbol{C}_{\alpha,i} = \left(m^\alpha, \phi^i\right)_{0,K}. \tag{4.8}$$

This gives the matrix representation of Π in the monomial basis:

$$\boldsymbol{\Pi}_* = \boldsymbol{H}^{-1}\boldsymbol{C}.$$

Reasoning similarly as in the first part of the proof of Theorem 4.3, we have that $\boldsymbol{H}$ is invertible. Moreover, we observe that the first n_{k-2} rows of $\boldsymbol{C}$ are given from the degrees of freedom $\mathcal{P}^K$, while the last rows are given from the definition of $V_k(K)$, since

$$(m, \phi^i)_{0,K} = \left(m, \Pi^\nabla \phi^i\right)_{0,K}, \quad \forall m \in \mathcal{M}^*_{k-1}(K) \cup \mathcal{M}^*_k(K).$$

Remark 4.11. Note that once we have computed $\boldsymbol{D}$, we also have the first n_{k-1} columns and rows of $\boldsymbol{H}$. Indeed; the last n_{k-2} rows of $\boldsymbol{D}$ are the first n_{k-2} moments of m^1 to m^{n_k} over K. Any product of two monomials $m^{\boldsymbol{\alpha}}m^{\boldsymbol{\beta}}$, with $|\boldsymbol{\alpha}|, |\boldsymbol{\beta}| \leq n_{k-1}$ can be written as a product of two monomials $m^{\tilde{\boldsymbol{\alpha}}}m^{\tilde{\boldsymbol{\beta}}}$, with $|\tilde{\boldsymbol{\alpha}}| \leq n_k$, $|\tilde{\boldsymbol{\beta}}| \leq n_{k-2}$. Thus, all inner products $(m^\alpha, m^\beta)_{0,K}$ with $\alpha, \beta \leq n_{k-1}$ are already computed (with a scaling factor $|K|^{-1}$) in $\boldsymbol{D}$. ◦

Remark 4.12. As mentioned, the first n_{k-2} rows of $\boldsymbol{C}$ are given from $\mathcal{P}^K$: For simplicity, consider a polygon K, with n edges. Similar to what we found for $\boldsymbol{B}$, we have that $\boldsymbol{C}_{\alpha,i} = |K|\chi^{(kn+\alpha)}(\phi^i)$, where n is the number of vertices of K. Further, for the last rows, we have that $\left(m^\alpha, \Pi^\nabla \phi^i\right)_{0,K} = \sum_\beta \pi^i_\beta (m^\alpha, m^\beta)_{0,K}$. This gives that $\boldsymbol{C}$ can be computed using the degrees of freedom:

$$\boldsymbol{C}_{\alpha,i} = \begin{cases} |K|, & 1 \leq \alpha \leq n_{k-2},\ i = kn + \alpha, \\ 0, & 1 \leq \alpha \leq n_{k-2},\ i \neq kn + \alpha, \\ (\boldsymbol{H}\boldsymbol{\Pi}^\nabla_*)_{\alpha,i}, & n_{k-2} + 1 \leq \alpha \leq n_k. \end{cases}.$$

The same is true in $\mathbb{R}^3$, but with different index limits for i. ◦

4.5 Computing the local load term

The next step is to compute the local load term $\boldsymbol{b}^K$. We recall from Definition 2.43 that $f_h = \Pi_{k-1} f$, so that

$$\boldsymbol{b}^K_i = \left(\Pi_{k-1} f, \phi^i\right)_{0,K} = \left(f, \Pi_{k-1}\phi^i\right)_{0,K}. \tag{4.9}$$

We denote $(\tilde{\boldsymbol{\Pi}}_*)_{\alpha,i}$ by $\tilde{\pi}^i_\alpha$, where, $\tilde{\boldsymbol{\Pi}}_*$ is the matrix representation of Π_{k-1} in the monomial basis $\mathcal{M}_{k-1}(K)$, and can be found by multiplying the inverse of the first n_{k-1} rows and columns of $\boldsymbol{H}$ by the first n_{k-1} rows and all N^K columns of $\boldsymbol{C}$, as defined in (4.7) and (4.8), respectively. From Section 4.4, it is clear that $\Pi_{k-1}\phi^i = \sum_\alpha \tilde{\pi}^i_\alpha m^\alpha$. The integrals (4.9) can then be computed using a suitable quadrature rule, as described in Section 4.1.2. Note that from Remark 4.11 and

Remark 4.12, once we have computed $\boldsymbol{A}^K$, we have done most of the work involved in computing $\tilde{\boldsymbol{\Pi}}_*$

In order to reduce the use of numerical integration, we could also approximate f by a function in $V_k(K)$ by $f \approx \sum_i \chi^i(f)\phi^i$. This gives

$$\Pi_{k-1} f = \Pi_{k-1}\left(\sum_{i=1}^{N^K} \chi^i(f)\phi^i\right) = \sum_{i=1}^{N^K} \chi^i(f)\Pi_{k-1}\phi^i = \sum_{\alpha=1}^{n_{k-1}} \sum_{i=1}^{N^K} \chi^i(f)\tilde{\pi}_\alpha^i m^\alpha,$$

which leads to

$$\boldsymbol{b}_i^K = \sum_{\alpha,\beta=1}^{n_k-1} \sum_{j=1}^{N^K} \tilde{\pi}_\alpha^j \chi^j(f)(m^\alpha, \phi^i)_{0,K}.$$

As usual, we denote the vector of degrees of freedom of f by $\hat{\boldsymbol{f}}$, and write this in matrix form as

$$\boldsymbol{b}^K = \tilde{\boldsymbol{C}}^T \tilde{\boldsymbol{\Pi}}_* \hat{\boldsymbol{f}}.$$

Here, $\tilde{\boldsymbol{C}}$ the first n_{k-1} rows and all N^K columns of $\boldsymbol{C}$, and as above, $\tilde{\boldsymbol{\Pi}}_*$ is the matrix representation of Π_{k-1} in the basis $\mathcal{M}_{k-1}(K)$. Note that using this approach involves calculating the degrees of freedom of f. Hence, if we choose $\mathcal{E}^K$ to be the values at $k-1$ points on each edge of k, this approach reduces the number of numerical integrations by $n_V + n_E(k-1)$, where n_V and n_E are the number of vertices and edges of K. Note that we must ensure that the convergence properties of Theorem 2.44 still holds with this approximation of the right-hand side. Using Proposition 4.3 in [33], which is a local version of the interpolation error Proposition 2.24, we see that this is indeed the case.

We also recall from Section 2.7 that for $k \geq 3$, we can choose $f_h = \Pi_{k-2} f$, and still have optimal convergence properties. This can be implemented in the exact same way as described above.

It should be mentioned that we can compute the local load term in any way we want, as long as we approximate it to the right order of magnitude. From a practical point of view, the approximation should be adapted to the problem we are solving .

4.6 Putting it all together

We can now construct the global stiffness matrix of our problem by a local to global mapping. In order to do so, we need a consistent ordering of the degrees of freedom, both locally and globally.

For a polygon K, with n vertices and edges, we order the N^K degrees of freedom of a function $v_h \in V_k(K)$ in the following way: The first n degrees of freedom are $\mathcal{V}^K$, the values of v_h at the vertices, in counter-clockwise direction, the next $n(k-1)$ are $\mathcal{E}^K$, the first $k-1$ moments of v_h over each edge of K, in counter-clockwise direction, and the last $k(k-1)/2$ are $\mathcal{P}^K$, the first $k(k-1)/2$ moments of v_h over K.

Globally, on a grid with N_V internal vertices, N_E internal edges and N_P polygons, we will have a predetermined ordering of the nodes, edges and cells. We then have the following ordering of the N degrees of freedom of a function $v_h \in V_h$: The first N_V degrees of freedom are $\mathcal{V}$, the values at the internal vertices, the next $(k-1)N_E$ are $\mathcal{E}$, the first $k-1$ moments of v_h over each of the internal edges, and the last $N_P k(k-1)/2$ are the first $k(k-1)/2$ moments of v_h over each of the N_P polygons.

In $\mathbb{R}^3$ we adopt the same ordering, but with the degrees of freedom $\mathcal{F}^K$, which are the $k(k-1)/2$ moments over each face, after $\mathcal{E}^K$ and before $\mathcal{P}^K$. Note that for a general polyhedron, there is no natural way to order the nodes, edges and faces, and we must simply decide one.

In order to construct the global stiffness matrix, we compute, for each element K, the local stiffness matrix $\boldsymbol{A}^K$ and load vector $\boldsymbol{b}^K$ as described above, and map it to the global stiffness matrix $\boldsymbol{A}$ and load vector $\boldsymbol{b}$ by a mapping

$$\boldsymbol{\delta} : K \to \text{(global degrees of freedom)},$$

that, for a cell K, gives the corresponding global vertex numbers, edge numbers and cell number of K. An illustration for a first order VEM in $\mathbb{R}^2$ is shown in Figure 4.3.

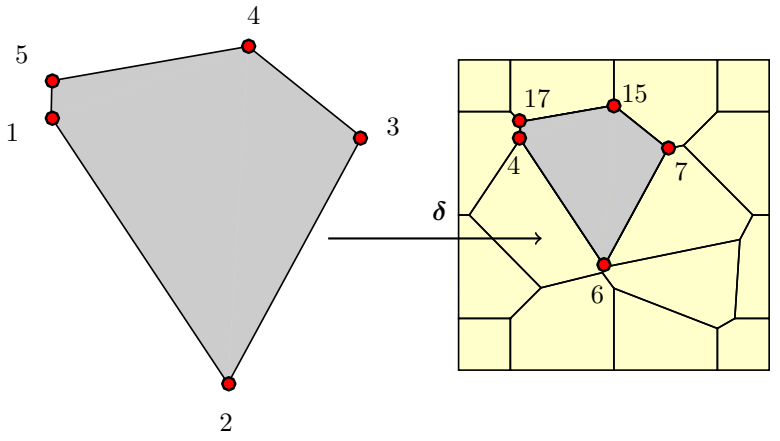


Figure 4.3: The mapping $\boldsymbol{\delta}$, taking local degrees of freedom to global degrees of freedom.

4.7 Boundary Conditions

Up to this point, we have only considered homogeneous Dirichlet boundary conditions. As we will see, we can easily modify our method to incorporate

both inhomogeneous Dirichlet and Neumann boundary conditions. To this end, consider the following problem

$$\begin{aligned} -\Delta u &= f, && \boldsymbol{x} \in \Omega, \\ u &= g_D, && \boldsymbol{x} \in \Gamma_D, \\ \partial_n u &= g_N, && \boldsymbol{x} \in \Gamma_N, \end{aligned}$$

where $\Omega \subset \mathbb{R}^2$ or $\mathbb{R}^3$, and Γ_D and Γ_N are the Dirichlet and Neumann boundaries, with

$$\Gamma_D \cup \Gamma_N = \partial\Omega, \quad \Gamma_D \cap \Gamma_N = \emptyset.$$

Moreover, $f \in L^2(\Omega)$. We define the function spaces

$$\begin{aligned} U &= \{w \in H^1(\Omega) : w|_{\Gamma_D} = g_D,\, \partial_n w|_{\Gamma_N} = g_N\}, \\ V &= \{w \in H^1(\Omega) : w|_{\Gamma_D} = 0\}. \end{aligned}$$

We multiply by a function $v \in V$ and integrate over Ω:

$$\begin{aligned} -(\Delta u, v)_{0,\Omega} &= (\nabla u, \nabla v)_{0,\Omega} - (\partial_n u, v)_{0,\partial\Omega} \\ &= (\nabla u, \nabla v)_{0,\Omega} - (g_N, v)_{0,\Gamma_N}, \end{aligned}$$

which leads to the weak formulation

$$\text{Find } u \in U \text{ such that} \quad a(u,v) = (f,v)_{0,\Omega} + (g_N, v)_{0,\Gamma_N} \quad \forall v \in V. \tag{4.10}$$

The question is now how we can obtain a VEM implementation of this weak formulation. We observe that the Neumann conditions are given by

$$\boldsymbol{b}_i = (f, \phi^i)_{0,\Omega} + (g_N, \phi^i)_{0,\Gamma_N}.$$

Further, the Dirichlet conditions can easily be imposed by setting $\boldsymbol{A}_{i,j} = \delta_{ij}$ and $\boldsymbol{b}_i = \chi^i(g_D)$ whenever i is such that $\operatorname{supp}(\phi^i) \cap \Gamma_D \neq \emptyset$. Notice that the Neumann conditions are incorporated into the model by adding a term to the right-hand side, while the Dirichlet conditions are incorporated as absolute: We know the exact solution values at the Dirichlet boundary Γ_D, and construct our matrix system such that the approximated solution is exact at Γ_D. We say that Neumann conditions are weakly imposed, while Dirichlet conditions are strongly imposed.

Denoting the set of degrees of freedom that resides on the Dirichlet boundary by the multiindex $\boldsymbol{\iota}_D$, we have that the stiffness matrix and load vector for (4.10) becomes

$$\boldsymbol{A}_{i,j} = \begin{cases} a_h(\phi^i, \phi^j), & i \notin \boldsymbol{\iota}_D, \\ \delta_{ij}, & i \in \boldsymbol{\iota}_D, \end{cases} \tag{4.11}$$

$$\boldsymbol{b}_i = \begin{cases} \langle f_h, \phi^i \rangle + (g_N, \phi^i)_{0,\Gamma_N}, & i \notin \boldsymbol{\iota}_D, \\ \chi^i(g_D), & i \in \boldsymbol{\iota}_D. \end{cases} \tag{4.12}$$

A straightforward approach to incorporate inhomogeneous boundary conditions is then to calculate $\boldsymbol{A}$ and $\boldsymbol{b}$ *as there were no boundary conditions*, and then change the rows of $\boldsymbol{A}$ and the elements of $\boldsymbol{b}$ according to (4.11) and (4.12).

Remark 4.13. Note that in $\mathbb{R}^2$, it is easy to calculate $(g_N, \phi^i)_{0,\Gamma_N}$, since Γ_N is a set of edges, and $\phi^i|_E \in \mathbb{P}_k(E)$ for all edge E. Hence, similar to Remark 4.1, we can use a Lobatto quadrature and evaluate the integral. In order to preserve the convergence rate of $k+1$, we need the quadrature rule to converge with rate $k+1$. In $\mathbb{R}^3$, however, we cannot apply this approach, since Γ_N is now a collection of faces, and we do not know the shape of our basis functions ϕ^i on ∂K. The solution is then to approximate the integral by

$$(g_N, \phi^i)_{0,\Gamma_N} \approx \sum_{F \subset \Gamma_N} \left(g_N, \Pi^{\nabla,F} \phi^i\right)_{0,F},$$

and then use the technique of triangulating each face F and apply a quadrature rule, as described in Section 4.1.2. ◦

Chapter 5

Numerical Examples

Understanding is, after all, what science is all about – and science is a great deal more than mindless computation.

Roger Penrose

Having thoroughly gone through the theoretical aspects of the virtual element method for Poisson problems, and examined the implementation details, we are now ready to test the method.

In this chapter, we will discuss some of the implementation details of the MATLAB implementation of VEM done as part of this thesis. Then, we present two simple model problems with known solutions, one in $\mathbb{R}^2$, and one in $\mathbb{R}^3$. For these two problems, we will confirm the convergence rates predicted by the theory. We will also look at how different choices of the stability term affects the solution for a given problem. Next, we briefly review the most common methods used in reservoir simulations; the two-point and mulitpoint flux-approximation methods, abbreviated TPFA and MPFA, respectively, and the mimietic finite difference method (MFD). Finally, we compare these three methods with VEM for some simple test cases. As we investigate the different examples, we will also present some implementation details for VEM not mentioned in Chapter 4.

5.1 Implementation details

We have implemented the virtual element method using the open source MATLAB reservoir simulation toolbox (MRST), developed at SINTEF ICT, see [19, 20] for a comprehensive introduction. The implementation is the author's own work, and can be downloaded from the author's git repository [18].

The implementation supports first and second order, and is done as described in Chapter 4. The degrees of freedom $\mathcal{E}^K$ are chosen to be the values at the Lobatto quadrature points on each edge, as mentioned Remark 2.15 and Remark 2.31. The reason is better interpretability of the calculated solution. Moreover, the right-hand side is approximated using $\Pi_{k-1} f \approx \sum_i \chi^i(f) \Pi_{k-1} \phi^i$, as suggested in Section 4.5.

Choosing the stability term is also possible. By default, the matrix representation of the stability term s^K for each cell K is taken to be $\boldsymbol{QQ}^T$, where $\boldsymbol{Q}$ is an orthogonal basis of the columns of $\boldsymbol{I} - \boldsymbol{\Pi}^\nabla$, as described in Section 4.2. One can also choose the stability term to be $\boldsymbol{Q\Sigma Q}^T$, where $\boldsymbol{\Sigma}$ is any $n_{\text{ker}} \times n_{\text{ker}}$ positive definite diagonal matrix. Note that in theory, $\boldsymbol{\Sigma}$ could be any full, positive definite matrix. However, as we saw in Section 3.2 and Section 3.3, it was sufficient with a diagonal matrix to make VEM equivalent to the finite element method and finite difference methods. For regular Cartesian grids, one can also choose $\boldsymbol{Q}$ to be as in Example 3.11 in $\mathbb{R}^2$ or Example 3.12 in $\mathbb{R}^3$, respectively.

For evaluation of integrals of non-polynomial functions over polygons and polyhedra, we have applied the technique of triangulating the polygon or polyhedron, as described in Section 4.1.2. The quadrature rules used for triangles are the ones described in [30], while the ones used for tetrahedra are described in [37].

We believe this is the first implementation of higher-order VEM for Poisson problems in three dimensions.

5.2 A 2D model problem

We will consider the following Poisson problem:

$$\begin{aligned} -\Delta u &= 4(\pi^2 - 1)e^{-2x}\cos(2\pi y), && \boldsymbol{x} \in \Omega, \\ u &= e^{-2x}\cos(2\pi y), && \boldsymbol{x} \in \Gamma_D, \\ \partial_n u &= 2\cos(2\pi y), && \boldsymbol{x} \in \Gamma_N, \end{aligned} \tag{5.1}$$

where $\Omega = [0,1] \times [0,1]$, and Γ_N is the Neumann boundary:

$$\Gamma_N = \{(x,y) \in \{0\} \times [0,1]\},$$

which we refer to as the western boundary. Further, $\Gamma_D = \partial\Omega \setminus \Gamma_N$ is the Dirichlet boundary, which is then the southern, eastern and northern boundaries of Ω. The exact solution to this problem is

$$u = e^{-2x}\cos(2\pi y).$$

5.2.1 Estimating the L^2-norm

We denote the approximated solution to (5.1) obtained from a k-th order VEM on a grid $\mathcal{T}_h$ by u_h. We know from Theorem 2.44 that we expect the L^2-norm of the error $u - u_h$ to scale with h^{k+1}. The L^2-norm can be split as follows:

$$\|u - u_h\|_{0,\Omega}^2 = \sum_{K \in \mathcal{T}_h} \|u - u_h\|_{0,K}^2.$$

Since we do not know the shape of u_h on the interior of each polygon K, we cannot evaluate this exactly. However, we do know its projection $\Pi^\nabla u_h$, and

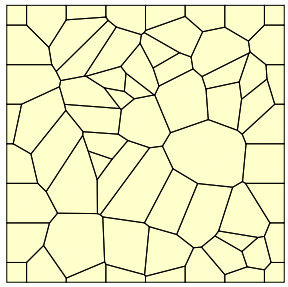
we approximate the L^2-error by

$$\|u - u_h\|_{0,\Omega}^2 \approx \sum_{K \in \mathcal{T}_h} \left\|u - \Pi^{\nabla} u_h\right\|_{0,K}^2 . \tag{5.2}$$

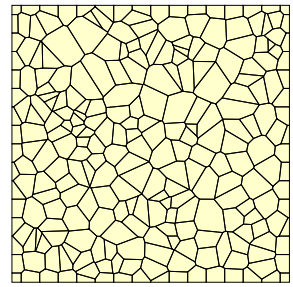
This can then be calculated using a suitable quadrature rule.

5.2.2 Numerical solutions

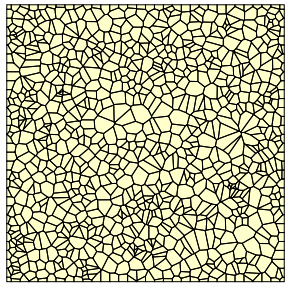
We now analyse the virtual element method by solving (5.1) using a first and second order VEM on four grids. The stability term for each polygon is chosen to be the default form $\boldsymbol{QQ}^T$. The grids are generated by setting equally spaced points on the boundary, along with random points inside the domain. A triangulation is then made of these points. Finally, the dual grid of this triangle grid is calculated. See [4] for details. The grids are shown in Figure 5.1, with the number of polygons n_K and mean diameter h indicated. The approximated solutions u_h to (5.1) are shown in Figure 5.2a.



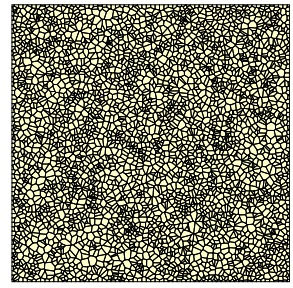
(a) Grid 1: $n_K = 60$, $h \approx 19 \cdot 10^{-2}$.



(b) Grid 2: $n_K = 250$, $h \approx 9.6 \cdot 10^{-2}$.

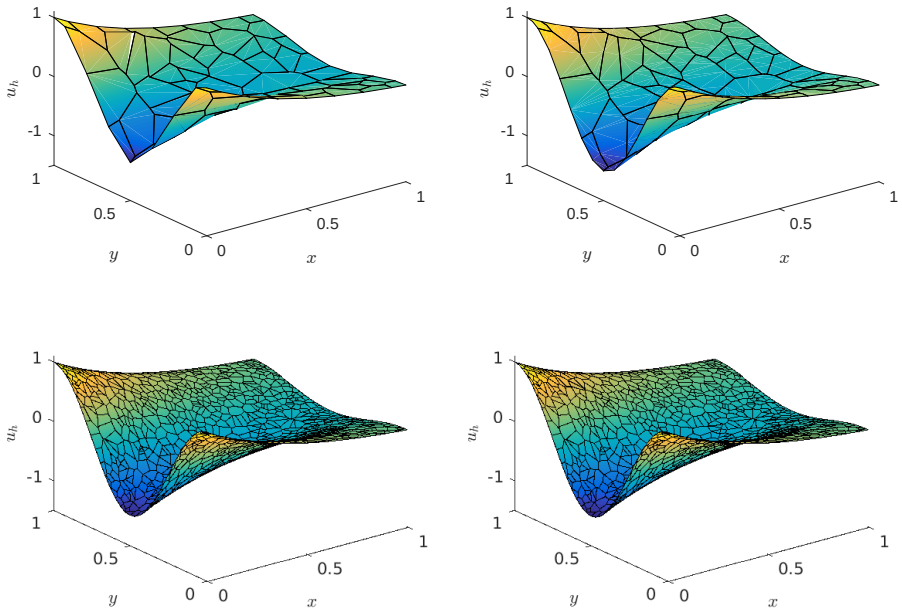


(c) Grid 3: $n_K = 1020$, $h \approx 4.8 \cdot 10^{-2}$.

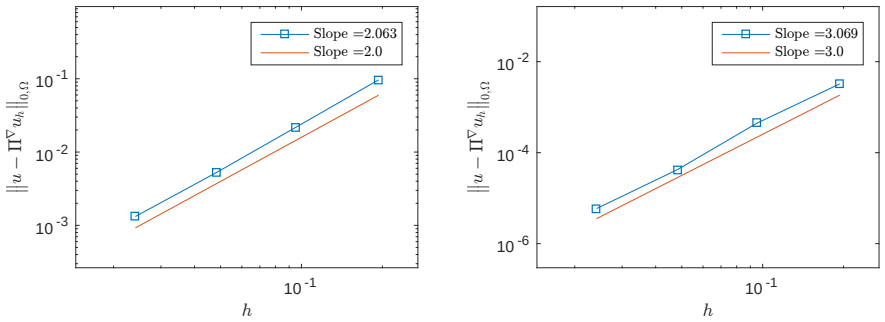


(d) Grid 4: $n_K = 4090$, $h \approx 2.4 \cdot 10^{-2}$.

Figure 5.1: The four grids used to solve (5.1).



(a) Solutions to (5.1) using a first order (left column) and second order (right column) VEM on grids 1 (upper row) and 3 (lower row) in Figure 5.1.



(b) Log-log plot of the approximated L^2-norm of the error using a first order (right) and second order (left) VEM.

Figure 5.2: Solutions and convergence rates for (5.1).

In terms of visualization, we now see the advantage of choosing the degrees of freedom $\mathcal{E}^K$ to be the values at $k-1$ points on each edge: We can use these values when we plot the solution, and we see that for the first grid, the second order solution is much smoother than the first order solution.

A log-log plot of the approximated L^2-norm of the error using (5.2) with respect to the mean grid diameter h is shown in Figure 5.2b. We observe that we have convergence rates close to $k+1$, which is in accordance with Theorem 2.44.

5.3 A 3D model problem

We now consider the following Poisson problem:

$$\begin{aligned} -\Delta u &= -\left(x^2+2\right) ye^z, && \boldsymbol{x} \in \Omega, \\ u &= x^2 ye^z, && \boldsymbol{x} \in \Gamma_D, \\ \partial_n u &= x^2 ye, && \boldsymbol{x} \in \Gamma_N, \end{aligned} \tag{5.3}$$

where $\Omega = [0,1] \times [0,1] \times [0,1]$. The boundary is split into the Neumann boundary Γ_N, defined as

$$\Gamma_N \coloneqq \{(x,y,z) \in [0,1] \times [0,1] \times \{1\}\},$$

that is, the top boundary. The Dirichlet boundary $\Gamma_D \coloneqq \partial\Omega \setminus \Gamma_N$, is then the bottom, northern, southern, eastern and western boundaries. The exact solution to this problem is

$$u = x^2 ye^z. \tag{5.4}$$

Again, we can approximate L^2-error using (5.2).

5.3.1 Computing averages

We would like to be able to plot the solutions u_h to 3D problems on the grids we use to solve them. For $k \geq 2$, this is straight-forward, since the degrees of freedom $\mathcal{F}$ gives the moments

$$|F|^{-1} \int_F u_h \, \mathrm{d}s$$

for all faces F of the grid. These values are the averages of the approximated solution over each of the faces, and we can simply plot these for all visible faces of the grid to see what the solution looks like. However, these values are not calculated for $k = 1$. The solution is then to use the definition of the VEM space $\mathbb{V}_1(K)$: For a function $v_h \in \mathbb{V}_1(K)$, we know that the restriction $v_h|_F$ to a face F of K is in $V_1(F)$, and it follows that

$$\int_F u_h \, \mathrm{d}s = \int_F \Pi^{\nabla,F} u_h \, \mathrm{d}s.$$

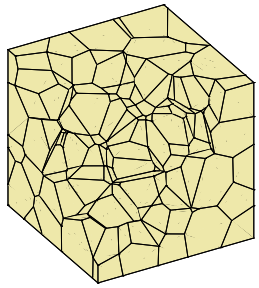
Note that the same is true for the averages over the cells, since

$$\int_K u_h \, \mathrm{d}\boldsymbol{x} = \int_K \Pi^{\nabla,K} u_h \, \mathrm{d}\boldsymbol{x}.$$

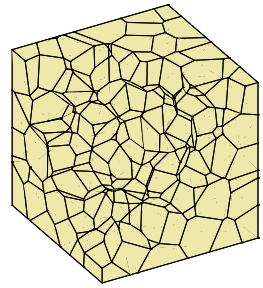
Hence, for a first order solution, we are able to calculate the degrees of freedom $\mathcal{F}$ and $\mathcal{P}$ as for a second order solution using the projection operator Π^{∇}. Obviously, the same is true for $\mathcal{P}$ in $\mathbb{R}^2$.

5.3.2 Numerical solutions

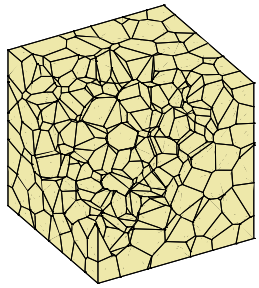
As in the 2D case, we now analyse the virtual element method by solving (5.3) using a first and second order VEM on four grids. See [4] on how these grids are generated. The grids are shown in Figure 5.3, and the approximated solutions u_h to (5.1) using the first and last grids are shown in Figure 5.4a.



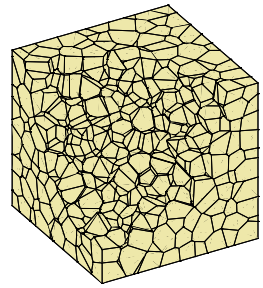
(a) Grid 1: $n_K = 125$, $h \approx 3.6 \cdot 10^{-1}$.



(b) Grid 2: $n_K = 216$, $h \approx 3.0 \cdot 10^{-1}$.

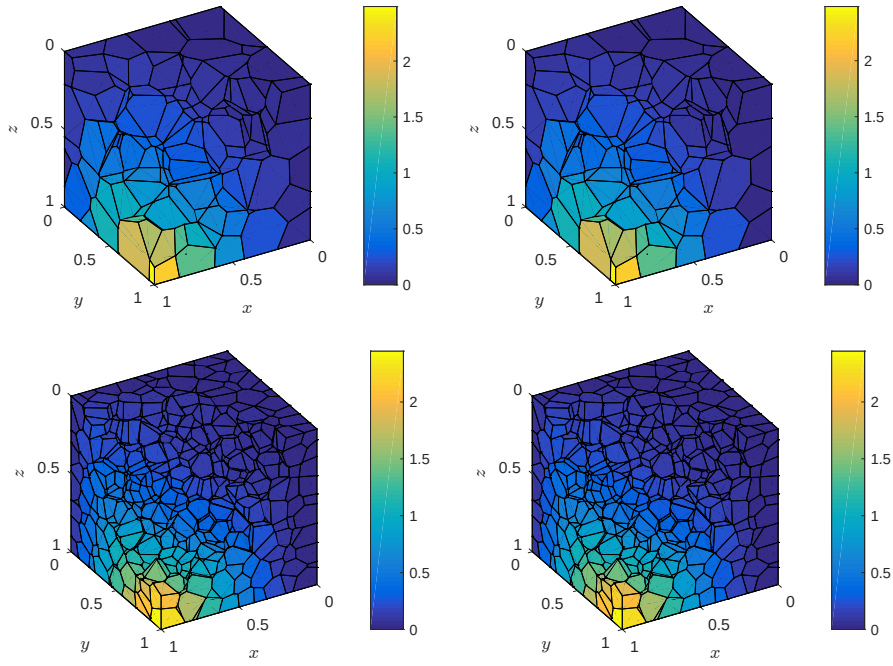


(c) Grid 3: $n_K = 504$, $h \approx 2.2 \cdot 10^{-1}$.

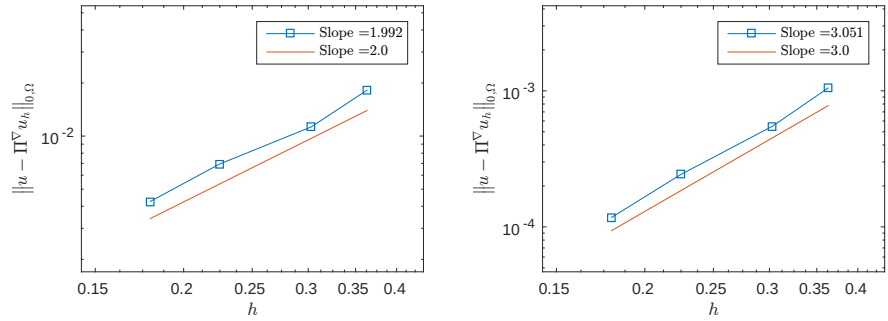


(d) Grid 4: $n_K = 1000$, $h \approx 1.8 \cdot 10^{-1}$.

Figure 5.3: The four grids used to solve (5.3).



(a) Solutions to (5.3) using a first order (left column) and second order (right column) VEM on grids 1 (upper row) and 4 (lower row) in Figure 5.3.



(b) Log-log plot of the approximated L^2-norm of the error using a first order (left) and second order (right) VEM.

Figure 5.4: Solutions and convergence rates for (5.3).

A log-log plot of the approximated L^2-norm of the error (5.2) with respect to the mean grid diameter is shown in Figure 5.4b. The As in 2D, we observe that we have convergence rates close to $k+1$, which is in accordance with Theorem 2.44.

5.4 Effect of the stability term

As mentioned, the implementation supports choosing the stability term of the method. We will now see how different choices of the stability term impacts the approximated solution. Consider the domain $\Omega = [-1,1] \times [-1,1]$, with a unit point source placed at the origin. This can be described by the differential equation

$$-\Delta u = \delta(\boldsymbol{x}), \tag{5.5}$$

where δ is the Dirac delta function. The solution to (5.5) is know as the fundamental solution of the Laplace equation [15, Chapter 2.2.1]. In $\mathbb{R}^2$, it reads

$$u = -\frac{1}{2\pi} \log\left(|\boldsymbol{x}|\right). \tag{5.6}$$

Note that u is symmetric around the origin.

5.4.1 Implementing sources and sinks

In the VEM framework, the point source problem (5.5) gives the local source term $(\delta, \cdot)_{0,K}$. In this case, we cannot apply the same approach as for a function $f \in L^2(K)$. Instead, we construct our grid $\mathcal{T}_h$ such that the cell K_S has centroid $\boldsymbol{x}_{K_S} = (0,0)^T$, and approximate the Dirac delta function by a function $|K_S|^{-1}\iota_{K_S}(\boldsymbol{x})$, where

$$\iota_{K_S}(\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{x} \in K_S, \\ 0, & \text{otherwise.} \end{cases}$$

The right-hand side then reads

$$(\delta, \phi^i)_{0,K} \approx |K_S|^{-1}(\iota_{K_S}, \phi^i)_{0,K} = \begin{cases} |K_S|^{-1} \int_{K_S} \phi^i \,\mathrm{d}\boldsymbol{x}, & K = K_S, \\ 0, & \text{otherwise.} \end{cases}$$

For $k \geq 2$, this integral is given from the degrees of freedom $\mathcal{P}^K$, while for $k = 1$, it can be computed using the method described in 5.3.1.

More generally, we can easily define a source or sink with flux rate Q, centered at a point $\boldsymbol{c} \in \Omega$: We simply constructing the grid such that $\boldsymbol{x}_{K_S} = \boldsymbol{c}$, and use the approximation

$$Q\delta(\boldsymbol{x} - \boldsymbol{c}) \approx Q|K_S|^{-1}\iota_{K_S}(\boldsymbol{x}).$$

Sinks are naturally implemented as negative sources, with flux rate $-Q$. For the implementation used here, a source or sink is thus defined by defining the

source cell K_S, along with its flux rate. The same is, of course, true in three dimensions.

5.4.2 Solutions using different stability terms

We construct a Cartesian grid $\mathcal{T}_{h_x,h_y}$ over Ω, with aspect ratio $h_y/h_x = 11/101$. In order to analyse the effect of different stability terms, we recall from Example 3.11 that the exact stability term for rectangles reads $s^K = \boldsymbol{Q}\boldsymbol{P}^{-1}\boldsymbol{\Lambda}\boldsymbol{P}^{-1}\boldsymbol{Q}^T$, where

$$\boldsymbol{Q} = \sqrt{\frac{9}{4h_xh_y}}(1,-1,1,-1)^T, \quad \boldsymbol{P} = \boldsymbol{Q}^T\boldsymbol{Q}, \quad \boldsymbol{\Lambda} = 3\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}\right).$$

We use this as a staring point, and remember the fact that the stability term can be scaled by any positive constant. Thus, we find the constant w that minimizes the error $e_h \coloneqq u_h - u$ in a suitable norm, and use

$$w\boldsymbol{Q}\boldsymbol{P}^{-1}\boldsymbol{\Lambda}\boldsymbol{P}^{-1}\boldsymbol{Q}^T \tag{5.7}$$

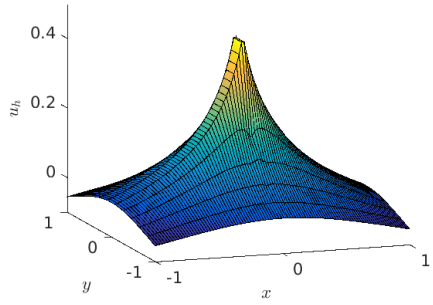
as our stability term.

We seek to minimize the error in two norms: the approximated L^2-norm (5.2), and the Euclidean norm $|\hat{e}_h|$. For comparison, we also include two other solutions: The solution using the default stability term $\boldsymbol{Q}\boldsymbol{Q}^T$, with $\boldsymbol{Q}$ an orthonormal basis for $\boldsymbol{I} - \boldsymbol{\Pi}^\nabla$ for each cell, and the solution with stability term (5.7) and $w = 1$. From Proposition 3.20, we know that the latter is the well-known finite element method. The results are shown in Figure 5.5a-5.5d, with the scaled error norms indicated. Moreover, the scaled error norms with respect to the weight w is shown in Figure 5.5e. In order to avoid contribution to the error close to the singularity, we have excluded the cells with centroid inside a circle of radius $h_y/2$ centered at the origin when measuring the error.

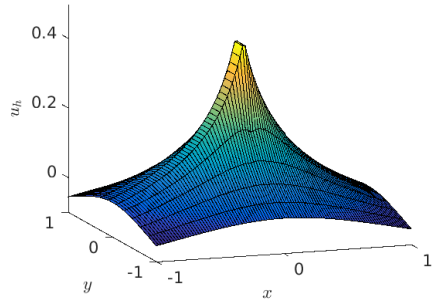
For the solution using the default stability term, and the FEM solution, we observe a strange dip along the line $x = 0$, and these solutions are obviously not correct in the local sense. For the solutions optimal in the L^2 norm and euclidean norm, we see that this dip is not evident. Moreover, we observe that the weight w minimizing the L^2 error and the error in the Euclidean norm are close to each other, and the solutions are visibly indistinguishable. Further, the L^2-norms of these solutions are very close to each other. Generally, we see that the error in the two norms seems to follow the same trend with respect to the weight w, but with the Euclidean norm being more sensitive to the choice of w.

5.5 Comparing with other methods

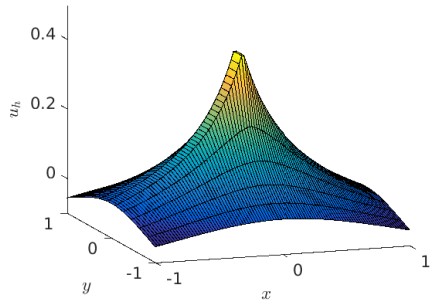
Having seen that the implementation of the virtual element method behaves as expected, we now move on to compare it with other methods. First, we give a brief review of some of the most common methods used in reservoir simulations.



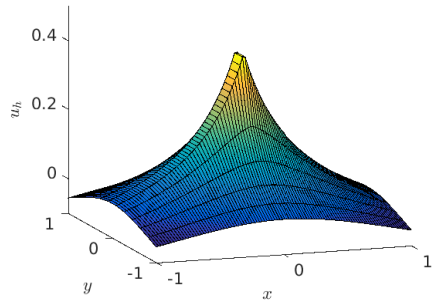
(a) Default stability term.
$\frac{\|e_h\|_{0,\Omega}}{\|u\|_{0,\Omega}} \approx 5.27 \cdot 10^{-2}$, $\frac{|\hat{e}_h|}{|\hat{u}|} \approx 4.90 \cdot 10^{-2}$.



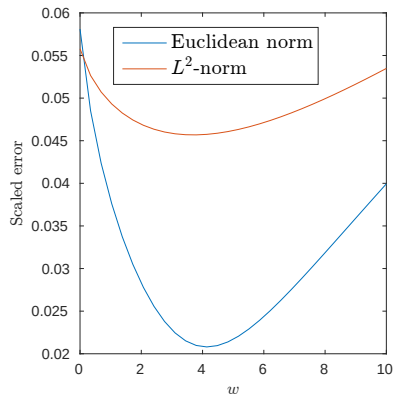
(b) Stability term (5.7), $w = 1$.
$\frac{\|e_h\|_{0,\Omega}}{\|u\|_{0,\Omega}} \approx 4.94 \cdot 10^{-2}$, $\frac{|\hat{e}_h|}{|\hat{u}|} \approx 3.80 \cdot 10^{-2}$.



(c) Stability term (5.7), $w = 3.70$.
$\frac{\|e_h\|_{0,\Omega}}{\|u\|_{0,\Omega}} \approx 4.57 \cdot 10^{-2}$, $\frac{|\hat{e}_h|}{|\hat{u}|} \approx 2.11 \cdot 10^{-2}$.



(d) Stability term (5.7), $w = 4.14$.
$\frac{\|e_h\|_{0,\Omega}}{\|u\|_{0,\Omega}} \approx 4.57 \cdot 10^{-2}$, $\frac{|\hat{e}_h|}{|\hat{u}|} \approx 2.08 \cdot 10^{-2}$.



(e) Scaled error norms with respect to the weighted stability term (5.7).

Figure 5.5: Approximation of the fundamental solution of the Laplace equation using different stability terms.

Similar to what we discussed in Section 2.2, we will consider the following subsurface flow problem:

$$\nabla \cdot \boldsymbol{v} = q, \quad \boldsymbol{v} = -\boldsymbol{K}\nabla p, \quad \boldsymbol{x} \in \Omega \subset \mathbb{R}^d, \tag{5.8}$$

where $\boldsymbol{v}$ is the fluid velocity, p is the fluid pressure, $\boldsymbol{K}$ is the permeability, q is a source or sink, and $d = 2$ or 3. We assume that suitable boundary conditions are given, but these are not essential in describing the framework of the methods. Note that (5.8) is equivalent to the elliptic equation $-\nabla \cdot \boldsymbol{K}\nabla p = q$. Moreover, we recognize the second equation as Darcy's law.

In the discussion of the methods, we will assume that we have a grid $\mathcal{T}_h$ covering Ω, with N_P polygons or polyhedrons.

5.5.1 Two-point flux-approximation

The two-point flux approximation (TPFA) is a finite volume method, and the current industry standard in reservoir simulations [20]. There are several sources covering the TPFA method, see for example [19, Chapter 6.4.1] or [20]. A brief description will be given here. We consider a cell K_i, and use the divergence theorem to obtain the following integral form of the first equation of (5.8):

$$\int_{\partial K_i} \boldsymbol{v} \cdot \boldsymbol{n} \, \mathrm{d}s = \int_{K_i} q \, \mathrm{d}\boldsymbol{x}. \tag{5.9}$$

Next, we consider two cells K_i and K_j, with common boundary $F_{i,j}$. Note that this gives a natural way to define *half-faces*: The half-face $F_{i,j}$ is the common face of cell K_i and cell K_j, with orientation given by the normal vector $\boldsymbol{n}_{i,j}$ pointing from cell K_i to cell K_j. In the same manner, $F_{j,i}$ shares vertices and area with $F_{i,j}$ but is oriented with $\boldsymbol{n}_{j,i} \coloneqq -\boldsymbol{n}_{i,j}$. The flux across each half-face is

$$v_{i,j} \coloneqq \int_{F_{i,j}} \boldsymbol{v} \cdot \boldsymbol{n} \, \mathrm{d}s.$$

We denote the centroid of $F_{i,j}$ by $\boldsymbol{x}_{i,j}$, and use the midpoint rule and Darcy's law to obtain

$$v_{i,j} \approx |F_{i,j}| \boldsymbol{v}(\boldsymbol{x}_{i,j}) \cdot \boldsymbol{n}_{i,j} = -|F_{i,j}| \left(\boldsymbol{K}\nabla p\right)(\boldsymbol{x}_{i,j}) \cdot \boldsymbol{n}_{i,j}. \tag{5.10}$$

As the name indicates, we now want to approximate the flux over each half-face by using the pressure at two points; one value at the cell centroid, and one at the face centroid. We denote them by π_i and $\pi_{i,j}$, respectively. Assuming that the pressure is linear in each cell, we the know that π_i equals the average pressure p_i inside K_i. Then (5.10) can be approximated by

$$v_{i,j} \approx |F_{i,j}| \boldsymbol{K}(p_i - \pi_{i,j}) \frac{\boldsymbol{c}_{i,j}}{|\boldsymbol{c}_{i,j}|^2} \cdot \boldsymbol{n}_{i,j},$$

where $\boldsymbol{c}_{i,j}$ is the vector from the cell centroid to the face centroid. Defining the half-face transmissibilities

$$T_{i,j} = |F_{i,j}|\boldsymbol{K}_i \frac{\boldsymbol{c}_{i,j}}{|\boldsymbol{c}_{i,j}|^2} \cdot \boldsymbol{n}_{i,j},$$

where $\boldsymbol{K}_i$ is the permeability associated with cell K_i, we can write this as

$$v_{i,j} \approx T_{i,j}(p_i - \pi_{i,j}). \tag{5.11}$$

An illustration is given in Figure 5.6.

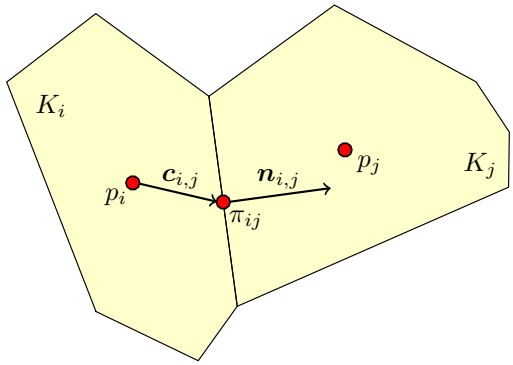


Figure 5.6: Illustration of the TPFA method.

We now impose two physical conditions: *(i)* continuity of fluxes across faces, $v_{ij} := v_{i,j} = -v_{j,i}$, and *(ii)* continuity of face pressures, $\pi_{ij} := \pi_{i,j} = \pi_{j,i}$. This gives the two equations

$$T_{i,j}^{-1} v_{ij} = p_i - \pi_{ij}, \quad -T_{j,i}^{-1} v_{ij} = p_j - \pi_{ij}.$$

Defining $T_{ij} := \left(T_{i,j}^{-1} + T_{j,i}^{-1}\right)^{-1}$, the transmissibility associated with the connection between K_i and K_j, and inserting the approximated face fluxes into (5.9), we obtain the system

$$\sum_{j=1}^{N_P} T_{ij}(p_i - p_j) = q_i \quad \forall K_i \in \mathcal{T}_h,$$

where $q_i = \int_{K_i} q \, \mathrm{d}\boldsymbol{x}$ is the flux density out of cell K_i. In a reservoir setting, this is the rate at which fluid is produced in cell K_i.

It can be shown that the TPFA method is only convergent for so-called $\boldsymbol{K}$-orthogonal grids. A sufficient condition for $\boldsymbol{K}$-orthogonality is that $\boldsymbol{K}\boldsymbol{n}_{i,j}$ is parallel to $\boldsymbol{c}_{i,j}$ for all cells in the grid. Due to this fact, the TPFA method can suffer from significant gird-orientation effects [19, 20].

A method that is consistent for $\boldsymbol{K}$-orthogonal grids is the multipoint flux approximation method (MPFA) is (see for example [1, 2, 20]). This method can be derived in much the same way as the TPFA method, but uses the pressure at several neighboring points in order to approximate the pressure gradient in (5.10).

5.5.2 Hybrid formulation

Let K_i be a cell with n_F faces. We assume that our method can be written on the form

$$\boldsymbol{v}_i = \boldsymbol{T}_i(\boldsymbol{e}_i p_i - \boldsymbol{\pi}_i),$$

where $\boldsymbol{v}_i = (v_{i1}, \dots, v_{in_F})^T$ is the vector of fluxes across each face of K_i, $\boldsymbol{e}_i = (1, \dots, 1)^T \in \mathbb{R}^{n_F}$, $\boldsymbol{\pi}_i$ is the vector of face pressures, and $\boldsymbol{T}_i$ is a matrix of half-face transmissibilities. Note that this is equivalent to (5.11) in vector form, and in this case, $\boldsymbol{T}_i$ is the diagonal matrix with diagonal $(T_{i,1}, \dots, T_{i,n_F})^T$. However, in the general case, $\boldsymbol{T}_i$ is a full matrix. We write this as

$$\boldsymbol{M}_i \boldsymbol{v}_i = \boldsymbol{e}_i p_i - \boldsymbol{\pi}_i, \tag{5.12}$$

where $\boldsymbol{M}_i \coloneqq \boldsymbol{T}_i^{-1}$ can be associated to local inner products.

Again, we will impose conditions *(i)* and *(ii)* of continuity of face fluxes and face pressures. However, unlike in TPFA, we add them as separate equations. This yields the matrix system

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} & \boldsymbol{D} \\ \boldsymbol{C}^T & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{D}^T & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ -\boldsymbol{p} \\ \boldsymbol{\pi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{q} \\ \boldsymbol{0} \end{bmatrix}. \tag{5.13}$$

Here, $\boldsymbol{v}$ are the outward fluxes for each half-face of the cell. This gives that both v_{ij} and $-v_{ij}$ appears in $\boldsymbol{v}$ for each face F_{ij}. In the same manner, $\boldsymbol{p}$ and $\boldsymbol{\pi}$ contains the cell and face pressures. The matrix $\boldsymbol{B}$ is block diagonal, with diagonal $(\boldsymbol{M}_1, \dots, \boldsymbol{M}_{N_P})$, where N_P are the number of cells in the grid, and $\boldsymbol{C}$ is a block diagonal matrix with diagonal $(\boldsymbol{e}_1, \dots, \boldsymbol{e}_{N_P})$. Finally, a column of $\boldsymbol{D}$ corresponds to a unique face in the grid, and has two unit entries for each face.

We refer to (5.13) as the *hybrid formulation* of (5.8), see [20] for details. The mimetic finite difference method (MFD) [35] is a method that can be stated in this form [20].

In a first-order mimetic finite difference method, we approximate the pressure by a function that is piecewise linear on each cell, such that $p_i = \boldsymbol{x} \cdot \boldsymbol{a} + b$ for some vector $\boldsymbol{a}$ and constant b. The flux across $F_{i,j}$ and pressure drop from the cell centroid to the face centroid is then

$$v_{ij} = -|F_{i,j}| \boldsymbol{n}_{i,j}^T \boldsymbol{K}_i \boldsymbol{a}, \quad p_i - \pi_{ij} = \boldsymbol{c}_{i,j}^T \boldsymbol{a}.$$

We insert this into (5.12) to obtain the following consistency conditions for the local inner products:

$$\boldsymbol{MNK} = \boldsymbol{C}, \quad \boldsymbol{NK} = \boldsymbol{TC}, \tag{5.14}$$

where

$$\boldsymbol{N} = (|F_{i,1}|\boldsymbol{n}_{i,1}, \ldots, |F_{i,n_F}|\boldsymbol{n}_{i,n_F})^T, \quad \boldsymbol{C} = (\boldsymbol{c}_{i,1}, \ldots, \boldsymbol{c}_{i,n_F})^T.$$

Fro simplicity, we have omitted the cell index i. It can be shown that $\boldsymbol{C}^T\boldsymbol{N} = |K|\boldsymbol{I}$, see [7]. Multiplying the first consistency condition in (5.14) by $\boldsymbol{K}^{-1}\boldsymbol{C}^T\boldsymbol{N}$, we get that

$$\boldsymbol{M}\boldsymbol{N} = |K|^{-1}\boldsymbol{C}\boldsymbol{K}^{-1}\boldsymbol{C}^T\boldsymbol{N}.$$

Thus, a local inner product satisfying (5.14) can be written as

$$\boldsymbol{M} = |K|^{-1}\boldsymbol{C}\boldsymbol{K}^{-1}\boldsymbol{C}^T.$$

Similar to VEM, we must add a term to ensure stability. Thus, any valid local inner product can be written as

$$\boldsymbol{M} = |K|^{-1}\boldsymbol{C}\boldsymbol{K}^{-1}\boldsymbol{C}^T + \boldsymbol{M}_s,$$

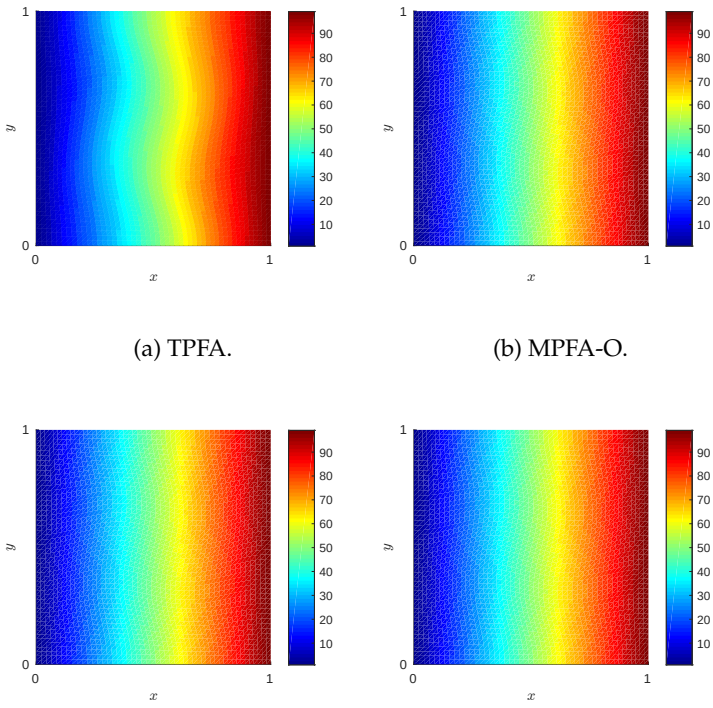
where $\boldsymbol{M}_s$ is a matrix defined such that $\boldsymbol{M}_s\boldsymbol{N} = 0$ and $\boldsymbol{M}$ is symmetric positive definite.

Remark 5.1. Notice the similarity between MFD and VEM: In both methods, for a given order k, we ensure that the local inner product is exact for all polynomials in $\mathbb{P}_k(K)$. For functions not in $\mathbb{P}_k(K)$, we add a stability term so that we are approximating the local inner product to the right order of magnitude. This is of course no coincidence; after all, MFD is the method from which VEM has evolved.

As mentioned in Chapter 1, since we are free to choose the stability term $\boldsymbol{M}_s$, we can, like with VEM, obtain certain other methods from MFD. For example, this is the case for TPFA, the Raviart-Thomas method [20], and some MPFA methods, like MPFA-O and the local flux mimetic MPFA [16, 21]. In our examples, we will consider the MPFA-O method.

The methods described above are all implemented in MRST. Note that these methods gives the pressure values at the face centroids and cell centroids. We know that for $k \geq 2$, VEM gives the average pressure for each cell, and we know from Section 5.3.1 that we can compute these averages for $k = 1$ as well. Hence, we can use these values to compare the methods. Before we do so, we stress the point that the TPFA, MPFA and MFD methods are all conservative in the discrete sense, and thus well-suited for reservoir simulation. For VEM, we do not have this property.

For simplicity, we will consider two-dimensional problems, and isotropic permeability, so that $\boldsymbol{K} = \boldsymbol{I}$. We will use the default stability term for the VEM methods. For the stability term $\boldsymbol{M}_s$ of the MFD method, we will use the MRST inner product `IP_SIMPLE`, see [20] for details.



(a) TPFA.

(b) MPFA-O.

(c) MFD.

(d) VEM.

Figure 5.7: Solutions to (5.15) using four different methods.

5.5.3 Pressure drop

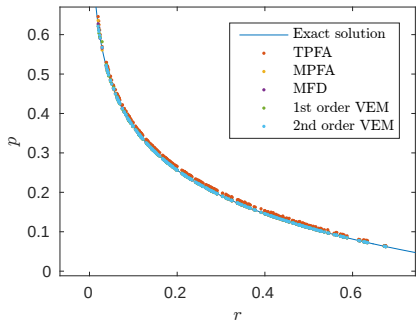
First, we consider a problem similar to one described in [20]. This a pressure drop problem, which we formulate as

$$\begin{aligned} \Delta p &= 0, & \boldsymbol{x} &\in \Omega, \\ \partial_n p &= 0, & \boldsymbol{x} &\in \Gamma_n \cup \Gamma_s, \\ p &= 100, & \boldsymbol{x} &\in \Gamma_e, \\ p &= 0, & \boldsymbol{x} &\in \Gamma_w, \end{aligned} \tag{5.15}$$

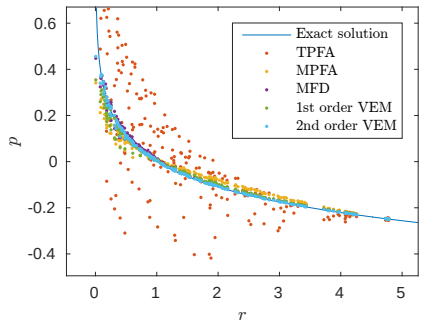
where $\Omega = [0,1]\times[0,1]$, and the the subscripts n, s, e and w refers to the northern, southern, eastern and western boundaries. We define a regular Cartesian grid $\mathcal{T}_{0.2,0.2}$ over Ω, and use the MRST function `twister` to deform $\mathcal{T}_{0.2,0.2}$ into a grid of quadrilaterals by adding $0.03\sin(\pi x)\sin(3\pi(y-0.5))$ to the x coordinates of the vertices, and subtracting the same quantity from the y coordinates. Note that the resulting grid does not have $\boldsymbol{K}\boldsymbol{n}_{ij}$ parallel to $\boldsymbol{c}_{ij}$, so the sufficient condition for $\boldsymbol{K}$-orthogonality mentioned above is not satisfied. The resulting solutions to

(5.15) using TPFA, MPFA, MFD and first order VEM using this grid are shown in Figure 5.7.

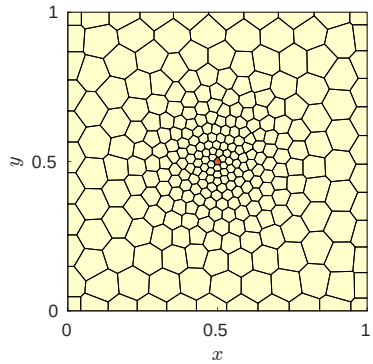
Since there is a given pressure difference between the eastern and western boundaries, no flow through the northern and southern boundaries, and no sources or sinks, we expect the pressure drop to be linear, decreasing from the eastern to the western boundary. We clearly see that the solution produced by TPFA is incorrect, while MFD, MPFA and first order VEM seems to produce correct solutions.



(a) Solutions on $\Omega = [0, 1] \times [0, 1]$.



(b) Solutions on $\Omega = [0, 10] \times [0, 1]$.



(c) Grid over $\Omega = [0, 1] \times [0, 1]$. Source cell in red.

Figure 5.8: Solutions and grid for (5.16).

5.5.4 Point source

Again, we consider a unit point source, placed at the point $\boldsymbol{c} \in \Omega \subset \mathbb{R}^2$. In a reservoir simulation setting, this is equivalent to a well with constant flux rate

normalized to 1. We know from Section 5.4 that this can be stated as

$$\begin{aligned} -\Delta p &= \delta(\boldsymbol{x}-\boldsymbol{c}), && \boldsymbol{x}\in\Omega \\ p &= -\frac{1}{2\pi}\log\left(|\boldsymbol{x}-\boldsymbol{c}|\right), && \boldsymbol{x}\in\partial\Omega. \end{aligned} \tag{5.16}$$

We consider a rectangular domain $\Omega = [0, H_x] \times [0, H_y]$, and place the point source in the middle of the domain, so that $\boldsymbol{c} = (H_x/2, H_y/2)^T$. Next, we generate a grid of polygons over Ω, refined around $\boldsymbol{c}$. Using this grid, we calculate the solutions to (5.16) when $\Omega = [0, 1] \times [0, 1]$. From [4], it is clear that this grid does indeed satisfy the mentioned sufficient condition of $\boldsymbol{K}$-orthogonality. To study aspect ratio effects, we also stretch the same grid in the x-direction, so that it covers the domain $\Omega = [0, 10] \times [0, 1]$. This grid no longer satisfies the sufficient $\boldsymbol{K}$-orthogonality condition. We can exploit the radial symmetry of the exact solution, and plot the cell pressures with respect to the distance $r := |\boldsymbol{x}_K - \boldsymbol{c}|$ from the cell centroid to $\boldsymbol{c}$. The results are shown in Figure 5.8.

For the first grid, the results produced by all methods are close to the exact solution. The MPFA, MFD and VEM solutions are almost indistinguishable from the exact solution, while the TPFA solution is slightly higher. For the second grid, we see that the TPFA method suffers from the high aspect ratio, and is obviously not correct. The MPFA method exhibits some grid orientation effect as well. Slightly better is the first order VEM, but we do observe some of the same behavior as we saw in Section 5.4. The MFD and second order VEM both produce satisfactory solutions.

5.6 Final remarks

We have now seen that the MATLAB implementation behaves as expected with respect to convergence for simple problems in two and three dimensions. Moreover, it produces satisfactory results when compared with other methods for simple test cases. Since the implementation is compatible with MRST, this facilitates comparison of VEM with the industry standards of reservoir simulation, both with respect to discretization methods and grids.

Finally, in light of Section 5.4, we note that there are several applications when choosing the stability term is useful, and, in some cases, necessary, to obtain an accurate solution. For example, in [26], it is described how the 3D stencils from Appendix B, with $h_x = h_y = h_z$, can be combined to solve 3D visco-acoustic wave propagation problems while minimizing the dispersion. Theorem 3.29 tells us how to choose the stability term in order to obtain any combination of these stencils, and the same dispersion analysis thus applies to VEM as well. Another possible usage of the stability term is to address the lack of monotonicity of solutions in reservoir simulation. There exists several studies addressing this issue, particularly for the MPFA method [24, 25]. If one can use the stability term in VEM in order to ensure monotonicity for arbitrary grids, this would be a great advantage. In general, we have the possibility of adjusting the stability term in order to avoid unphysical solutions to the problem we are solving.

Chapter 6

Conclusion and Closing Remarks

Finally, after six months or so, you find the light switch, you turn it on, and suddenly it's all illuminated.

Andrew John Wiles

6.1 Conclusion

The most important theoretical aspects of the virtual element method for Poisson problems have been presented and summarized. The theory has also been supplemented with definitions, remarks and theorems. In particular, we have provided detailed proofs of the well-posedness of the projection operator Π^{∇}, and the fact that the projection of any function in the virtual element space can be computed exactly.

We have also shown how to construct the exact stability term s^K, which is the stability term that makes the approximated bilinear form of the virtual element method equal to the exact bilinear form. By analysing how the projection operator Π^{∇} behaves under affine transformations, we have shown that we can choose the stability term such that the local stiffness matrices of the first order finite element and virtual element methods for Poisson problems are equal for certain cell geometries. These are triangles, tetrahedra, parallelograms, parallelepipeds and triangular prisms with parallelogram sides. We have also shown how we can scale the exact stability term in order to make the linear system of equations obtained from VEM equal to that of finite difference discretizations of Poisson problems, both in two and three dimensions.

The implementation process of VEM for Poisson problems has been described in detail, while justifying some of the choices in the definitions of the VEM function spaces and projection operators. A method of approximating the

local load term has been presented. Moreover, we have discussed how to implement inhomogeneous Dirichlet and Neumann boundary conditions. Building upon this, VEM has been implemented in MATLAB using the open-source MATLAB reservoir simulation toolbox (MRST). The implementation supports two and three dimensions, and first and second order. We believe this is the first implementation of higher-order VEM for Poisson problems in three dimensions.

The VEM implementation has been applied to two test cases, one in $\mathbb{R}^2$ and one in $\mathbb{R}^3$, on randomly generated polygonal and polyhedral grids. The convergence rates where found to be close to $k+1$, where k is the order of the method, which is in accordance with the theory. We have also used the implementation to see how the stability term can be used to minimize a given norm of the error of the approximated solution. Finally, we have given a short introduction to some of the most common methods used in reservoir simulations: TPFA, MPFA and MFD, and the VEM implementation has been compared to these methods for two test cases on polygonal grids. The solutions produced by VEM was similar to or better than the solutions produced by TPFA, MPFA and MFD.

6.2 Closing Remarks

The virtual element method has opened a whole new world of interesting problems within numerical mathematics, both theoretical, and technical. One of the most interesting features of VEM is the stability term, or rather the freedom in choosing it. We have seen that it is possible to obtain both finite element and finite difference discretizations from suitable choices of this term, and it is reasonable to assume that it is possible to obtain several other methods from VEM as well. Such equivalences are of particular interest, since we then know that the properties of these methods also applies to VEM. Moreover, the equivalence might provide useful information about the other method based on the existing theory for VEM, and give a natural way to extend the method to more general cell geometries.

In FEM simulations, a common way of obtaining a sufficiently accurate solution to a problem is to refine the grid. This is usually done by refining the cells with the largest contribution to the error [14, 29]. However, because of the restrictions on the cell geometries, this is not trivial. Due to the flexibility of VEM with respect to cell geometries, it is well-suited for such refinement techniques. Another approach is order refinement, in which one uses a discretization of higher order on a subset of the cells where the error is expected to be large [29]. To preserve continuity of the calculated solution, it is then necessary to match the degrees of freedom in neighboring cells. Again, since VEM is flexible with respect to cell geometries, it is possible to split the common face of the neighboring cell with the lowest order into several faces in order to add more degrees of freedom to this cell, and consequently match them with the degrees of freedom of the neighboring cell.

Other interesting extensions of this work includes formulating VEM for more general elliptic problems [34]. In particular, we saw in Chapter 2 that in reservoir simulations, the pressure equation is formulated using the operator $\nabla \cdot \boldsymbol{K} \nabla$.

A first step in applying VEM to reservoir simulations is then to discretize this operator in the VEM framework. Moreover, as mentioned in Chapter 1, it is desirable with a conservative method in reservoir simulations. In light of this, it would be interesting to investigate VEM in the framework of mixed and hybrid finite element methods [6], in order to obtain a conservative VEM discretization. However, due to the time available for writing this thesis, we will have to leave these problems unexplored for now.

Appendix A

Finite Element Function Spaces

The definitions given here can be found in [29]. For the sake of self-containedness, they are also included here.

Definition A.1. The Lobatto shape functions are defined as follows:

$$l_0(x) = \frac{1-x}{2}, \quad l_1(x) = \frac{1+x}{2}, \quad l_k(x) = \sqrt{\frac{2k-1}{2}} \int_{-1}^{x} L_{k-1}(\xi)\,\mathrm{d}\xi,\ k \geq 2,$$

where $L_k(x)$ are the Legendre polynomials,

$$L_k(x) = \frac{1}{2^k k!} \frac{\mathrm{d}^k}{\mathrm{d}x^k}(x^2-1)^k, \quad k = 0, 1, \ldots.$$

We also define the *kernel functions* $\kappa_k(x)$, satisfying

$$l_k(x) = l_0(x) l_1(x) \kappa_{k-2}(x).$$

∘

Definition A.2. Let $K_t = \{(x,y) \in \mathbb{R}^2 : -1 < x, y, x+y \leq 0\} \subset \mathbb{R}^2$ be the reference triangle in Figure 3.3. We denote the affine coordinates on K_t by

$$\xi_1(x,y) = l_1(y), \quad \xi_2(x,y) = 1 - l_1(x) - l_1(y), \quad \xi_3(x,y) = l_1(x).$$

finite element space for K_t is denoted $Q_k(K_t)$, and its basis functions are as follows:

♢ Vertex interpolants $\{\varphi_t^{V_i}\}_{i=1}^3$, which takes on the value 1 at one of the vertices, and vanish at all the others. We express them through the affine coordinates:

$$\varphi_t^{V_1}(x,y) = \xi_2, \quad \varphi_t^{V_2}(x,y) = \xi_3, \quad \varphi_t^{V_3}(x,y) = \xi_1.$$

♢ Edge functions $\{\varphi_t^{E_i,p}\}_{i=1,p=2}^{3,k}$, whose trace coincides with the Lobatto shape function l_p on edge i, and vanishes at all the others. We express the through the Lobatto shape functions as follows:

$$\varphi_t^{E_1,p}(x,y) = \xi_2\xi_3\kappa_{p-2}(\xi_3-\xi_2),$$
$$\varphi_t^{E_2,p}(x,y) = \xi_3\xi_1\kappa_{p-2}(\xi_1-\xi_3),$$
$$\varphi_t^{E_3,p}(x,y) = \xi_1\xi_2\kappa_{p-2}(\xi_2-\xi_1).$$

♢ Bubble functions $\{\varphi_t^{B,p,q}\}_{p=1,q=1}^{p+q\leq k-1}$, vanishing on the boundary:

$$\varphi_t^{B,p,q}(x,y) = \xi_1\xi_2\xi_3\kappa_{p-1}(\xi_3-\xi_2)\kappa_{q-1}(\xi_2-\xi_1).$$

∘

Definition A.3. Let $K_T = \{(x,y,z)\in\mathbb{R}^3 : -1 < x,y,z, \quad x+y+z < -1\}$ be the reference tetrahedron in Figure 3.4. We denote the affine coordinates on K_T by

$$\xi_1 = l_1(y), \quad \xi_2 = 1 - l_1(x) - l_1(y) - l_1(z), \quad \xi_3 = l_1(x), \quad \xi_4 = l_1(z).$$

Note that ξ_i vanishes at face F_i. Using these, we define the local FEM space for K_T: The local finite element space for K_T is denoted $\mathbb{Q}_k(K_T)$, and is defined as follows:

♢ Vertex interpolants $\{\varphi_T^{V_i}\}_{i=1}^4$, which takes on the value 1 at one of the vertices, and vanish at all the others. We express them through the Lobatto shape functions by

$$\varphi_T^{V_i}(x,y,z) = \xi_{i_1},$$

where F_{i_1} corresponds to the only face not containing vertex V_i.

♢ Edge functions $\{\varphi_T^{E_i,p}\}_{i=1,p=2}^{4,k}$, whose trace coincides with the Lobatto shape function l_p on edge i, and vanishes at all the others. Let E_i be the edge from V_{j_1} to V_{j_2}, and let F_{i_1} and F_{i_2} denote the two faces sharing a single vertex V_{j_1} or V_{j_2} with E_i. Then,

$$\varphi_T^{E_i,p}(x,y,z) = \xi_{i_1}\xi_{i_2}\kappa_{k-2}(\xi_{i_1}-\xi_{i_2}).$$

♢ Face functions $\{\varphi_T^{F_i,p,q}\}_{i=1,p=1,q=1}^{4,p+q\leq k-1}$, whose trace is of polynomial order $p+q+1$ on face F_i, and vanishes at all the others. Let V_a, V_b and V_c be the vertices of face F_i, ordered such that $a<b<c$. Moreover, let ξ_a, ξ_b and ξ_c be the affine coordinates such that $\xi_a(V_a) = \xi_b(V_b) = \xi_c(V_c) = 1$. Then,

$$\varphi_T^{F_i,p,q}(x,y,z) = \xi_a\xi_b\xi_c\kappa_{p-1}(\xi_b-\xi_a)\kappa_{q-1}(\xi_a-\xi_c).$$

♢ Bubble functions $\{\varphi_T^{Bp,q,r}\}_{p=1,q=1,r=1}^{p+q+r\leq k-1}$, vanishing on the boundary:

$$\varphi_T^{B,p,q,r}(x,y) = \kappa_{p-1}(\xi_1-\xi_2)\kappa_{q-1}(\xi_3-\xi_2)\kappa_{r-1}(\xi_4-\xi_2)\prod_{i=1}^{4}\xi_i.$$

○

Definition A.4. Let $K_s = [-1,1]\times[-1,1] \subset \mathbb{R}^2$ be the reference square in Figure 3.5. The local finite element space for K_s is denoted $Q_k(K_s)$, and can be expressed through the Lobatto shape functions:

$$\varphi_s^{p,q}(x,y) = l_p(x)l_q(y), \quad 0 \leq p,q \leq k.$$

They are ordered as follows:

♢ Vertex interpolants $\{\varphi_s^{V_i}\}_{i=1}^4$, which takes on the value 1 at one of the vertices, and vanish at all the others:

$$\varphi_s^{V_i}(x,y) = l_{d_x}(x)l_{d_y}(y),$$

where $\boldsymbol{d} = (d_x, d_y)$ is a multiindex defined as follows: Let E_x and E_y be the two edges containing vertex V_i, oriented parallel to the coordinate axes x and y, respectively. Then

$$d_\xi = \begin{cases} 0, & V_i \text{ is on the left of } E_\xi \\ 1, & V_i \text{ is on the right of } E_\xi \end{cases}.$$

♢ Edge functions $\{\varphi_s^{E_i,p}\}_{i=1,p=2}^{4,k}$, whose trace coincides with the Lobatto shape function l_p on edge i, and vanishes at all the others:

$$\varphi_s^{E_i,p}(x,y) = l_{d_x}(x)l_{d_y}(y),$$

where $\boldsymbol{d} = (d_x, d_y)$ is a multiindex defined as follows:

$$d_\xi = \begin{cases} p, & E_i \text{ is parallell to coordinate axis } \xi, \\ 0, & E_i \text{ is on the left of } K_s \text{ w.r.t. the remaining coordinate axis}, \\ 1, & E_i \text{ is on the right of } K_s \text{ w.r.t. the remaining coordinate axis}. \end{cases}$$

♢ Bubble functions $\{\varphi_s^{B,p,q}\}_{p=2,q=2}^{k,k}$, vanishing on the boundary:

$$\varphi_s^{B,p,q}(x,y) = l_p(x)l_q(y).$$

○

Definition A.5. Let $K_C = [-1,1]\times[-1,1]\times[-1,1] \subset \mathbb{R}^3$ be the reference cube in Figure 3.6. The local finite element space for K_C is denoted $\mathbb{Q}_k(K_C)$, and can

be expressed through the Lobatto shape functions:

$$\varphi_C^{p,q,r}(x,y) = l_p(x)l_q(y)l_r(z), \quad 0 \leq p,q,r \leq k.$$

They are ordered as follows:

♢ Vertex interpolants $\{\varphi_C^{V_i}\}_{i=1}^8$, which takes on the value 1 at one of the vertices, and vanish at all the others:

$$\varphi_C^{V_i}(x,y,z) = l_{d_x}(x)l_{d_y}(y)l_{d_z}(z)$$

where $\boldsymbol{d} = (d_x, d_y, d_z)$ is a multiindex defined as follows: Let E_x, E_y and E_z be the three edges containing vertex V_i, oriented parallel to the coordinate axes. Then

$$d_\xi = \begin{cases} 0, & V_i \text{ is on the left of } E_\xi \\ 1, & V_i \text{ is on the right of } E_\xi \end{cases}.$$

♢ Edge functions $\{\varphi_C^{E_i,p}\}_{i=1,p=2}^{12,k}$, whose trace coincides with the Lobatto shape function l_p on edge E_i, and vanishes at all the others:

$$\varphi_C^{E_i,p}(x,y,z) = l_{d_x}(x)l_{d_y}(y)l_{d_z}(z),$$

where $\boldsymbol{d} = (d_x, d_y, d_z)$ is a multiindex defined as follows: Let

$$d_\xi = \begin{cases} p, & E_i \text{ is parallell to coordinate axis } \xi, \\ 0, & E_i \text{ is on the left of } K_C \text{ w.r.t. the remaining coordinate axes}, \\ 1, & E_i \text{ is on the right of } K_C \text{ w.r.t. the remaining coordinate axes}. \end{cases}$$

♢ Face functions $\{\varphi_C^{F_i,p,q}\}_{i=1,p=2,q=2}^{6,k,k}$, whose trace is of directional polynomial orders p, q on face F_i, and vanishes at all the others:

$$\varphi_C^{F_i,p,q}(x,y,z) = l_{d_x}(x)l_{d_y}(y)l_{d_z}(z),$$

where $\boldsymbol{d} = (d_x, d_y, d_z)$ is a multiindex defined as follows:

$$d_\eta = \begin{cases} p, & F_i \text{ spans the plane } (\eta,\zeta), \\ q, & F_i \text{ spans the plane } (\xi,\eta), \\ 0, & F_i \text{ is on the left of } K_C \text{ w.r.t. the remaining coordinate axis}, \\ 1, & F_i \text{ is on the right of } K_C \text{ w.r.t. the remaining coordinate axis}. \end{cases}$$

♢ Bubble functions $\{\varphi_C^{B,p,q,r}\}_{p=2,q=2,r=2}^{k,k,k}$, vanishing on the boundary:

$$\varphi_C^{B,p,q,r}(x,y,z) = l_p(x)l_q(y)l_r(z).$$

○

Definition A.6. Let K_P be the reference prism

$$K_P = \{(x, y, z) \in \mathbb{R}^3 \,:\, -1 < x, y, z, \quad x + y < 0, z < 1\}$$

in Figure 3.8. We denote the affine coordinates on K_P by

$$\begin{aligned}
\xi_1(x, y, z) &= l_1(y), \\
\xi_2(x, y, z) &= 1 - l_1(x) - l_1(y), \\
\xi_3(x, y, z) &= l_1(x), \\
\xi_4(x, y, z) &= l_1(z), \\
\xi_5(x, y, z) &= l_0(z).
\end{aligned}$$

Note that each of the affine coordinates vanishes at one of the faces of K_P. The local kth order finite element space for K_P is denoted $\mathbb{Q}_k(K_P)$, and is ordered as follows:

- ♢ Vertex interpolants $\{\varphi^{V_i}\}_{i=1}^{6}$, which takes on the value 1 at one of the vertices, and vanish at all the others. We express them through the affine coordinates by
$$\varphi^{V_i}(x, y, z) = \xi_{i_1}\xi_{i_2},$$
where F_{i_1} and F_{i_2} corresponds to the only two faces not containing vertex V_i.

- ♢ Edge functions $\{\varphi^{E_i,p}\}_{i=1,p=2}^{9,k}$, whose trace coincides with the Lobatto shape function l_p on edge i, and vanishes at all the others. Let E_i be the edge from V_{i_1} to V_{i_2}, and let F_{j_1} and F_{j_2} be the two faces sharing a single vertex V_{i_1} or V_{i_2} with E_i, respectively. Let F_{j_3} be the face not containing any of the vertices of E_i. The edge functions for edge E_i is then
$$\varphi^{E_i,p}(x, y, z) = \xi_{j_1}\xi_{j_2}\kappa_{k-2}(\xi_{j_1} - \xi_{j_2})\xi_{j_3}.$$

- ♢ Triangular face functions $\{\varphi^{F_i,p,q}\}_{i=4,p=1,q=1}^{5,p+q\leq k-1}$, whose trace is of local polynomial order r on the triangular face F_i, with $3 \leq r \leq k$ and $1 \leq p, q$, $p + q \leq r - 1$, and vanishes at all the others. Let V_a, V_b and V_c be the vertices of face F_i, ordered such that $a < b < c$. Moreover, let ξ_a, ξ_b and ξ_c be the affine coordinates such that $\xi_a(V_a) = \xi_b(v_b) = \xi_c(v_c) = 1$. Let ξ_d be the affine coordinate corresponding to the other triangular face. Then,
$$\varphi^{F_i,p,q}(x, y, z) = \xi_a\xi_b\xi_c\kappa_{p-1}(\xi_b - \xi_a)\kappa_{q-1}(\xi_a - \xi_c)\xi_d.$$

- ♢ Rectangular face functions $\{\varphi^{F_i,p,q}\}_{i=1,p=2,q=2}^{3,k,k}$, whose trace is of local directional polynomial order p, q, with $2 \leq p, q \leq k$, and vanishes at all the others. There is a unique pair of edges parallel to the xy-plane, belonging to face F_i. We select the edge E_j going from V_{j_1} to V_{j_2}, belonging to the bottom face F_4. Further, we denote the pair of faces sharing a single vertex V_{j_1} or V_{j_2} with edge E_j by F_{i_1} and F_{i_2}, respectively. The face function for

face F_i is then

$$\varphi^{F_i,p,q}(x,y,z) = \xi_{i_1}\xi_{i_2}\xi_4\xi_5\kappa_{p-2}(\xi_{i_1} - \xi_{i_2})\kappa_{q-2}(\xi_4 - \xi_5).$$

♢ Bubble functions $\{\varphi_B^{p,q,r}\}_{p=1,q=1,r=2}^{p+q\leq k-1,k}$, vanishing on the boundary:

$$\varphi_B^{p,q,r}(x,y) = \xi_1\xi_2\xi_3\kappa_{p-1}(\xi_3 - \xi_2)\kappa_{q-1}(\xi_2 - \xi_1)l_r(z).$$

○

Appendix B

Finite Difference Stencils

Definition B.1. Let $\Omega = [-H_x, H_x] \times [-H_y, H_y] \subset \mathbb{R}^2$, and let $\mathcal{T}_{h_x,h_y}$ be a regular Cartesian grid over Ω. We denote the aspect ratio by $\epsilon = h_y/h_x$. Finally, let $u : \Omega \to \mathbb{R}$ be a smooth function. We define the following two finite difference stencils for the discrete Laplace operator:

(i) Stencil 1: The stencil using the usual Cartesian coordinate system. It is shown in Figure 3.10a. The discretizations of the second order partial derivatives are

$$\left[\frac{\partial^2 u}{\partial x^2}\right]_{i,j} = \frac{1}{4h_x^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}),$$
$$\left[\frac{\partial^2 u}{\partial y^2}\right]_{i,j} = \frac{1}{4h_y^2}(u_{i,j-1} - 2u_{i,j} + u_{i,j+1}).$$

We denote the resulting matrix multiplied by the area $4h_xh_y$ by $\boldsymbol{F}_c$.

(ii) Stencil 2: The stencil using the coordinate system (d_1, d_2), where d_i are unit vectors parallel to the diagonals of the square. The stencil is shown in Figure 3.10b. We have that

$$\left[\frac{\partial^2 u}{\partial d_1^2}\right]_{i,j} = \frac{1}{4(h_x^2 + h_y^2)}(u_{i-1,j-1} - 2u_{i,j} + u_{i+1,j+1}),$$
$$\left[\frac{\partial^2 u}{\partial d_2^2}\right]_{i,j} = \frac{1}{4(h_x^2 + h_y^2)}(u_{i-1,j+1} - 2u_{i,j} + u_{i+1,j-1}),$$
$$\left[\frac{\partial^2 u}{\partial d_1 d_2}\right]_{i,j} = \frac{1}{4(h_x^2 + h_y^2)}(u_{i,j-1} - u_{i-1,j} - u_{i+1,j} + u_{i,j+1}),$$

and the Laplacian in the Cartesian coordinate system is

$$(d_1, d_2) : \Delta = \frac{1}{4}(\epsilon + \epsilon^{-1})^2\left(\frac{\partial^2}{\partial d_1^2} + \frac{\partial^2}{\partial d_2^2}\right) - \frac{1}{2}(\epsilon^2 - \epsilon^{-2})\frac{\partial^2}{\partial d_1 d_2}.$$

We denote the resulting matrix multiplied by the area $4h_xh_y$ by $\boldsymbol{F}_d$.

∘

The following stencil is also defined for $h_x = h_y = h_z$ in [26].

Definition B.2. Let $\Omega = [-H_x, H_x] \times [-H_y, H_y] \times [-H_z, H_z] \subset \mathbb{R}^3$, and $\mathcal{T}_{h_x,h_y,h_z}$ be a regular Cartesian grid over Ω. Let

$$\epsilon_{yx} = \frac{h_y}{h_x}, \quad \epsilon_{xz} = \frac{h_x}{h_z}, \quad \epsilon_{zy} = \frac{h_z}{h_y}$$

express the aspect ratios. Finally, let $u : \Omega \to \mathbb{R}$ be a smooth function. We define the following three finite difference stencils for the discrete Laplace operator:

(i) Stencil 1: The stencil using the usual Cartesian coordinate system (x, y, z), shown in Figure 3.11a. The discretizations of the second order partial derivatives are

$$\begin{aligned}
\left[\frac{\partial^2 u}{\partial x^2}\right]_{i,j,k} &= \frac{1}{4h_x^2}(u_{i-1,j,k} - 2u_{i,j,k} + u_{i+1,j,k}),\\
\left[\frac{\partial^2 u}{\partial y^2}\right]_{i,j,k} &= \frac{1}{4h_y^2}(u_{i,j-1,k} - 2u_{i,j,k} + u_{i,j+1,k}),\\
\left[\frac{\partial^2 u}{\partial z^2}\right]_{i,j,k} &= \frac{1}{4h_z^2}(u_{i,j,k-1} - 2u_{i,j,k} + u_{i,j,k+1}).
\end{aligned}$$

We denote the resulting matrix multiplied by the volume $8h_xh_yh_z$ by $\boldsymbol{F}_c$.

(ii) Stencil 2: The average of three stencils using the coordinate systems (x, η_x, ζ_x), (ξ_y, y, ζ_y) and (ξ_z, η_z, z), obtained by rotating the Cartesian coordinate system counter-clockwise around one of its axes. The stencil using coordinate system (x, η_x, ζ_x) is shown in Figure 3.11b. The discretizations of the second order partial derivatives ∂_x^2, ∂_y^2, and ∂_z^2 are as for Stencil 1, and

$$\begin{aligned}
\left[\frac{\partial^2 u}{\partial \eta_x^2}\right]_{i,j,k} &= \frac{1}{4(h_y^2 + h_z^2)}(u_{i,j-1,k-1} - 2u_{i,j,k} + u_{i,j+1,k+1}),\\
\left[\frac{\partial^2 u}{\partial \zeta_x^2}\right]_{i,j,k} &= \frac{1}{4(h_y^2 + h_z^2)}(u_{i,j-1,k+1} - 2u_{i,j,k} + u_{i,j+1,k-1}),\\
\left[\frac{\partial^2 u}{\partial \xi_y^2}\right]_{i,j,k} &= \frac{1}{4(h_x^2 + h_z^2)}(u_{i-1,j,k+1} - 2u_{i,j,k} + u_{i+1,j,k-1}),\\
\left[\frac{\partial^2 u}{\partial \zeta_y^2}\right]_{i,j,k} &= \frac{1}{4(h_x^2 + h_z^2)}(u_{i-1,j,k-1} - 2u_{i,j,k} + u_{i+1,j,k+1}),\\
\left[\frac{\partial^2 u}{\partial \xi_z^2}\right]_{i,j,k} &= \frac{1}{4(h_x^2 + h_y^2)}(u_{i-1,j-1,k} - 2u_{i,j,k} + u_{i+1,j+1,k}),\\
\left[\frac{\partial^2 u}{\partial \eta_z^2}\right]_{i,j,k} &= \frac{1}{4(h_x^2 + h_y^2)}(u_{i-1,j+1,k} - 2u_{i,j,k} + u_{i+1,j-1,k}),
\end{aligned}$$

$$\left[\frac{\partial^2 u}{\partial \eta_x \zeta_x}\right]_{i,j,k} = \frac{1}{4(h_y^2+h_z^2)}(u_{i,j,k-1} - u_{i,j-1,k} - u_{i,j+1,k} + u_{i,j,k+1}),$$
$$\left[\frac{\partial^2 u}{\partial \xi_y \zeta_y}\right]_{i,j,k} = \frac{1}{4(h_x^2+h_z^2)}(u_{i-1,j,k} - u_{i,j,k-1} - u_{i,j,k+1} + u_{i+1,j,k}),$$
$$\left[\frac{\partial^2 u}{\partial \xi_z \eta_z}\right]_{i,j,k} = \frac{1}{4(h_x^2+h_y^2)}(u_{i,j-1,k} - u_{i-1,j,k} - u_{i+1,j,k} + u_{i,j+1,k}).$$

The Laplacian in the Cartesian coordinate system is

$$(x,\eta_x,\zeta_x) : \Delta = \frac{\partial^2}{\partial x^2} + \frac{1}{4}(\epsilon_{zy}+\epsilon_{zy}^{-1})^2\left(\frac{\partial^2}{\partial \eta_x^2}+\frac{\partial^2}{\partial \zeta_x^2}\right) - \frac{1}{2}(\epsilon_{zy}^2-\epsilon_{zy}^{-2})\frac{\partial^2}{\partial \eta_x \zeta_x},$$
$$(\xi_y,y,\zeta_y) : \Delta = \frac{\partial^2}{\partial y^2} + \frac{1}{4}(\epsilon_{xz}+\epsilon_{xz}^{-1})^2\left(\frac{\partial^2}{\partial \xi_y^2}+\frac{\partial^2}{\partial \zeta_y^2}\right) - \frac{1}{2}(\epsilon_{xz}^2-\epsilon_{xz}^{-2})\frac{\partial^2}{\partial \xi_y \zeta_y},$$
$$(\xi_z,\eta_z,z) : \Delta = \frac{\partial^2}{\partial z^2} + \frac{1}{4}(\epsilon_{yx}+\epsilon_{yx}^{-1})^2\left(\frac{\partial^2}{\partial \xi_z^2}+\frac{\partial^2}{\partial \eta_z^2}\right) - \frac{1}{2}(\epsilon_{yx}^2-\epsilon_{yx}^{-2})\frac{\partial^2}{\partial \xi_z \eta_z}.$$

We denote the resulting matrices multiplied by the volume $8h_xh_yh_z$ by $\boldsymbol{F}_x$, $\boldsymbol{F}_y$ and $\boldsymbol{F}_z$, respectively, and use its average $\boldsymbol{F}_{\boldsymbol{x}} \coloneqq (\boldsymbol{F}_x + \boldsymbol{F}_y + \boldsymbol{F}_z)/3$.

(iii) Stencil 3: The average of four stencils using the coordinate systems

$$(d_1,d_2,d_3), \quad (d_1,d_2,d_4), \quad (d_1,d_3,d_4), \quad (d_2,d_3,d_4),$$

where d_i are unit vectors parallel to the main diagonals of the cube. The average stencil is shown in Figure 3.11c. Defining $h^2 = h_x^2 + h_y^2 + h_z^2$, we have that

$$\left[\frac{\partial^2 u}{\partial d_1^2}\right]_{i,j,k} = \frac{1}{4h^2}(u_{i-1,j-1,k-1} - 2u_{i,j,k} + u_{i+1,j+1,k+1}),$$
$$\left[\frac{\partial^2 u}{\partial d_2^2}\right]_{i,j,k} = \frac{1}{4h^2}(u_{i-1,j+1,k-1} - 2u_{i,j,k} + u_{i+1,j-1,k+1}),$$
$$\left[\frac{\partial^2 u}{\partial d_3^2}\right]_{i,j,k} = \frac{1}{4h^2}(u_{i-1,j-1,k+1} - 2u_{i,j,k} + u_{i+1,j+1,k-1}),$$
$$\left[\frac{\partial^2 u}{\partial d_4^2}\right]_{i,j,k} = \frac{1}{4h^2}(u_{i-1,j+1,k+1} - 2u_{i,j,k} + u_{i+1,j-1,k-1}),$$
$$\left[\frac{\partial^2 u}{\partial d_1 d_2}\right]_{i,j,k} = \frac{1}{4h^2}(u_{i-1,j,k-1} - u_{i,j-1,k} - u_{i,j+1,k} + u_{i+1,j,k+1}),$$
$$\left[\frac{\partial^2 u}{\partial d_1 d_3}\right]_{i,j,k} = \frac{1}{4h^2}(u_{i-1,j-1,k} - u_{i,j,k-1} - u_{i,j,k+1} + u_{i+1,j+1,k}),$$
$$\left[\frac{\partial^2 u}{\partial d_1 d_4}\right]_{i,j,k} = \frac{1}{4h^2}(-u_{i,j-1,k-1} + u_{i-1,j,k} + u_{i+1,j,k} - u_{i,j+1,k+1}),$$
$$\left[\frac{\partial^2 u}{\partial d_2 d_3}\right]_{i,j,k} = \frac{1}{4h^2}(-u_{i,j-1,k+1} + u_{i-1,j,k} + u_{i+1,j,k} - u_{i,j+1,k-1}),$$

$$\left[\frac{\partial^2 u}{\partial d_2 d_4}\right]_{i,j,k} = \frac{1}{4h^2}(u_{i-1,j+1,k} - u_{i,j,k-1} - u_{i,j,k+1} + u_{i+1,j-1,k}),$$
$$\left[\frac{\partial^2 u}{\partial d_3 d_4}\right]_{i,j,k} = \frac{1}{4h^2}(u_{i-1,j,k+1} - u_{i,j-1,k} - u_{i,j+1,k} + u_{i+1,j,k-1}).$$

The Laplacian in the Cartesian coordinate system is

(d_1, d_2, d_3) :
$$\Delta = h^2\left(\frac{1}{4}\left(\frac{1}{h_y^2}+\frac{1}{h_z^2}\right)\frac{\partial^2}{\partial d_1^2} + \frac{1}{4}\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}\right)\frac{\partial^2}{\partial d_2^2} + \frac{1}{4}\left(\frac{1}{h_x^2}+\frac{1}{h_z^2}\right)\frac{\partial^2}{\partial d_3^2} - \frac{1}{2h_y^2}\frac{\partial^2}{\partial d_1 d_2} - \frac{1}{2h_z^2}\frac{\partial^2}{\partial d_1 d_3} + \frac{1}{2h_x^2}\frac{\partial^2}{\partial d_2 d_3}\right)$$

(d_1, d_2, d_4) :
$$\Delta = h^2\left(\frac{1}{4}\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}\right)\frac{\partial^2}{\partial d_1^2} + \frac{1}{4}\left(\frac{1}{h_y^2}+\frac{1}{h_z^2}\right)\frac{\partial^2}{\partial d_2^2} + \frac{1}{4}\left(\frac{1}{h_x^2}+\frac{1}{h_z^2}\right)\frac{\partial^2}{\partial d_4^2} - \frac{1}{2h_y^2}\frac{\partial^2}{\partial d_1 d_2} + \frac{1}{2h_x^2}\frac{\partial^2}{\partial d_1 d_4} - \frac{1}{2h_z^2}\frac{\partial^2}{\partial d_2 d_4}\right)$$

(d_1, d_3, d_4) :
$$\Delta = h^2\left(\frac{1}{4}\left(\frac{1}{h_x^2}+\frac{1}{h_z^2}\right)\frac{\partial^2}{\partial d_1^2} + \frac{1}{4}\left(\frac{1}{h_y^2}+\frac{1}{h_z^2}\right)\frac{\partial^2}{\partial d_3^2} + \frac{1}{4}\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}\right)\frac{\partial^2}{\partial d_4^2} - \frac{1}{2h_z^2}\frac{\partial^2}{\partial d_1 d_3} + \frac{1}{2h_x^2}\frac{\partial^2}{\partial d_1 d_4} - \frac{1}{2h_y^2}\frac{\partial^2}{\partial d_3 d_4}\right)$$

(d_2, d_3, d_4) :
$$\Delta = h^2\left(\frac{1}{4}\left(\frac{1}{h_x^2}+\frac{1}{h_z^2}\right)\frac{\partial^2}{\partial d_2^2} + \frac{1}{4}\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}\right)\frac{\partial^2}{\partial d_3^2} + \frac{1}{4}\left(\frac{1}{h_y^2}+\frac{1}{h_z^2}\right)\frac{\partial^2}{\partial d_4^2} + \frac{1}{2h_x^2}\frac{\partial^2}{\partial d_2 d_3} - \frac{1}{2h_z^2}\frac{\partial^2}{\partial d_2 d_4} - \frac{1}{2h_y^2}\frac{\partial^2}{\partial d_3 d_4}\right).$$

We will denote the resulting matrices multiplied by the volume $8h_x h_y h_z$ by $\boldsymbol{F}_1, \boldsymbol{F}_2, \boldsymbol{F}_3$ and $\boldsymbol{F}_4$, and use its average $\boldsymbol{F}_d \coloneqq (\boldsymbol{F}_1 + \boldsymbol{F}_2 + \boldsymbol{F}_3 + \boldsymbol{F}_4)/4$.

∘

References

[1] I. Aavatsmark. "An introduction to multipoint flux approximations for quadrilateral grids". In: *Computational Geosciences* 6.3-4 (2002), pp. 405–432.

[2] L. Agélas, C. Guichard, and R. Masson. "Convergence of finite volume MPFA O type schemes for heterogeneous anisotropic diffusion problems on general meshes". In: *International Journal on Finite Volumes* (2010), volume–7.

[3] B. Ahmad et al. "Equivalent projectors for virtual element methods". In: *Computers & Mathematics with Applications* 66.3 (2013), pp. 376 –391.

[4] R. L. Berge. "Unstructured pebi grids adapting to geological feautres in subsurface reservoirs". MA thesis. Norwegian University of Science and Technology, 2016.

[5] S. Brenner and R. Scott. *The mathematical theory of finite element methods*. Vol. 15. Springer Science & Business Media, 2007.

[6] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*. Vol. 15. Springer Science & Business Media, 2012.

[7] Franco Brezzi, Konstantin Lipnikov, and Valeria Simoncini. "A family of mimetic finite difference methods on polygonal and polyhedral meshes". In: *Mathematical Models and Methods in Applied Sciences* 15.10 (2005), pp. 1533–1551.

[8] A. Cangiani and G. Manzini. "Flux reconstruction and solution post-processing in mimetic finite difference methods". In: *Computer Methods in Applied Mechanics and Engineering* 197.9 (2008), pp. 933–945.

[9] A. Cangiani et al. "Hourglass stabilization and the virtual element method". In: *International Journal for Numerical Methods in Engineering* 102.3-4 (2015), pp. 404–436.

[10] T. K. Carne. *Geometry and groups*. 2012. URL: https://www.dpmms.cam.ac.uk/~tkc/GeometryandGroups/GeometryandGroups.pdf (visited on 06/06/2016).

[11] R. Courant, K. Friedrichs, and H. Lewy. "Über die partiellen Differenzengleichungen der mathematischen Physik". In: *Mathematische Annalen* 100.1 (1928), pp. 32–74.

[12] R. Courant et al. "Variational methods for the solution of problems of equilibrium and vibrations". In: *Bull. Amer. Math. Soc* 49.1 (1943), pp. 1–23.

[13] T. Dupont and R. Scott. “Polynomial approximation of functions in Sobolev spaces”. In: *Mathematics of Computation* 34.150 (1980), pp. 441–463.

[14] K. Eriksson et al. *Computational differential equations*. Vol. 1. Cambridge University Press, 1996.

[15] L. C. Evans. *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society, 1998. ISBN: 9780821807729.

[16] R. A. Klausen and A. F. Stephansen. “Mimetic mpfa”. In: *Proc. 11th European conference on the mathematics of oil recovery*. 2008, pp. 8–11.

[17] Ø. S. Klemetsdal. “Higher-order virtual element methods for irregular grids with application to reservoir simulation”. Master’s project.

[18] Ø. S. Klemetsdal. *MATLAB implementation of the virtual element method*. 2016. URL: https://strene@bitbucket.org/strene/the-virtual-element-method.git (visited on 06/10/2016).

[19] K.-A. Lie. *An introduction to reservoir simulation using MATLAB: User guide for the Matlab reservoir simulation toolbox (MRST), SINTEF ICT*. 2015.

[20] K.-A. Lie et al. “Open-source MATLAB implementation of consistent discretisations on complex grids”. In: *Computational Geosciences* 16.2 (2012), pp. 297–322.

[21] K. Lipnikov, M. Shashkov, and I. Yotov. “Local flux mimetic finite difference methods”. In: *Numerische Mathematik* 112.1 (2009), pp. 115–152.

[22] G. R. Liu et al. “Theoretical aspects of the smoothed finite element method (SFEM)”. In: *International journal for numerical methods in Engineering* 71.8 (2007), pp. 902–930.

[23] S. Natarajan, S. Bordas, and E. T. Ooi. “On the equivalence between the cell–based smoothed finite element method and the virtual element method”. In: *arXiv preprint arXiv:1407.1909* (2014).

[24] J. M. Nordbotten and I. Aavatsmark. “Monotonicity conditions for control volume methods on uniform parallelogram grids in homogeneous media”. In: *Computational Geosciences* 9.1 (2005), pp. 61–72.

[25] J. M. Nordbotten and G. T. Eigestad. “Discretization on quadrilateral grids with improved monotonicity properties”. In: *Journal of computational physics* 203.2 (2005), pp. 744–760.

[26] S. Operto et al. “3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study”. In: *Geophysics* 72.5 (2007), SM195–SM211.

[27] B. N. Parlett. *The symmetric eigenvalue problem*. Vol. 7. SIAM, 1980.

[28] Society of Petroleum Engineers. *SPE Reservoir Simulation Conference 2017*. 2016. URL: http://www.spe.org/events/en/2017/conference/17rsc/homepage.html (visited on 06/06/2016).

[29] P. Solin, K. Segeth, and I. Dolezel. *Higher-order finite element methods*. CRC Press, 2003.

[30] G. Strang and G. J. Fix. *An analysis of the finite element method*. Vol. 212. Prentice-Hall Englewood Cliffs, NJ, 1973.

[31] N. Sukumar and A. Tabarraei. "Conforming polygonal finite elements". In: *International Journal for Numerical Methods in Engineering* 61.12 (2004), pp. 2045–2066.

[32] V. Thomée. "From finite differences to finite elements: A short history of numerical analysis of partial differential equations". In: *Journal of Computational and Applied Mathematics* 128.1 (2001), pp. 1–54.

[33] L. Beirão da Veiga et al. "Basic Principles of the Virtual Element Method". In: *Mathematical Models and Methods in Applied Sciences* 23.01 (2013), pp. 199–214.

[34] L. Beirão da Veiga et al. "The hitchhiker's guide to the virtual element method". In: *Mathematical Models and Methods in Applied Sciences* 24.08 (2014), pp. 1541–1573.

[35] L. B. da Veiga, L. Lipnikov, and G. Manzini. *Mimetic Finite Difference Method for Elliptic Problems*. Vol. 11. Springer, 2014.

[36] H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education, 2007.

[37] H. Xiao and Z. Gimbutas. "A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions". In: *Computers & Mathematics with Applications* 59.2 (2010), pp. 663 –676.