

## LA-UR-16-27263

Approved for public release; distribution is unlimited.

Title: C1-continuous Virtual Element Method for Poisson-Kirchhoff plate problem

Author(s): Gyrya, Vitaliy  
Mourad, Hashem Mohamed

Intended for: Report

Issued: 2016-09-22

---

**Disclaimer:**

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

# $C^1$ -continuous Virtual Element Method for Poisson-Kirchhoff plate problem

V. Gyrya, H. Mourad

September 20, 2016

## Abstract

We present a family of  $C^1$ -continuous high-order Virtual Element Methods for Poisson-Kirchhoff plate bending problem. The convergence of the methods is tested on a variety of meshes including rectangular, quadrilateral, and meshes obtained by edge removal (i.e. highly irregular meshes). The convergence rates are presented for all of these tests.

## 1 Introduction

Plate bending is a classical problem which has received a fair amount of attention from researchers over many years. Plates are flat structures whose thickness is much smaller than their other dimensions. Plate response to lateral loading is dominated by bending modes of deformation. In fact, Kirchhoff plate theory, which is limited to *thin* plates, assumes that cross-sections remain planar and normal to the plate's mid-surface under deformation. Thus, transverse shear deformation is ignored. Accordingly, the plate's deformation is completely defined by the lateral displacement field of the plate's mid-plane.

The problem is governed by a fourth-order PDE, and as shown in the sequel, its weak formulation involves second-order derivatives of the transverse displacement. Thus,  $C^1$ -continuity of the solution must be enforced, so that the plate remains continuous and free of “kinks”, and also to ensure convergence. Due to this continuity requirement, early practitioners of the finite element method encountered difficulties when they first tried to construct elements for the plate bending problem. To circumvent these difficulties, mixed finite element methods were introduced, which include derivatives of displacement as nodal degrees of freedom. Examples of this approach include the HCT triangular element [9]. A number of non-conforming elements, where normal derivative continuity is enforced at the nodes but not along inter-element edges, were also developed and used successfully. For example, a non-conforming element based on piecewise quadratics on triangular meshes is described in

[14]. More recent approaches include the discontinuous Galerkin methods [10] and weak Galerkin finite element methods for the biharmonic equation on polytopal meshes [15].

In this paper we consider a conforming discretization of plate bending problem, which in the simplest settings can be viewed as the biharmonic equation. The method presented here belongs to the class of newly developed Virtual Element (VE) methods that first were developed for Poisson problem [1]. Since then VE methods have been developed for a number of other equations, such as Stokes equation [8], linear Elasticity [2], both in conformal and non-conformal formulations. In particular we note the development of VE methods with arbitrary regularity [4].

The development of VE methods could be viewed as a continuation of development of Mimetic Finite Difference (MFD) methods [12, 11, 5, 7]. For readers interested in MFD methods we recommend a review article [13] and the book [3]. One of the major differences between MFD and VE methods is the emphasis of the later ones on existence of underlying continuous approximation spaces. These spaces are described implicitly and in practice do not require to be resolved explicitly. Yet, existence of the underlying continuous spaces allows one to view VE method as a generalization of Finite Element methods to general polygonal and polyhedral meshes. This simplifies the convergence analysis and provides guidelines for the choice of the degrees of freedom.

The discretization presented in this paper is closely related to the discretization described in [6]. We note that our use of different degrees of freedom on the edges of polygonal elements. We also note lack of independent numerical verification of convergence results in in [6]. This paper fills in this gap and tests the robustness of the discretization for rather irregular meshes.

The rest of this paper is organized as follows. In section 2 we present the PDE formulation as well as the weak formulation of the problem. These will be used in the consecutive section 3 for the construction of VE discretization. In section 4 we present numerical examples where we evaluate convergence rates of the solutions for various polynomial order orders of the schemes on a variety of meshes. In section 5 we make several concluding remarks. Finally, in section 6 we have two appendices where we moved some auxiliary results and some discussions on practical aspects of implementation of the method.

## 2 Poisson-Kirchhoff plate theory formulations

Let  $\Omega \subset \mathbb{R}^2$  be a bounded domain in two dimensions with the boundary  $\Gamma = \partial\Omega$ . The Poisson-Kirchhoff plate PDE is given as

$$\nabla^2 : (C : \nabla^2 \phi) = f \quad \text{in } \Omega. \quad (1)$$

The boundary conditions are

$$\phi = g \quad \text{on } \Gamma, \quad (2)$$

$$\mathbf{n} \cdot \nabla \phi = h \quad \text{on } \Gamma. \quad (3)$$

The boundary value problem (1-3) has a unique solution for any boundary data  $g$  and  $h$  and any  $f$ .

The weak formulation for (1) with the boundary conditions (2-3) is obtained by multiplying (1) by a test function and integrating LHS by parts twice

$$\begin{aligned} \int_{\Omega} \nabla^2 : (C : \nabla^2 \phi) w \, d\Omega &= \int_{\Omega} (C : \nabla^2 \phi) : \nabla^2 w \, d\Omega - \\ &- \int_{\Gamma} (C : \nabla^2 \phi)(\mathbf{n} \cdot \nabla w) \, d\Gamma + \\ &+ \int_{\Gamma} [\nabla \cdot (C : \nabla^2 \phi)] \cdot \mathbf{n} \, w \, d\Gamma. \end{aligned} \quad (4)$$

This gives the following weak formulation: find  $\phi \in H^2(\Omega)$  satisfying the boundary conditions (2-2) such that for any  $w \in H^2(\Omega)$  with the boundary conditions (2-2) replaced by the homogeneous ones (i.e. replacing  $g$  and  $h$  by 0) one has

$$a_{\Omega}(\phi, w) = (f, w)_{\Omega}, \quad (5)$$

where

$$a_{\Omega}(\phi, w) = \int_{\Omega} (C : \nabla^2 \phi) : \nabla^2 w \, d\Omega, \quad (6)$$

$$(f, w)_{\Omega} = \int_{\Omega} f \, w \, d\Omega. \quad (7)$$

**Remark 1.** *Note that the boundary integrals in (4) vanish for the test functions  $w$ .*

### 3 Virtual Element discretization

Consider a tessellation of the domain  $\Omega$  into polygonal elements. Let  $E$  be a representative element. Define the approximation space  $\mathcal{V}_{h,E} = \mathcal{P}_k \oplus \mathcal{B}$ , for  $k \geq 3$ , as follows

$$\mathcal{V}_h(E) = \{ \phi_h \in C^1(E) \mid \nabla^4 \phi_h \in \mathcal{P}_{k-4}, \phi|_e \in \mathcal{P}_k(e), \mathbf{n} \cdot \nabla \phi|_e \in \mathcal{P}_{k-1}(e) \}. \quad (8)$$

Here  $e$  is any edge of the element  $E$ . The limitation  $k \geq 3$  on the order is due to the condition  $\nabla^4 \phi_h \in \mathcal{P}_{k-4}$ , although we could try and consider what happens for  $k = 2$ .

The *degrees of freedom* (DoF) can be taken as follows:

**Vertices**, 3 values:

- Values of the function  $\phi$ .
- Values of the derivative of the function, i.e.  $\nabla \phi$ .

**Edges,  $2k - 5$  values:**

- Values in  $k - 3$  points of the function  $\phi$ .
- Values in  $k - 2$  points of the normal derivative of the function, i.e.  $\mathbf{n} \cdot \nabla \phi$ .

**Face/element,  $(k - 2)(k - 3)/2$  values:**

- Moments of order  $k - 4$  inside the element scaled with the area of the element, i.e. values

$$\frac{1}{|E|} \int_E \phi m_i dE \quad \text{for } m_i \in \mathcal{M}_{k-4}(E),$$

where  $\mathcal{M}_n(E)$  is a set of scaled monomials

$$\mathcal{M}_n(E) = \left\{ \left( \frac{x}{h_E} \right)^{p_x} \left( \frac{y}{h_E} \right)^{p_y} \right\}_{p_x + p_y \leq n}, \quad p_x, p_y \geq 0$$

and  $h_E$  is the diameter of the element  $E$ .

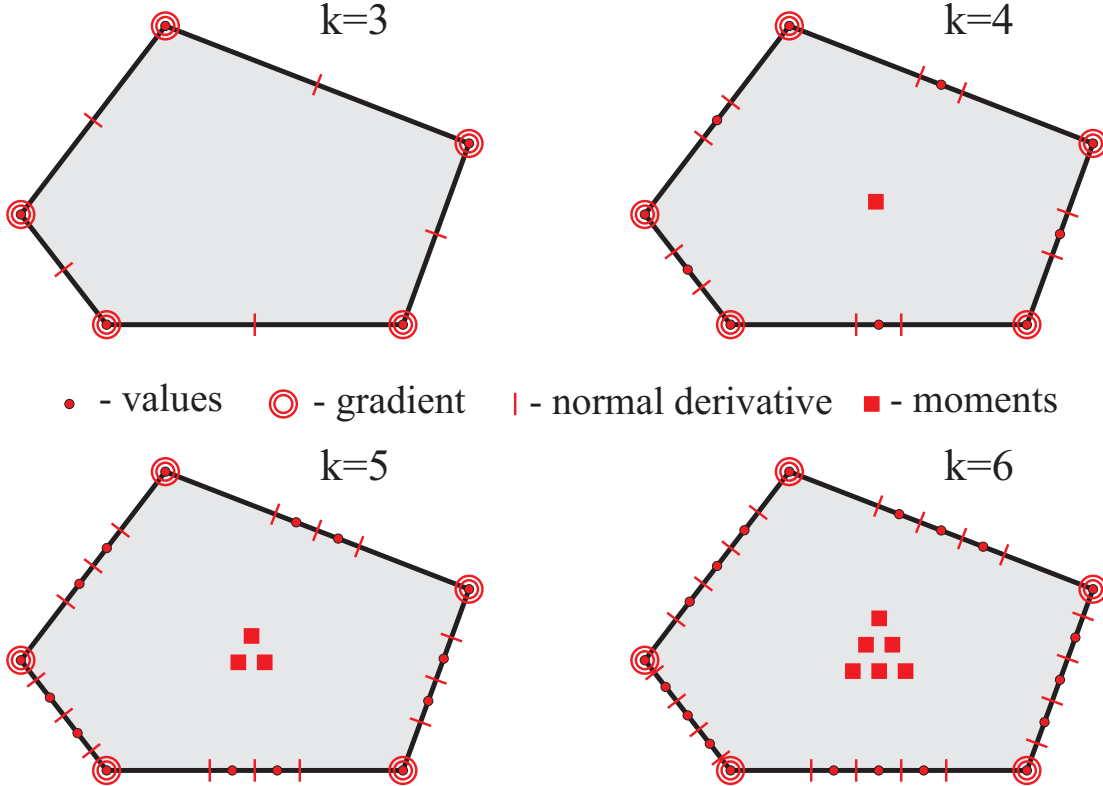


Figure 1: Illustration of DoF for order  $k = 3, \dots, 6$ .

**Theorem 1.** *The space  $\mathcal{V}_h(E)$  defined in (8) is unisolvent with respect to the above DoF, i.e. there is a one-to-one correspondence between the functions in  $\mathcal{V}_h(E)$  and the set of DoF.*

*Proof.* We will prove the theorem by

- first, showing that the dimension of the space  $\mathcal{V}_h(E)$  matches the number of DoF for an element  $E$ , and
- second, showing that if for a function  $\phi_h \in \mathcal{V}_h(E)$  all DoF are zero, then the function itself is zero.

The dimension of the space  $\mathcal{V}_h(E)$  is

$$\begin{aligned} \dim(\mathcal{V}_h(E)) &= \dim(\mathcal{P}_{k-4}(E)) + n_e (\dim(\mathcal{P}_k(e)) + \dim(\mathcal{P}_{k-1}(e))) - 3n_v = \\ &= \frac{(k-2)(k-3)}{2} + n_e((k+1) + k) - 3n_v = \\ &= \frac{(k-2)(k-3)}{2} + n_e(2k-2), \end{aligned} \quad (9)$$

where  $n_e = n_v$  is the number of edges/vertices in the element  $E$ . We subtracted  $3n_v$  because at every vertex we need to enforce three continuity conditions: one for the value and two for the derivative.

The number of DoF per element is

$$\frac{(k-2)(k-3)}{2} + n_e(2k-5) + n_v 3 = \frac{(k-2)(k-3)}{2} + n_v(2k-2). \quad (10)$$

Therefore, the dimension of the space  $\mathcal{V}_h(E)$  and the space of independent DoF for each element  $E$  are the same.

Suppose never-the-less the space  $\mathcal{V}_h(E)$  is not unisolvent with respect to the DoF. This would mean that there is non-zero  $\phi_h \in \mathcal{V}_h(E)$  whose DoF are all zero. Consider integration by parts for the energy integral

$$\begin{aligned} \int_E (\nabla^2 \phi_h)^2 dE &= \int_E (\nabla^4 \phi_h) \phi_h dE - \\ &\quad - \sum_e \int_e \mathbf{n}_e \cdot \nabla (\nabla^2 \phi_h) \phi_h de + \sum_e \int_e (\nabla^2 \phi_h) \mathbf{n}_e \cdot \nabla \phi_h de = 0. \end{aligned} \quad (11)$$

We get zero because each of the integrals is zero.

The first integral,  $\int_E (\nabla^4 \phi_h) \phi_h dE$ , is zero because by definition of the space  $\mathcal{V}_h(E)$  we have  $(\nabla^4 \phi_h) \in \mathcal{P}_{k-4}(E)$  and integrals of  $\phi_h$  with respect polynomials  $\mathcal{P}_{k-4}(E)$  over the element  $E$  are zero, as they are given by element's DoF.

The second integral is zero because  $\phi_h|_e \in \mathcal{P}_k(e)$  and since all DoF are zero, hence,  $\phi_h|_e \equiv 0$ . Similarly, the third integral is zero because  $\mathbf{n}_e \cdot \nabla \phi_h|_e \in \mathcal{P}_{k-1}(e)$  and since all DoF are zero, hence,  $\mathbf{n}_e \cdot \nabla \phi_h|_e \equiv 0$ .

Since  $(\nabla^2 \phi_h)^2 \geq 0$ , the condition (11) implies that  $(\nabla^2 \phi_h) \equiv 0$  almost everywhere (in the sense of measure). Since, based on the DoF  $\phi_h \equiv 0 \equiv \frac{\partial}{\partial n} \phi_h$  on the boundary  $\partial E$ , this implies that  $\phi_h \equiv 0$  everywhere inside the element  $E$ . This completes the proof of unisolvency.  $\square$

### 3.1 Computability/consistency condition

The computability/consistency condition is the condition on the ability to compute the bilinear form when one of the arguments is a polynomial. Rewrite the equation (4) for integration by parts in the opposite order

$$\begin{aligned} \int_E (C : \nabla^2 m_i) \nabla^2 w \, dE &= \int_E \nabla^2 : (C : \nabla^2 m_i) w \, dE + \\ &+ \sum_{e \in \partial E} \int_e (C : \nabla^2 m_i) \nabla_{\mathbf{n}} w \, de - \\ &- \sum_{e \in \partial E} \int_{e \in \partial E} [\nabla_{\mathbf{n}} (C : \nabla^2 m_i)] w \, de, \end{aligned} \quad (12)$$

where  $m_i \in \mathcal{M}_k(E)$  is a monomial. Then we have the following:

- The *first integral* on the right is computable for any  $w \in \mathcal{V}_h(E)$  in terms of face/element DoF of  $w$  since  $\nabla^2 : (C : \nabla^2 m_i) \in \mathcal{P}_{k-4}(E)$ .
- The *second* and the *third integrals* are computable for any  $w \in \mathcal{V}_h(E)$  because  $(\mathbf{n} \cdot \nabla w)$  and  $w$  can be uniquely reconstructed on the boundary in terms of the boundary DoF.

Computability of (12) for any monomial  $m_i \in \mathcal{M}_k(E)$  means that we can write a condition on the stiffness matrix  $\mathbb{A}_E$  as follows

$$M_i^T \mathbb{A}_E W = \int_{\Omega} (C : \nabla^2 m_i) \nabla^2 w \, d\Omega = R_i^T W. \quad (13)$$

Here the column vectors  $M_i$  and  $W$  consist of DoF of  $m_i \in \mathcal{M}_k(E)$  and  $w \in \mathcal{V}_h(E)$ , respectively. The column vector  $R_i$  is a vector of coefficients with DoF of  $w$  in the RHS of (12) that, as indicated above, is computable for every  $m_i \in \mathcal{M}_k(E)$ .

Hereafter, we will assume that whenever we are working with a function  $b \in \mathcal{V}_h(E)$  that  $B$  stands for its representation in terms of DoF.

The relation (13) defines an orthogonality within the approximation space

$$\mathcal{V}_h(E) = \mathcal{P}_k(E) \oplus \mathcal{B}(E). \quad (14)$$

Here  $\mathcal{B}(E)$  consists of all functions  $b \in \mathcal{V}_h(E)$  such that

- $P^T \mathbb{A}_E B = 0$  for all  $p \in \mathcal{P}_k(E)$



- $P^T B = 0$  for all  $p \in \mathcal{P}_k(E)$ , s.t.  $(C : \nabla^2 p) \equiv 0$ .

**Remark 2.** The second condition is added to define  $\mathcal{B}$  uniquely and to guaranty there are no overlaps between  $\mathcal{P}_k(E)$  and  $\mathcal{B}(E)$ .

Define a projection operator  $\Pi_k : \mathcal{V}_h(E) \rightarrow \mathcal{P}_k(E)$  as a linear map that

- preserves polynomials, i.e.  $\Pi_k[p] = p$  for any  $p \in \mathcal{P}_k(E)$ ;
- vanishes on  $\mathcal{B}(E)$ , i.e.  $\Pi_k[b] = 0$  for any  $b \in \mathcal{B}(E)$ .

This definition of  $\Pi_k$  is unique since it is a linear map and  $\mathcal{P}_k(E)$  and  $\mathcal{B}(E)$  are linearly independent subspaces that together give the whole space  $\mathcal{V}_h(E)$ .

Note the following property of the projection,  $\Pi_k$ :

$$\int_E (C : \nabla^2 p) \nabla^2 w \, dE = \int_E (C : \nabla^2 p) \nabla^2 \Pi_k[w] \, dE, \quad (15)$$

for all  $p \in \mathcal{P}_k(E)$  and for all  $w \in \mathcal{V}_h(E)$ . As a consequence we have

$$\begin{aligned} \int_E (C : \nabla^2 u) \nabla^2 w \, dE &= \int_E (C : \nabla^2 \Pi_k[u]) \nabla^2 \Pi_k[w] \, dE + \\ &+ \int_E (C : \nabla^2 (I - \Pi_k)[u]) \nabla^2 (I - \Pi_k)[w] \, dE \end{aligned} \quad \text{for all } u, w \in \mathcal{V}_h(E). \quad (16)$$

Choose a basis in the subspace  $\mathcal{B}(E)$ . Since we do not want to compute the shape of functions in  $\mathcal{B}(E)$  we will define the basis only in terms of their DoF. Suppose this basis is  $B_i$ . Then we can write the stiffness matrix  $\mathbb{A}_E$  using change of basis as follows

$$\mathbb{A}_E = [\mathbb{M} \mid \mathbb{B}]^{-T} \begin{bmatrix} \tilde{\mathbb{A}}_{\mathcal{P}\mathcal{P}} & o \\ 0 & \tilde{\mathbb{A}}_{\mathcal{B}\mathcal{B}} \end{bmatrix} [\mathbb{M} \mid \mathbb{B}]^{-1}. \quad (17)$$

Here  $\mathbb{M} = \{M_i\}$  is a matrix whose columns are DoF of monomials;  $\mathbb{B} = \{B_i\}$  is the matrix whose columns are DoF of the basis of  $\mathcal{B}(E)$ ;  $\tilde{\mathbb{A}}_{\mathcal{P}\mathcal{P}}$  is a computable block corresponding to products of polynomials with polynomials; and, finally,  $\tilde{\mathbb{A}}_{\mathcal{B}\mathcal{B}}$  is a non-computable block corresponding to the products within the subspace  $\mathcal{B}(E)$ .

The block  $\tilde{\mathbb{A}}_{\mathcal{B}\mathcal{B}}$  does not affect the order of the scheme and can be selected relatively arbitrarily. Assuming that columns of the matrices  $\mathbb{M}$  and  $\mathbb{B}$  are scaled similarly  $\tilde{\mathbb{A}}_{\mathcal{B}\mathcal{B}}$  should scale like  $\tilde{\mathbb{A}}_{\mathcal{P}\mathcal{P}}$ , i.e.

$$\text{eig}(\tilde{\mathbb{A}}_{\mathcal{B}\mathcal{B}}) \sim \max \left\{ \text{eig}(\tilde{\mathbb{A}}_{\mathcal{P}\mathcal{P}}) \right\}. \quad (18)$$

**Remark 3.** We take maximum in the RHS of (18) since  $\tilde{\mathbb{A}}_{\mathcal{P}\mathcal{P}}$  has some eigenvalues that are zero. These eigenvalues correspond to constants and linear functions, i.e. rigid body motions.

In practice, we take

$$\tilde{\mathbb{A}}_{\mathcal{B}\mathcal{B}} = \mu \text{eig}(\tilde{\mathbb{A}}_{\mathcal{P}\mathcal{P}}) \mathbb{I}, \quad (19)$$

where  $\mathbb{I}$  is an identity matrix of an appropriate size and  $\mu$  is a parameter.

### 3.2 Scaling

To this point we presented the DoF from the point of the information they contain. We did not carefully consider the scaling. Enforcing proper scaling is essential for the condition number of the global system. In this section we consider how the bilinear form and the DoF change under scaling of the element and, ultimately, choose the appropriate scaling.

This section is structured as follows. We first assume the naive choice of scaling for the DoF that was presented in Section 3. We will discover that for this scaling the condition number of the local matrix (defined as a ratio of largest to smallest nonzero eigenvalues) grows at least as  $O(h^{-1})$  for small  $h$ . We then propose an alternative (a “good”) choice of scaling for which the condition number of the local matrix is  $O(1)$  in  $h$ .

Consider a bilinear form

$$a^E(u, v) := \int_E (C : \nabla^2 u) \nabla^2 v \, dE \quad (20)$$

defined on a generic element  $E$ . Consider two elements  $E_1$  – of unit area,  $|E_1| = 1$ , and  $E_h = hE_1$  obtained by rescaling of  $E_1$ . Take two pairs of functions  $(u_1, v_1)$  and  $(u_h, v_h)$  related to one another as follows

$$\begin{cases} u_h(\mathbf{x}_h) = u_1(\frac{\mathbf{x}_h}{h}) = u_1(\mathbf{x}_1) \\ v_h(\mathbf{x}_h) = v_1(\frac{\mathbf{x}_h}{h}) = v_1(\mathbf{x}_1), \end{cases} \quad \text{where } \mathbf{x}_h = h\mathbf{x}_1.$$

One may think of  $(u_h, v_h)$  as representations of the same shape as  $(u_1, v_1)$  but on a rescaled domain  $E_h$ .

The scaling relationship between the bilinear forms  $a^{E_1}(u_1, v_1)$  and  $a^{E_h}(u_h, v_h)$  is as follows

$$\begin{aligned} a^{E_h}(u_h, v_h) &= \iint_{E_h} (C : \nabla_{\mathbf{x}_h}^2 u_h)(\nabla_{\mathbf{x}_h}^2 v_h) \, dx_h \, dy_h = \\ &= \iint_{E_1} (h^{-2} C : \nabla_{\mathbf{x}_1}^2 u_1)(h^{-2} \nabla_{\mathbf{x}_1}^2 v_1) \, d(hx_1) \, d(hy_1) = h^{-2} a^{E_1}(u_1, v_1). \end{aligned} \quad (21)$$

Here we used the fact that  $\nabla_{\mathbf{x}_h} = \nabla_{h\mathbf{x}_1} = h^{-1} \nabla_{\mathbf{x}_1}$ .

Let  $U_1$  and  $V_1$  be DoF representation of the functions  $u_1$  and  $v_1$  on the element  $E_1$  and let  $U_h$  and  $V_h$  be DoF representation of the functions  $u_h$  and  $v_h$ , respectively, on the element  $E_h$ . Define a diagonal scaling matrix  $\mathbb{S}_h$  corresponding to the scaling of the DoF described in Section 3, i.e. it has 1 for entries corresponding to moments inside the element, 1 for entries corresponding to values of the function, and  $h^{-1}$  for values of derivatives. We have then have the following relation between  $(U_1, U_h)$  and  $(V_1, V_h)$

$$U_h = \mathbb{S}_h U_1 \quad \text{and} \quad V_h = \mathbb{S}_h V_1.$$

1. The DoF corresponding to moments inside the element are independent of scaling  $h$ .

2. The DoF corresponding to function values (values at the vertices and along the edges) also do not depend on  $h$ .
3. The DoF corresponding to values of derivatives (on edges and at vertices) scale as  $h^{-1}$ .

From (21) the stiffness matrices  $\mathbb{A}^{E_h}$  and  $\mathbb{A}^{E_1}$  satisfy the following scaling relation

$$\begin{aligned} U_h^T \mathbb{A}^{E_h} V_h &= h^{-2} U_1^T \mathbb{A}^{E_1} V_1, \\ U_h^T \mathbb{A}^{E_h} V_h &= (\mathbb{S}_h U_1)^T \mathbb{A}^{E_h} (\mathbb{S}_h V_1) = U_1^T (\mathbb{S}_h^T \mathbb{A}^{E_h} \mathbb{S}_h) V_1. \end{aligned} \quad (22)$$

Since (22) holds for any  $u$  and  $v$ , we have

$$\mathbb{A}^{E_1} = h^2 \mathbb{S}_h^T \mathbb{A}^{E_h} \mathbb{S}_h \quad \text{or in opposite direction} \quad \mathbb{A}^{E_h} = h^{-2} \mathbb{S}_h^{-T} \mathbb{A}^{E_1} \mathbb{S}_h^{-1}. \quad (23)$$

Note that  $\mathbb{S}_h$  is a diagonal matrix, so transpose in (23) can be dropped, i.e.  $\mathbb{S}_h^T = \mathbb{S}_h$ .

**Remark 4.** *Note that the relation (23) is bad for the condition number of the global problem. Indeed, multiplication by matrix  $\mathbb{S}_h$  scales some rows and columns of the “reference” matrix  $\mathbb{A}^E$  as  $h$  and others like 1. For small  $h$  this will make some eigenvalues scale like 1 and others like  $h$  or even  $h^2$ . Ideally, we wish to select such DoF that the discrete energy matrices  $\mathbb{A}^E$  and  $\mathbb{A}^{\tilde{E}}$  satisfy a simple relation  $\mathbb{A}^{\tilde{E}} = h^{-2} \mathbb{A}^E$ .*

Next we explain how to modify the definitions of the DoF to satisfy the relation  $\mathbb{A}^{\tilde{E}} = h^{-2} \mathbb{A}^E$ . The idea is to incorporate the scaling  $\mathbb{S}_h$  into the definition of the DoF rather than in the energy matrices  $\mathbb{A}^{\tilde{E}}$  and  $\mathbb{A}^E$ . This in particular means that we want to scale the DoF associated with derivative at vertices and on the edges with  $h$ , which is a parameter of the element.

For the DoF associated with derivatives on edges scaling can be done with edge length. For the DoF associated with derivatives at vertices scaling would have to involve some length parameter and there is no length associated with vertices. Therefore, this would require modifying the definition of the vertex DOF in a non-local way. For instance scaling with the length of the smallest of the adjacent edges. This part of the construction of the local stiffness matrix relies on the information from adjacent elements. To summarize, we will use the following scaling for the derivative DoF:

- For *edges*, value of the derivative times the length,  $h_e$ , of the edge  $e$ , i.e.  $h_e \nabla_{\mathbf{n}} u$  and  $h_e \nabla_{\tau} u$ .
- For *vertices*, value of the derivative times the length,  $h_v$ , of the smallest adjacent edge, i.e.  $h_v \nabla u$ .

We propose the following construction of the global matrix.

**Step 1:** We *first* construct the local matrix  $\mathbb{A}^E$ , for a “reference” element  $E$  with unit area, i.e.  $|E| = 1$ . During this step we assume the DoF to be scaled like in Section 3. Therefore, no knowledge of the global mesh is needed. We only need to know information about the element  $E$ .

**Step 2:** During the *second* step we determine scaling coefficients for every edge and every vertex. This step requires analysis of the mesh beyond one element in order to compute the scaling for vertices (based on all adjacent edges).

**Step 3:** During the *third* step we compute local matrices using the new scaling  $\mathbb{A}^{\tilde{E}} \approx h^{-2}(\mathbb{S}_h \mathbb{S}_{h_*}^{-1})^T \mathbb{A}^E (\mathbb{S}_h \mathbb{S}_{h_*}^{-1})$ . Here  $\mathbb{S}_h$  is the diagonal scaling matrix that multiplies DoF associated with derivatives by  $h$  and  $\mathbb{S}_{h_*}$  is the new diagonal scaling that multiplies DoF associated with derivatives at edges by  $h_e$  and derivatives at vertices by  $h_v$ . Thus, rows and columns of  $\mathbb{A}^E$  have to be multiplied by  $\frac{h}{h_e}$  and  $\frac{h}{h_v}$ , respectively. Here, the factor of  $h$  is due to scaling of the element and the factors of  $h_e^{-1}$  and  $h_v^{-1}$  are due to rescaling of the DoF.

**Step 4:** During the *fourth* step we assemble the local matrices  $\mathbb{A}^{\tilde{E}}$  into a global one  $\mathbb{A}$ .

### 3.3 Approximation of the forcing term

We approximate the forcing term for  $k > 3$  works as follows.

$$(f, w)_\Omega \approx \sum_E \Pi_{k-4}^E \{f\} w \, dE, \quad (24)$$

where  $\Pi_{k-4}^E \{f\}$  is an  $L^2$ -orthogonal projection of  $f$  onto  $\mathcal{P}_{k-4}(E)$  – polynomials of order up to  $k-4$ . This approximation is computable for any test function  $w$  as it can be written only in terms of the internal DoF of  $w$ .

For  $k = 3$  the approximation (24) has to be modified since in this case  $\Pi_{k-4}^E \{f\} \equiv 0$ . We propose the following modification

$$(f, w)_\Omega \approx \sum_E \text{mean}^E(w) \int_E f \, dE, \quad (25)$$

where  $\text{mean}^E(w)$  is an estimate of the average of  $w$  over the element  $E$  based only on the DoF of  $w$ . A particular example of such estimate is

$$\text{mean}^E(w) = \frac{1}{n_v^E} \sum_{v \in E} w(v), \quad (26)$$

where  $n_v^E$  is the number of vertices  $v$  of the element  $E$ .

A more sophisticated example of  $\text{mean}^E(w)$  could use extrapolation of  $w$  inside triangle

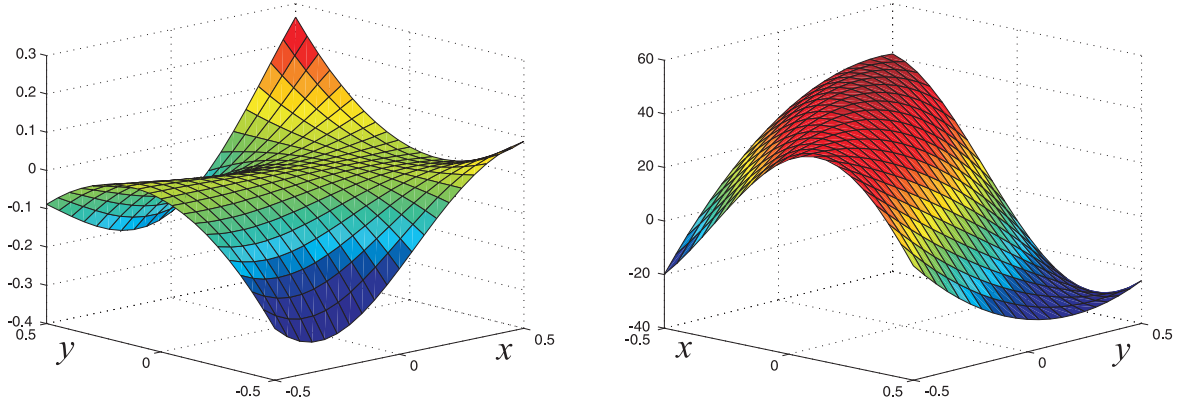


Figure 2: Illustration of the exact non-polynomial solution (left) and the corresponding forcing function (right). Note that, for ease of perception, the axis for the forcing function plot were rotated (in a clockwise direction when viewing from the top) compared to the exact solution axis.

## 4 Numerical results

The numerical results presented in this section are arranged according to the complexity of the problem starting with the simpler ones. This is also the order in which we tested our code.

We considered a problem on a square  $\Omega = [-\frac{1}{2}, \frac{1}{2}]^2$

$$\Delta^2 u(x, y) = f(x, y), \quad \text{subject to } u = u_{\text{exact}} \text{ on } \partial\Omega \quad (27)$$

with Dirichlet boundary conditions on the boundary computed for the exact solution.

We use the discrete energy norm as the measure of the error:

$$\mathcal{E}_{\text{energy}}^2 = (U - U_{\text{exact}})^T \mathbb{A} (U - U_{\text{exact}}). \quad (28)$$

In section 4.1 we test that the numerical scheme satisfies a so-called patch test. That is if an exact solution is a polynomial within our approximation space then the discretization recovers it exactly.

In sections 4.2, 4.3, and 4.4 we consider problems with non-polynomial exact solution on on square meshes, on quadrilateral meshed obtained by perturbation of square ones, and on meshes obtained by removing some edges from perturbed square meshes, respectively. We took the non-polynomial exact solution to be

$$u_{\text{exact}}(x, y) = \sin\left(\pi\left(x - \frac{1}{4}\right)\right) y\left(y + \frac{1}{2}x\right), \quad (29)$$

which implies that the forcing function has to be

$$f(x, y) = -2\pi^3 \cos\left(\pi\left(x - \frac{1}{4}\right)\right) y + \pi^2 \sin\left(\pi\left(x - \frac{1}{4}\right)\right) \left(\pi^2 y\left(y + \frac{1}{2}x\right) - 4\right). \quad (30)$$

## 4.1 Patch test

For each order of the method  $k = 3, \dots, 6$  we considered exacts solution  $\Delta^2 p = f$ , where  $p \in \mathcal{P}^k(\Omega)$  is a polynomial of order up to  $k$  on the domain  $\Omega$ . We performed the test with every monomial  $p$  for each of the orders of the method. The boundary conditions were obtained as the DoF for the exact solution.

For all the tests the numerical discretization recovered the exact solution within machine precision. The error in the energy norm was in the range  $(10^{-10}, 10^{-14})$  on all meshes with the larger errors corresponding to the finer meshes.

## 4.2 Square meshes

In this section we present the errors of the numerical solution in the energy norm when the solution is computed on square meshes. For some methods computation on square meshes leads to an addition error cancellation which manifests itself through super-convergence. As can be seen from the table below and the comparison of the errors on figures 3 and fig:rates perturbed for square and perturbed square meshes, respectively, the only significant difference in the convergence rate is for  $k = 3$ .

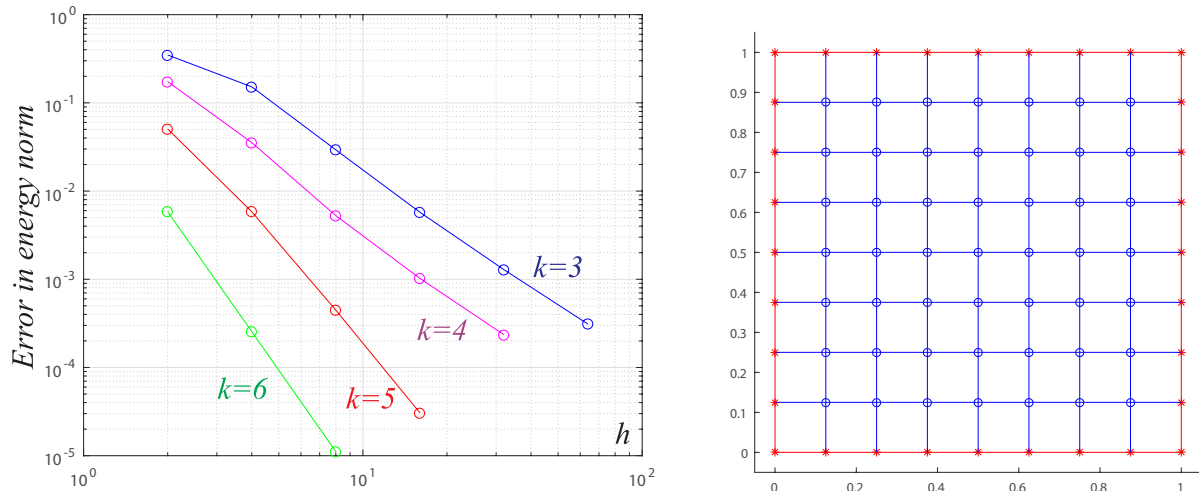


Figure 3: Convergence rate (left) for  $k = 3, 4, 5, 6$  on square meshes. On the right we show a sample mesh to be consistent with the following sections.

	$h = 2^{-1}$	$h = 2^{-2}$	$h = 2^{-3}$	$h = 2^{-4}$	$h = 2^{-5}$	$h = 2^{-6}$
$k = 3$	$3.49 \times 10^{-1}$	$1.52 \times 10^{-1}$	$2.93 \times 10^{-2}$	$5.75 \times 10^{-3}$	$1.28 \times 10^{-3}$	$3.10 \times 10^{-4}$
$k = 4$	$1.74 \times 10^{-1}$	$3.55 \times 10^{-2}$	$5.27 \times 10^{-3}$	$1.02 \times 10^{-3}$	$2.35 \times 10^{-4}$	—
$k = 5$	$5.06 \times 10^{-2}$	$5.83 \times 10^{-3}$	$4.46 \times 10^{-4}$	$3.06 \times 10^{-5}$	—	—
$k = 6$	$5.84 \times 10^{-3}$	$2.55 \times 10^{-4}$	$1.11 \times 10^{-5}$	—	—	—

The convergence rates was computed according the following formula

$$\text{Convergence rate} := \frac{\log_2(\mathcal{E}_{\text{energy},N}/\mathcal{E}_{\text{energy},1})}{\log_2(h_N/h_1)}. \quad (31)$$

The estimated convergence rates in the energy norm were as follows:

Polynomial order of the method	$k = 3$	$k = 4$	$k = 5$	$k = 6$
Estimated convergence rate	2.03	2.38	3.56	4.52

### 4.3 Perturbed square meshes

The errors in the numerical solution in the energy norm are shown in fig. 4 and in the table below. The convergence rates are not much different from those for perturbed square meshes.

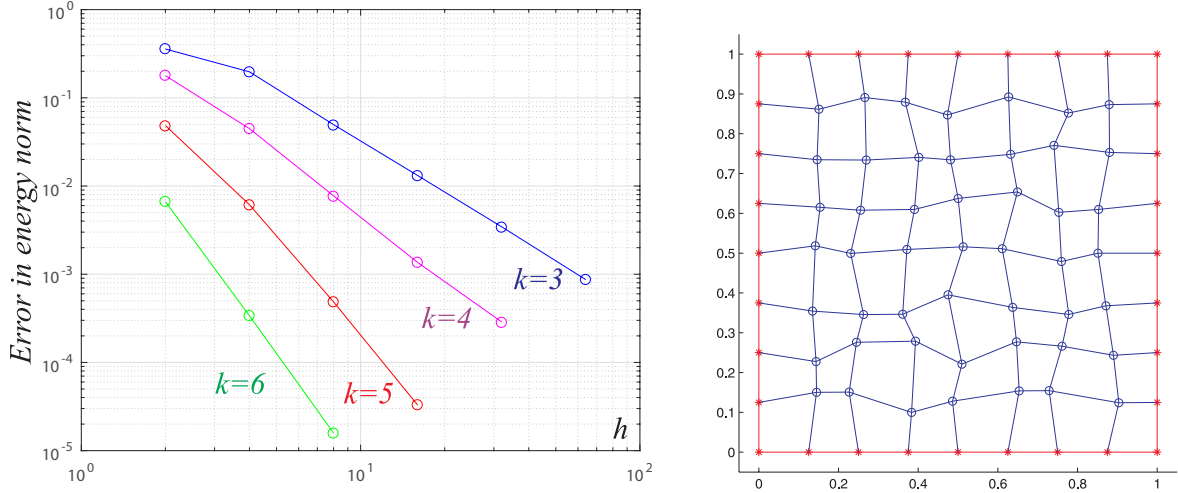


Figure 4: Convergence rate (left) for  $k = 3, 4, 5, 6$  on square meshes with randomly perturbed vertex locations computed as averages over ten randomly built mesh sequences. On the right we show a sample of the perturbed mesh used in calculations.

	$h = 2^{-1}$	$h = 2^{-2}$	$h = 2^{-3}$	$h = 2^{-4}$	$h = 2^{-5}$	$h = 2^{-6}$
$k = 3$	$3.58 \times 10^{-1}$ $\pm 4.91 \times 10^{-2}$	$1.97 \times 10^{-1}$ $\pm 2.18 \times 10^{-2}$	$4.97 \times 10^{-2}$ $\pm 1.61 \times 10^{-3}$	$1.32 \times 10^{-2}$ $\pm 2.65 \times 10^{-4}$	$3.44 \times 10^{-3}$ $\pm 3.94 \times 10^{-5}$	$8.68 \times 10^{-4}$ $\pm 7.14 \times 10^{-6}$
$k = 4$	$1.78 \times 10^{-1}$ $\pm 2.12 \times 10^{-2}$	$4.47 \times 10^{-2}$ $\pm 8.06 \times 10^{-3}$	$7.75 \times 10^{-3}$ $\pm 7.49 \times 10^{-4}$	$1.36 \times 10^{-3}$ $\pm 7.23 \times 10^{-5}$	$2.87 \times 10^{-4}$ $\pm 3.10 \times 10^{-6}$	— —
$k = 5$	$4.85 \times 10^{-2}$ $\pm 2.43 \times 10^{-3}$	$6.14 \times 10^{-3}$ $\pm 4.92 \times 10^{-4}$	$4.86 \times 10^{-4}$ $\pm 2.99 \times 10^{-5}$	$3.29 \times 10^{-5}$ $\pm 2.51 \times 10^{-7}$	— —	— —
$k = 6$	$6.64 \times 10^{-3}$ $\pm 1.07 \times 10^{-3}$	$3.38 \times 10^{-4}$ $\pm 3.88 \times 10^{-5}$	$1.59 \times 10^{-5}$ $\pm 1.72 \times 10^{-6}$	— —	— —	— —

Convergence rate on perturbed squares is computed as an average over ten randomly generated mesh sequences. The convergence rate was computed according to (31). The estimated rates in the energy norm were as follows:

Polynomial order of the method	$k = 3$	$k = 4$	$k = 5$	$k = 6$
Estimated convergence rate	1.74	2.32	3.51	4.34

#### 4.4 Further modified meshes

In this section we present results on modified meshes. As the basis we took meshes from the previous section and then removed a certain portion of internal edges joining element adjacent to this edge. To illustrate what we mean by this, compare the sample meshes in figures 4 and 5. The mesh in figure 5 was obtained from the mesh in figure 4 by removing six internal edges. Throughout the experiment we tried to maintain the number of deleted edges close to a fixed percentage of internal edges ( $\sim 5\%$ ). One should note that for large meshes this deletion process could lead to more and more complicated mesh elements. In principle for sufficiently large meshes this can even lead to elements cutting across large portions of the domain (“percolation”-type phenomena). Therefore, rate estimates in this experiment are less informative. Instead, this numerical experiment should be viewed as a stress test for the numerical scheme.

We considered the same problem as in the previous section. The average errors in energy norm were as follows:

Order \ $h$	1/2	1/4	1/8	1/16	1/32	1/64
$k = 3$	$4.09 \times 10^{-2}$	$1.18 \times 10^{-8}$	$3.50 \times 10^{-3}$	$1.04 \times 10^{-3}$	$2.93 \times 10^{-4}$	$7.93 \times 10^{-5}$
$\pm$ std	$7.57 \times 10^{-3}$	$2.08 \times 10^{-3}$	$4.65 \times 10^{-4}$	$1.12 \times 10^{-4}$	$1.53 \times 10^{-5}$	$3.04 \times 10^{-6}$
$k = 4$	$8.33 \times 10^{-3}$	$1.73 \times 10^{-3}$	$3.33 \times 10^{-4}$	$5.12 \times 10^{-5}$	$1.07 \times 10^{-5}$	—
$\pm$ std	$8.04 \times 10^{-4}$	$3.75 \times 10^{-4}$	$1.35 \times 10^{-4}$	$8.19 \times 10^{-6}$	$1.43 \times 10^{-6}$	—
$k = 5$	$7.66 \times 10^{-4}$	$6.86 \times 10^{-5}$	$7.23 \times 10^{-6}$	$4.81 \times 10^{-7}$	—	—
$\pm$ std	$1.59 \times 10^{-4}$	$1.27 \times 10^{-5}$	$2.73 \times 10^{-6}$	$8.28 \times 10^{-8}$	—	—
$k = 6$	$8.29 \times 10^{-5}$	$3.94 \times 10^{-6}$	$3.90 \times 10^{-7}$	—	—	—
$\pm$ std	$1.30 \times 10^{-5}$	$1.49 \times 10^{-6}$	$2.81 \times 10^{-7}$	—	—	—

The estimated rates in the energy norm were as follows:

Order of the method	$k = 3$	$k = 4$	$k = 5$	$k = 6$
Estimated convergence rate	1.8	2.4	3.5	3.8

Note the convergence rates have changed very little compared to the previous section.



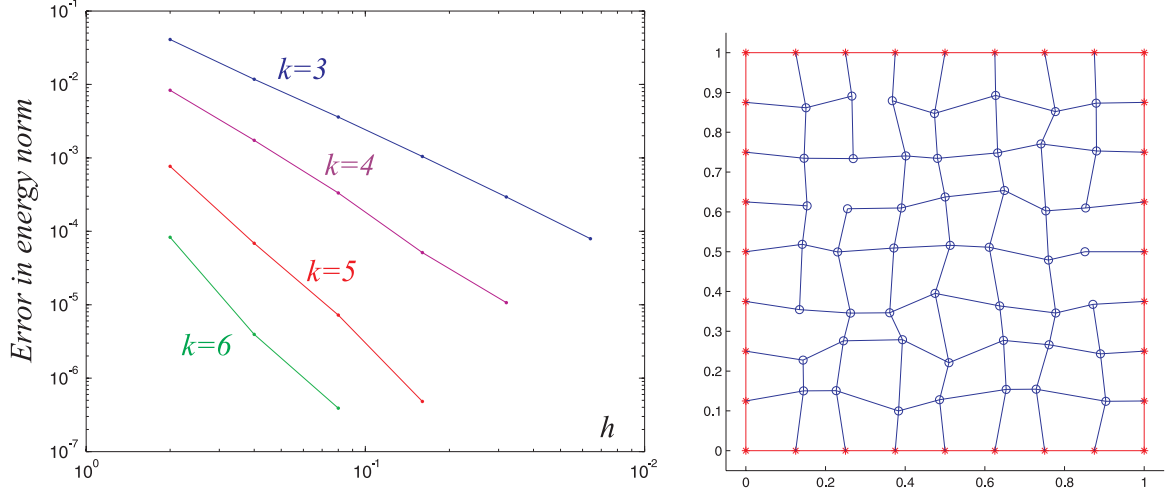


Figure 5: Convergence rate (left) for  $k = 3, 4, 5, 6$  on square meshes with randomly perturbed vertex locations computed as averages over ten randomly built mesh sequences. On the right we show a sample of the perturbed mesh used in calculations.

## 5 Conclusions

We presented virtual element methods (VEM) characterized by integer polynomial order  $k \geq 3$  for biharmonic equation in 2D. The new powerful features of these methods are (i)  $C^1$ -continuity of the approximating function and (ii) the ability of the method to work on rather general polygonal meshes, including those containing non-convex and degenerate elements. The construction of the method is the same independent of the shape of the element, number of vertices in the element or the polynomial order  $k$  (there are some differences for the construction of the RHS for  $k = 3$ ). The construction of the method in general steps is similar to that of finite element (FEM) construction since VEM is FEM-type method. The method of order polynomial order  $k$  satisfies the patch test of order  $k$  as has been demonstrated by our numerical tests.

One unexpected discovery is the apparent loss of convergence in the energy norm compared to the expect rates. Instead of the expected convergence of order  $k - 1$  we saw convergence of order  $k - 1.5$ . So far we were not able to explain this loss of convergence. Similar results have been observed for non-conformal VEM for biharmonic equation. That work has not been published yet, therefore we cannot present a reference here.

## Acknowledgements

This work was performed under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396. The authors gratefully acknowledge the support of the Laboratory Directed Research and Development (LDRD) program #20140270ER for the first author.

## 6 Appendix

### 6.1 Polynomial basis

For each polynomial order  $n \geq 2$  the dimension of the kernel of laplace differential operator for polynomials of order exactly  $n$  has dimension two. Here we list these polynomials for  $n = 2, \dots, 6$ :

$$\begin{aligned}
 n = 2 : & \quad x^2 - y^2 & \quad xy; \\
 n = 3 : & \quad x^3 - 3xy^2 & \quad 3x^2y - y^3; \\
 n = 4 : & \quad x^4 - 6x^2y^2 + y^4 & \quad x^3y - xy^3; \\
 n = 5 : & \quad x^5 - 10x^3y^2 + 5xy^4 & \quad 5x^4y - 10x^2y^3 + y^5; \\
 n = 6 : & \quad x^6 - 15x^4y^2 + 15x^2y^4 - y^6 & \quad 3x^5y - 10x^3y^3 + 3xy^5.
 \end{aligned} \tag{32}$$

### 6.2 Practical considerations

Consider a mesh consisting of polygonal elements. We will need the following data structures:

1. An array of vertices  $\{\mathbf{v}_i\}_i$  with  $x$ - and  $y$ -coordinates,  $\mathbf{v}_i = (x_i, y_i)$  of mesh vertices.
2. An array of edges  $\{\mathbf{e}_i\}_i$ ,  $\mathbf{e}_i = (e_{i,1}, e_{i,2})$ , containing indices of the vertices with the global orientation of the edge  $\mathbf{e}_i$  given by the order,  $(e_{i,1}, e_{i,2})$  vs  $(e_{i,2}, e_{i,1})$ .
3. An array of faces/elements  $\{\mathbf{f}_i\}_i$ ,  $\mathbf{f}_i = (f_{i,1}, f_{i,2}, \dots)$ , containing indices of the edges in a counter-clockwise order.

When working with a single element (also called face here) it is convenient to have a functionality that extracts only the necessary information out of the global mesh structure. This includes only those vertices and edges that are part of a given face. The functionality should include local-to-global map for the indices of the vertices, edges and the face being considered.

The DoF corresponding to one element are organized as follows:

1. First we list  $(k-2)(k-3)/2$  DoF associated with the faces.

2. Next, we list  $2k - 5$  DoF associated with the first edge. Within this block we first list  $k - 3$  DoF associated with the values of the function followed by  $k - 2$  DoF associated to the normal derivatives. Then the  $2k - 3$  DoF associated with the second, third and so on edges for a total of  $(2k - 3)n_E$  DoF.
3. Finally, we list three DoF for the first vertex, then three DoF for the second, third and so on vertices in the order they are indexed.

This organization is illustrated on Figure 6.

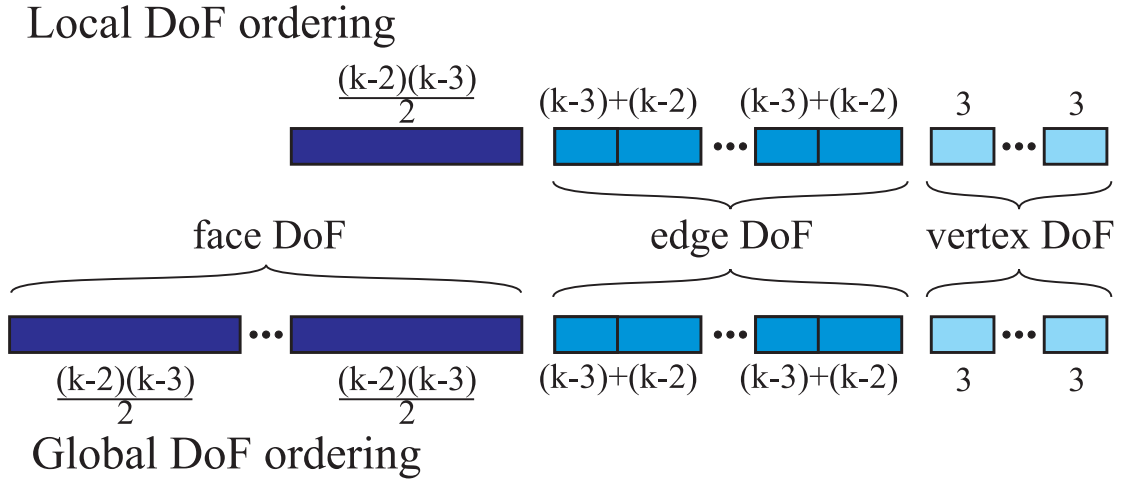


Figure 6: Illustration of ordering of DoF for one element.

## References

- [1] L. Beirão da Veiga, F. BREZZI, A. CANGIANI, G. MANZINI, L. D. MARINI, and A. RUSSO. Basic principles of virtual element methods. *Mathematical Models and Methods in Applied Sciences*, 23(01):199–214, 2013.
- [2] L. Beirão da Veiga, F. BREZZI, and L. D. MARINI. Virtual elements for linear elasticity problems. *SIAM J. NUMER. ANAL.*, 51:794–812, 2013.
- [3] L. Beirão da Veiga, K. Lipnikov, and G. Manzini. *The Mimetic Finite Difference Method for Elliptic Problems*. Springer, 2013.
- [4] L. Beirão da Veiga and G. Manzini. A virtual element method with arbitrary regularity. *IMA Journal of Numerical Analysis*, 2013.

- [5] Lourenço Beirão da Veiga, Vitaliy Gyrya, Konstantin Lipnikov, and Gianmarco Manzini. Mimetic finite difference method for the Stokes problem on polygonal meshes. *J. Comput. Phys.*, 228(19):7215–7232, 2009.
- [6] F. Brezzi and L.D. Marini. Virtual element method for plate bending problems. *Comput. Methods Appl. Mech. Engrg.*, 253:455–462, 2013.
- [7] Franco Brezzi, Konstantin Lipnikov, and Valeria Simoncini. A family of mimetic finite difference methods on polygonal and polyhedral meshes. *Math. Models Methods Appl. Sci.*, 15(10):1533–1551, 2005.
- [8] A. Cangiani, V. Gyrya, and G. Manzini. The non-conforming Virtual Element method for the Stokes equations. 2016.
- [9] Ray W Clough and James L Tocher. Finite element stiffness matrices for analysis of plates in bending. In *Proceedings of conference on matrix methods in structural analysis*, volume 1, pages 515–545, 1965.
- [10] G Engel, K Garikipati, TJR Hughes, MG Larson, L Mazzei, and RL Taylor. Continuous/discontinuous finite element approximations of fourth-order elliptic problems in structural and continuum mechanics with applications to thin beams and plates, and strain gradient elasticity. *Computer Methods in Applied Mechanics and Engineering*, 191(34):3669–3750, 2002.
- [11] Vitaliy Gyrya and Konstantin Lipnikov. High-order mimetic finite difference method for diffusion problems on polygonal meshes. *J. Comput. Phys.*, 227(20):8841–8854, 2008.
- [12] J.M. Hyman and M. Shashkov. Adjoint operators for the natural discretizations of the divergence, gradient and curl on logically rectangular grids. *Appl. Numer. Math.*, 25(4):413–442, 1997.
- [13] K. Lipnikov, G. Manzini, and M. Shashkov. Mimetic finite difference method. *J. Comp. Phys.*, 257:1163–1227, 2014.
- [14] L.S.D. Morley. The triangular equilibrium element in the solution of plate bending problems. *Aeronautical Quarterly*, 19:149–169, 1968.
- [15] Lin Mu, Junping Wang, and Xiu Ye. Weak galerkin finite element methods for the biharmonic equation on polytopal meshes. *International Journal of Numerical Analysis and Modeling*, 12:31–53, 2015.