

A gradient recovery method based on an oblique projection for the virtual element method

Balaje Kalyanaraman¹ Bishnu P. Lamichhane²
Michael H. Meylan³

(Received 25 February 2019; revised 2 October 2019)

Abstract

The virtual element method is an extension of the finite element method on polygonal meshes. The virtual element basis functions are generally unknown inside an element and suitable projections of the basis functions onto polynomial spaces are used to construct the elemental stiffness and mass matrices. We present a gradient recovery method based on an oblique projection, where the gradient of the L^2 -polynomial projection of a solution is projected onto a virtual element space. This results in a computationally efficient numerical method. We present numerical results computing the gradients on different polygonal meshes to demonstrate the flexibility of the method.

[DOI:10.21914/anziamj.v60i0.14041](https://doi.org/10.21914/anziamj.v60i0.14041) gives this article, © Austral. Mathematical Soc. 2019. Published October 12, 2019, as part of the Proceedings of the 18th Biennial Computational Techniques and Applications Conference. ISSN 1445-8810. (Print two pages per sheet of paper.) Copies of this article must not be made otherwise available on the internet; instead link directly to the DOI for this article.

Contents

1	Introduction	C202
2	Formulation	C203
2.1	Virtual element discretization	C203
2.2	Scaled monomials on the polygon	C205
3	Gradient recovery	C205
4	Results	C207
5	Conclusion	C212

1 Introduction

Gradient recovery methods are popular numerical techniques for approximating the gradient of a solution. They have the super convergence property and are used in adaptive refinement [9]. A gradient recovery method based on oblique projection was discussed by Lamichhane [3]. The method involves inverting a diagonal matrix which results in a computationally efficient method. The virtual element method [7, 8, 4] is an extension of the finite element method on polygonal meshes and offers numerous flexibilities in approximating the geometry of the domain and in adaptive algorithms. In this article, we extend the gradient recovery method for virtual element solutions obtained on polygonal meshes.

Section 2 provides a brief background on the virtual element method which is used to construct the gradient recovery operator. The procedure for computing the projection is standard and was described in detail by Beirão da Veiga et al. [8]. Section 3 defines the gradient recovery operator Q_h and discusses the construction of an appropriate linear system which is solved to obtain the gradient recovery. Finally, Section 4 presents some numerical results for

the gradient recovery method on various polygonal meshes. Throughout this article we consider the modified virtual element method discussed by Ahmad et al. [1].

2 Formulation

2.1 Virtual element discretization

Let \mathcal{T}_h be a sequence of non-overlapping decompositions of a polygonal domain $\Omega \subset \mathbb{R}^2$ into general polygonal elements K . For all elements $K \in \mathcal{T}_h$ we define the local space

$$\tilde{V}_K = \{v \in H^1(K) \cap C^0(\partial K) : \Delta v \in \mathbb{P}_1(K), v|_e \in \mathbb{P}_1(e) \quad \forall e \in \mathcal{E}_K\}, \quad (1)$$

where \mathbb{P}_1 denotes the space of polynomials of degree at most one and \mathcal{E}_K denotes the set of edges of the polygon K . The function $v \in \tilde{V}_K$ is uniquely determined on the polygon edges by knowing the values of the function v on the vertices of the polygon K . We denote by \mathcal{D} the set of degrees of freedom of the functions in \tilde{V}_K where the degrees of freedom are defined as the values of the function v on the vertices of K .

We define the projection operator $\Pi_K^\nabla : \tilde{V}_K \rightarrow \mathbb{P}_1(K)$ by

$$(\nabla \Pi_K^\nabla v, \nabla p)_K = (\nabla v, \nabla p)_K \quad \text{and} \quad P_0(\Pi_K^\nabla v) = P_0(v), \quad (2)$$

for all $p \in \mathbb{P}_1(K)$ and

$$P_0(v) = \frac{1}{N_v} \sum_{m=1}^{N_v} v(V_i),$$

where V_i denotes the i th vertex of the polygon and N_v is the number of vertices in the polygon. The operator Π_K^∇ is well defined on the local virtual element space \tilde{V}_K and is computed using the degrees of freedom \mathcal{D} .

The standard L^2 -projection operator $\Pi_K^0 : W_K \rightarrow \mathbb{P}_1(K)$ is defined as

$$(\Pi_K^0 \mathbf{v}, \mathbf{p})_K = (\mathbf{v}, \mathbf{p})_K \quad \text{for all } \mathbf{p} \in \mathbb{P}_1(K), \quad (3)$$

where the local virtual element space is defined as

$$W_K = \{ \mathbf{v} \in \tilde{V}_K : (\mathbf{v}, \mathbf{p})_K = (\Pi_K^\nabla \mathbf{v}, \mathbf{p})_K \quad \forall \mathbf{p} \in \mathbb{P}_1(K) \}, \quad (4)$$

for all $K \in \mathcal{T}_h$. The existence of the local virtual element space W_K was discussed by Ahmad et al. [1]. Since $\mathbf{v}|_e \in \mathbb{P}_1(e)$, we call W_K the linear virtual element space. However, the function \mathbf{v} may not be linear inside the polygon. Since the function $\mathbf{v} \in W_K$ is unknown in the interior of the polygon, the inner-product is computed using the extra condition

$$(\mathbf{v}, \mathbf{p})_K = (\Pi_K^\nabla \mathbf{v}, \mathbf{p})_K. \quad (5)$$

Moreover, from (3) and (5) we observe that for the linear case considered here, the projection $\Pi_K^0 \mathbf{v} = \Pi_K^\nabla \mathbf{v}$ [8, p.1561]. Thus, the global discrete virtual element space is obtained by assembling the local spaces and is defined as

$$V_h = \{ \mathbf{v} \in H^1(\Omega) : \mathbf{v}|_K \in W_K \quad \forall K \in \mathcal{T}_h \}. \quad (6)$$

Let $V_h = \text{span}\{\varphi_1, \varphi_2, \dots, \varphi_N\}$ be the set of basis functions for the virtual element space V_h . Then any function $\mathbf{u}_h \in V_h$ can be expressed as

$$\mathbf{u}_h(\mathbf{x}) = \sum_{i=1}^N \mathbf{u}_h(\mathbf{x}_i) \varphi_i(\mathbf{x}),$$

where N denotes the global degrees of freedom and the basis functions φ_i satisfy the Kronecker delta property,

$$\varphi_i(\mathbf{x}_j) = \delta_{ij},$$

where $\{\mathbf{x}_j\}_{j=1}^N$ is the set of vertices on \mathcal{T}_h . Finally, we define $\text{dof}_i(\mathbf{u}_h) = \mathbf{u}_h(\mathbf{V}_i)$ as the local degree of freedom of \mathbf{u}_h at the i th vertex of a polygon.

2.2 Scaled monomials on the polygon

Let h_K be the diameter and $\mathbf{p}_K = (\mathbf{x}_K, \mathbf{y}_K)$ be the centroid of the polygon K . We define the scaled monomials of order one as the set

$$\mathcal{M}_1(K) := \left\{ m_1 := 1, m_2 := \frac{\mathbf{x} - \mathbf{x}_K}{h_K}, m_3 := \frac{\mathbf{y} - \mathbf{y}_K}{h_K} \right\}. \quad (7)$$

The scaled monomials in (7) serve as a basis for the polynomial space $\mathbb{P}_1(K)$. Since $\Pi_K^\nabla \mathbf{u}_h \in \mathbb{P}_1(K)$ for some $\mathbf{u}_h \in \mathbf{V}_h$, we write

$$\Pi_K^\nabla \mathbf{u}_h = \sum_{i=1}^{N_v} \text{dof}_i(\mathbf{u}_h) \Pi_K^\nabla \varphi_i = \sum_{i=1}^{N_v} \text{dof}_i(\mathbf{u}_h) \sum_{\alpha=1}^3 s_{i,\alpha} \mathbf{m}_\alpha, \quad (8)$$

where the coefficients $s_{i,\alpha}$ are obtained from the definition of Π_K^∇ in (2). For our case, the same expression in equation (8) holds for $\Pi_K^0 \mathbf{u}_h$. Since the projection Π_K^∇ is defined locally inside the element, the basis functions φ_i are considered in the local sense in (8). Here $\varphi_i|_e \in \mathbb{P}_1(e)$ and it satisfies the Kronecker delta property on the vertices of the polygon [8] and hence the projection $\Pi_K^\nabla \varphi_i$ can be computed inside the polygon.

3 Gradient recovery

To compute the gradient recovery, the gradient of the virtual element solution $\nabla \mathbf{u}_h$ needs to be projected onto the virtual element space \mathbf{V}_h . An oblique projection using a bi-orthogonal basis yields a diagonal matrix which is inverted to obtain the gradient recovery.

The gradient recovery operator \mathbf{Q}_h projects $\nabla \mathbf{u}_h$ by finding $\mathbf{g}_h^k = \mathbf{Q}_h (\partial \mathbf{u}_h / \partial x_k) \in \mathbf{V}_h$ for $k = 1, 2$ such that

$$\sum_K (\Pi_K^0 \mathbf{g}_h^k, \Pi_K^0 \boldsymbol{\mu}_j)_K = \sum_K \left(\frac{\partial (\Pi_K^0 \mathbf{u}_h)}{\partial x_k}, \Pi_K^0 \boldsymbol{\mu}_j \right)_K, \quad (9)$$

with $(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}, \mathbf{y})$ and where the functions $\mu_j \in \mathcal{M}_h := \text{span}\{\mu_1, \mu_2, \dots, \mu_N\}$ satisfy the bi-orthogonal relation

$$(\Pi_K^0 \varphi_i, \Pi_K^0 \mu_j)_K = c_j \delta_{ij} \quad \text{for all } K \in \mathcal{T}_h, \quad (10)$$

where the scaling factors c_j are obtained using mass lumping. Let \mathbf{D}^K denote the local diagonal matrix defined by the bi-orthogonality relation (10) and \mathbf{D} denote the global matrix arising from the finite element assembling of the local matrices \mathbf{D}^K . From the definition of \mathbf{Q}_h in (9), the gradient recovery is performed using (10) without explicitly computing the bi-orthogonal basis μ_j and $\Pi_K^0 \mu_j$. Since the L^2 -projection $\Pi_K^0 \mathbf{u}_h$ is close to the virtual element solution \mathbf{u}_h on the polygon K , $\nabla \mathbf{u}_h$ is approximated by $\nabla \Pi_K^0 \mathbf{u}_h$. Expanding in terms of the basis,

$$\mathbf{g}_h^k = \sum_{i=1}^N \mathbf{g}_h^k(\mathbf{x}_i) \varphi_i \quad \text{and} \quad \mathbf{u}_h = \sum_{i=1}^N \mathbf{u}_h(\mathbf{x}_i) \varphi_i. \quad (11)$$

Let $\mathbf{g}_i^k = \text{dof}_i(\mathbf{g}_h^k)$ and $(\mathbf{u}_h)_i = \text{dof}_i(\mathbf{u}_h)$. Equations (9), (10) and (11) yield

$$\sum_K \sum_{i=1}^{N_v} \mathbf{g}_i^k (\Pi_K^0 \varphi_i, \Pi_K^0 \mu_j)_K = \sum_K \sum_{i=1}^{N_v} (\mathbf{u}_h)_i \left(\frac{\partial (\Pi_K^0 \varphi_i)}{\partial \mathbf{x}_k}, \Pi_K^0 \mu_j \right)_K, \quad (12)$$

which in matrix form is equivalent to the system of linear equations $\mathbf{D} \bar{\mathbf{g}}^k = \bar{\mathbf{f}}^k$. The basis functions in equation (12) are equivalent to local basis functions on the virtual element K and finite element assembly must be performed to obtain the global system. The right hand side of equation (12) is constructed as follows.

On each element $K \in \mathcal{T}_h$, since $\mathcal{M}_1(K) \subset \mathbb{P}_1(K) \subset V_{h|K}$, we use the ansatz

$$\mathbf{p}(\mathbf{x}) = \sum_{i=1}^{N_v} \text{dof}_i(\mathbf{p}) \varphi_i(\mathbf{x}) \quad \text{for } \mathbf{p} \in \mathbb{P}_1(K), \quad (13)$$

where the degrees of freedom and the basis functions are considered in the local sense. Using (8), the right-hand side in equation (12) becomes,

$$\begin{aligned}
 \left(\frac{\partial(\Pi_K^0 \varphi_i)}{\partial \chi_k}, \Pi_K^0 \mu_j \right)_K &= \sum_{\alpha=1}^3 s_{i,\alpha} \left(\frac{\partial m_\alpha}{\partial \chi_k}, \Pi_K^0 \mu_j \right)_K \\
 &= \sum_{\alpha=1}^3 s_{i,\alpha} \sum_{p=1}^{N_v} \text{dof}_p \left(\frac{\partial m_\alpha}{\partial \chi_k} \right) (\varphi_p, \Pi_K^0 \mu_j)_K \\
 &= \sum_{\alpha=1}^3 s_{i,\alpha} \sum_{p=1}^{N_v} \text{dof}_p \left(\frac{\partial m_\alpha}{\partial \chi_k} \right) (\Pi_K^0 \varphi_p, \Pi_K^0 \mu_j)_K \\
 &= \sum_{\alpha=1}^3 s_{i,\alpha} \sum_{p=1}^{N_v} \text{dof}_p \left(\frac{\partial m_\alpha}{\partial \chi_k} \right) D_{pj}^K.
 \end{aligned}$$

Written in matrix-form, we have $s_{i,\alpha} = (\mathbf{S})_{\alpha i}$ and $\text{dof}_p(\partial m_\alpha / \partial \chi_k) = (\mathbf{M}_k)_{p\alpha}$. Thus we have

$$\left(\frac{\partial(\Pi_K^0 \varphi_i)}{\partial \chi_k}, \Pi_K^0 \mu_j \right)_K = [\mathbf{D}^K \mathbf{M}_k \mathbf{S}]_{ji} := [\mathbf{Q}_k^K]_{ji}.$$

Thus the recovered gradient is a solution to the system of linear equations

$$\mathbf{D} \bar{\mathbf{g}}^k = \mathbf{Q}_k \bar{\mathbf{u}}_h = \bar{\mathbf{f}}^k,$$

where $\bar{\mathbf{u}}_h$ is the solution obtained from the virtual element method and \mathbf{Q}_k is the global matrix obtained by the finite element assembly of the local matrices \mathbf{Q}_k^K .

4 Results

We provide numerical examples to solve the standard Poisson equation

$$-\Delta \mathbf{u} = \mathbf{f} \quad \text{in } \Omega, \tag{14}$$

subject to appropriate boundary conditions. The finite dimensional virtual element weak formulation for the problem is to find $\mathbf{u}_h \in \mathbf{V}_h$ such that

$$\mathbf{a}_h(\mathbf{u}_h, \mathbf{v}_h) = \mathbf{l}_h(\mathbf{v}_h) \quad \text{for all } \mathbf{v}_h \in \mathbf{V}_h, \quad (15)$$

where the construction of the approximate bilinear form \mathbf{a}_h and the linear form \mathbf{l}_h were discussed by Beirão da Veiga et al. [7] and for the special case of $\mathbf{p} = 1$ by Sutton [4, p.6]. Vacca and Beirão da Veiga [6, p.15] defined a bilinear form \mathbf{m}_h for approximating the standard L^2 inner-product. Utilizing the stability properties of the approximate bilinear forms, it is useful to define the norms

$$|\mathbf{u}_h|_{1,h} = \sqrt{\mathbf{a}_h(\mathbf{u}_h, \mathbf{u}_h)} \quad \text{and} \quad \|\mathbf{u}_h\|_{0,h} = \sqrt{\mathbf{m}_h(\mathbf{u}_h, \mathbf{u}_h)},$$

equivalent to the standard H^1 -seminorm and the L^2 -norm, respectively. The equivalent norms are explicitly computed using the local L^2 -projection Π_K^0 and the rates of convergence for the various solutions are studied with respect to these norms.

The first example we consider is solving the Poisson equation with a smooth exact solution $\mathbf{u} = \sin(\pi x) \sin(\pi y)$. We then solve the problem using the virtual element method with homogeneous Dirichlet boundary condition on two different types of polygonal meshes, shown in Figures 1 and 2. All the meshes were generated using PolyMesher [5]. The error from the standard oblique projection on the whole and interior domain are $\mathbf{E}\mathbf{u} = \|\nabla\mathbf{u} - \mathbf{Q}_h\nabla\mathbf{u}_h\|_{0,h}$ and $\mathbf{E}_I\mathbf{u} = \|\nabla\mathbf{u} - \mathbf{Q}_h\nabla\mathbf{u}_h\|_{0,h,I}$, respectively. The standard H^1 -seminorm error on the entire domain is $\mathbf{E}_E\mathbf{u} = |\mathbf{u} - \mathbf{u}_h|_{1,h}$. Convergence analysis was performed on a uniform-square mesh and on a quasi-uniform Voronoi mesh. Figure 3 shows the convergence plots for the two different types of meshes.

For the uniform-square meshes, the rate of convergence of the recovered gradient error $\mathbf{E}\mathbf{u}$ is close to 1.5, which is higher than the rate observed for $\mathbf{E}_E\mathbf{u}$. For the error $\mathbf{E}_I\mathbf{u}$ we observe that the convergence rate is close to two, indicating that the solution deteriorates near the boundary of the domain for the uniform case. For the quasi-uniform Voronoi mesh, similar

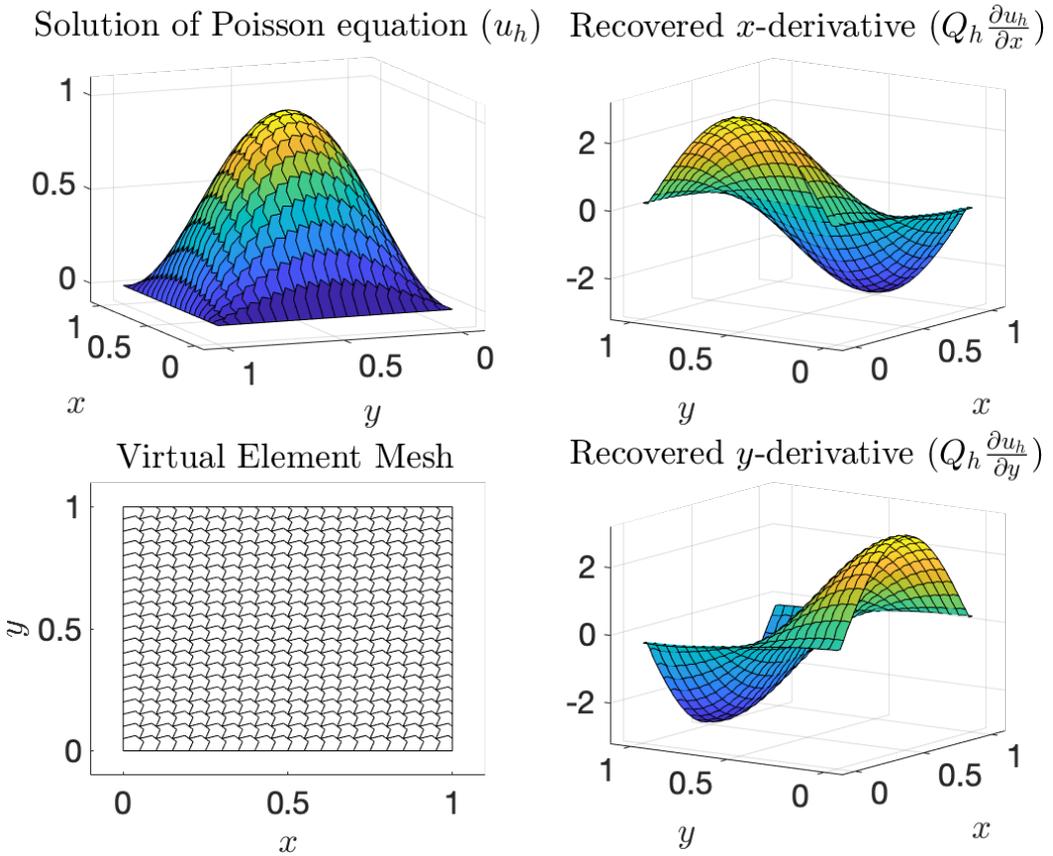
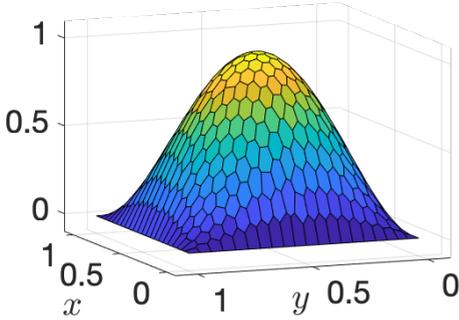
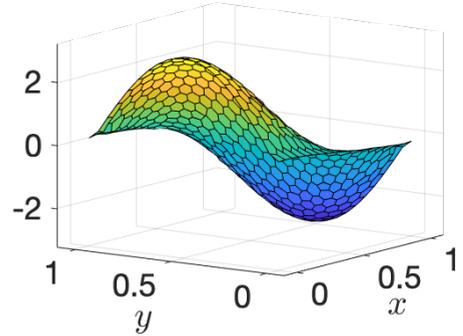


Figure 1: Solution of the Poisson equation and the recovered gradients on a virtual element mesh containing non-convex shaped elements. The error values are $|\mathbf{u} - \mathbf{u}_h|_{1,h} = 0.17631$, $\|\partial \mathbf{u} / \partial x - Q_h(\partial \mathbf{u}_h / \partial x)\|_{0,h} = 0.10382$ and $\|\partial \mathbf{u} / \partial y - Q_h(\partial \mathbf{u}_h / \partial y)\|_{0,h} = 0.10333$.

Solution of Poisson equation (u_h)Recovered x -derivative ($Q_h \frac{\partial u_h}{\partial x}$)

Virtual Element Mesh

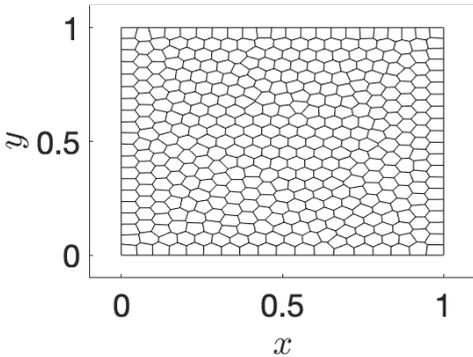
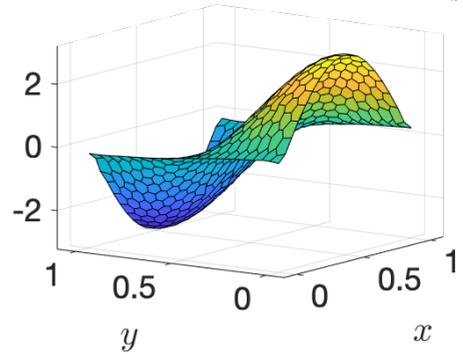
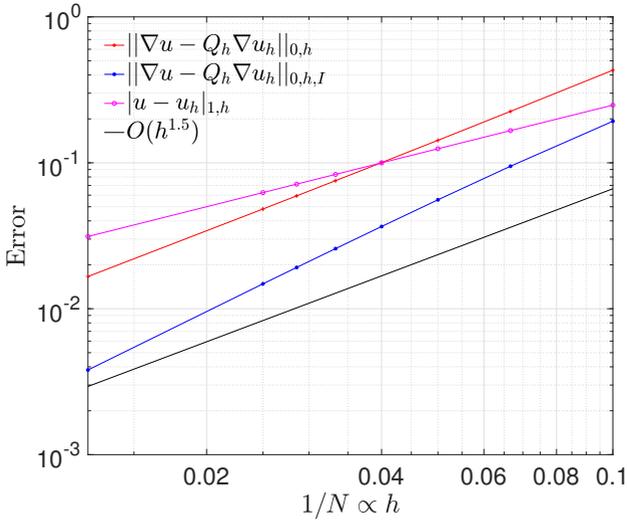
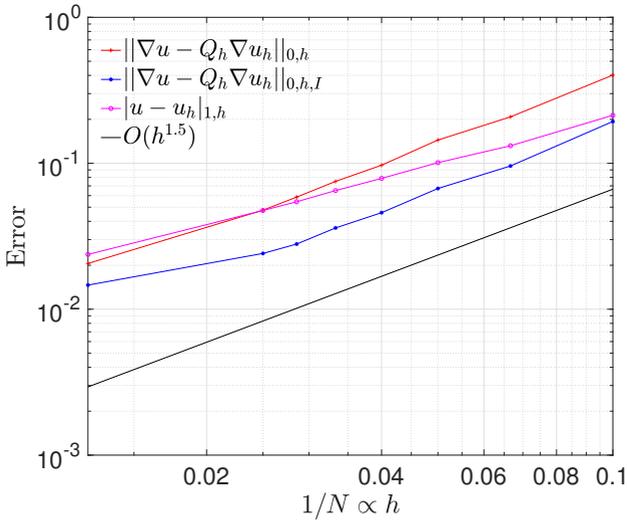
Recovered y -derivative ($Q_h \frac{\partial u_h}{\partial y}$)

Figure 2: Same as Figure 1 but on a smooth, quasi-uniform Voronoi mesh. The error values are $\|\mathbf{u} - \mathbf{u}_h\|_{1,h} = 0.10095$, $\|\partial \mathbf{u} / \partial x - Q_h(\partial \mathbf{u}_h / \partial x)\|_{0,h} = 0.10175$ and $\|\partial \mathbf{u} / \partial y - Q_h(\partial \mathbf{u}_h / \partial y)\|_{0,h} = 0.10222$.



(a) uniform-square meshes.



(b) Quasi-uniform Voronoi meshes.

Figure 3: Rate of convergence plots for gradient recovery where $E_{\mathbf{u}} = \|\nabla \mathbf{u} - Q_h \nabla \mathbf{u}_h\|_{0,h}$, $E_{I\mathbf{u}} = \|\nabla \mathbf{u} - Q_h \nabla \mathbf{u}_h\|_{0,h,I}$ and $E_{E\mathbf{u}} = |u - u_h|_{1,h}$.

trends are observed for the recovered gradient in $\mathbf{E}\mathbf{u}$. However, the error in the interior norm $\mathbf{E}_I\mathbf{u}$ converges with the same rate as $\mathbf{E}\mathbf{u}$. This is because the refinement is not perfectly uniform, unlike in the square meshes. In the finite element setting, Ilyas et al. [2] used an appropriate boundary modification technique to improve the rate of convergence on the boundary of the domain for uniform meshes. In both cases, as the mesh becomes finer, the error $\mathbf{E}\mathbf{u}$ becomes less than $\mathbf{E}_E\mathbf{u}$, indicating that $\mathbf{Q}_h\nabla\mathbf{u}_h$ is a better approximation to $\nabla\mathbf{u}$ than $\nabla\mathbf{u}_h$.

5 Conclusion

We have shown how to construct a gradient recovery operator based on an oblique projection for the virtual element method onto polygonal meshes. The basis functions are assumed to be linear on the edges of the polygons. We have established that the gradient recovery can be done without the computation of the bi-orthogonal basis. We have computed the rate of convergence of the gradient-recovery method and compared the same with the convergence of the standard gradient.

References

- [1] B. Ahmad, A. Alsaedi, F. Brezzi, L. D. Marini, and A. Russo. “Equivalent projectors for virtual element methods”. In: *Comput. Math. Appl.* 66.3 (2013), pp. 376–391. DOI: [10.1016/j.camwa.2013.05.015](https://doi.org/10.1016/j.camwa.2013.05.015) (cit. on pp. C203, C204).

- [2] Ilyas, M. and Lamichhane, B. P. and Meylan, M. H. “A gradient recovery method based on an oblique projection and boundary modification”. In: *Proceedings of the 18th Biennial Computational Techniques and Applications Conference, CTAC-2016*. Vol. 58. ANZIAM J. 2017, pp. C34–C45. DOI: [10.21914/anziamj.v58i0.11730](https://doi.org/10.21914/anziamj.v58i0.11730) (cit. on p. C212).
- [3] B. P. Lamichhane. “A gradient recovery operator based on an oblique projection”. In: *Electron. Trans. Numer. Anal.* 37 (2010), pp. 166–172. URL: <http://etna.mcs.kent.edu/volumes/2001-2010/vol37/abstract.php?vol=37&pages=166-172> (cit. on p. C202).
- [4] O. J. Sutton. “Virtual element methods”. PhD thesis. University of Leicester, Department of Mathematics, 2017. URL: <http://hdl.handle.net/2381/39955> (cit. on pp. C202, C208).
- [5] C. Talischi, G. H. Paulino, A. Pereira, and I. F. M. Menezes. “PolyMesher: a general-purpose mesh generator for polygonal elements written in Matlab”. In: *Struct. Multidiscip. O.* 45.3 (2012), pp. 309–328. DOI: [10.1007/s00158-011-0706-z](https://doi.org/10.1007/s00158-011-0706-z) (cit. on p. C208).
- [6] G. Vacca and L. Beirão da Veiga. “Virtual element methods for parabolic problems on polygonal meshes”. In: *Numer. Meth. Part. D. E.* 31.6 (2015), pp. 2110–2134. DOI: [10.1002/num.21982](https://doi.org/10.1002/num.21982) (cit. on p. C208).
- [7] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, and A. Russo. “Basic principles of virtual element methods”. In: *Math. Mod. Meth. Appl. Sci.* 23.01 (2013), pp. 199–214. DOI: [10.1142/S0218202512500492](https://doi.org/10.1142/S0218202512500492) (cit. on pp. C202, C208).
- [8] L. Beirão da Veiga, F. Brezzi, L. D. Marini, and A. Russo. “The hitchhiker’s guide to the virtual element method”. In: *Math. Mod. Meth. Appl. Sci.* 24.08 (2014), pp. 1541–1573. DOI: [10.1142/S021820251440003X](https://doi.org/10.1142/S021820251440003X) (cit. on pp. C202, C204, C205).
- [9] J. Xu and Z. Zhang. “Analysis of recovery type a posteriori error estimators for mildly structured grids”. In: *Math. Comput.* 73 (2004), pp. 1139–1152. DOI: [10.1090/S0025-5718-03-01600-4](https://doi.org/10.1090/S0025-5718-03-01600-4) (cit. on p. C202).

Author addresses

1. **Balaje Kalyanaraman**, School of Mathematical and Physical Sciences, University of Newcastle Callaghan, NSW 2308, Australia.
<mailto:Balaje.Kalyanaraman@uon.edu.au>
2. **Bishnu P. Lamichhane**, School of Mathematical and Physical Sciences, University of Newcastle Callaghan, NSW 2308, Australia.
<mailto:Bishnu.Lamichhane@newcastle.edu.au>
3. **Michael H. Meylan**, School of Mathematical and Physical Sciences, University of Newcastle Callaghan, NSW 2308, Australia.
<mailto:mike.meylan@newcastle.edu.au>