XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTIONAL METHODS IN ENGENEERING

Application of the Virtual Element Method for two-dimensions Poisson equations

Vinícius B. A. P. Pohl[1], Marcos Arndt[1,2]

[1]*Graduate Program in Numerical Methods in Engineering, Federal University of Parana*
81531-990, Curitiba, Parana, Brazil
vinicius.pohl@ufpr.br, arndt@ufpr.br
[2]*Graduate Program in Civil Engineering, Federal University of Parana*
81531-990, Curitiba, Parana, Brazil

Abstract. The Virtual Element Method (VEM) emerges to address the need for a wide variety of non-regular polygonal meshes within a computational domain. Widely used in mathematics, physics, and engineering, the Poisson equation — a differential equation that generalizes the Laplace equation — served as one of the initial applications for the development of the Virtual Element Method. This work proposes to evaluate the performance of a two-dimensional VEM applied to the Poisson equation across different mesh types, using linear approximations. The results obtained are compared with those from the Finite Element Method (FEM), serving as a reference. The objective of this work is to assess the accuracy and convergence behavior of the method with various mesh configurations.

Keywords: Virtual Element Method, Finite Element Method, Poisson equation

1 Introduction

The Virtual Element Method is a relatively recent numerical technique, with a limited historical trajectory when compared to the Finite Element Method (FEM), which is widely recognized in the scientific and engineering literature. The Virtual Element Method (VEM) was developed as a technique for solving partial differential equations (PDEs) using a Galerkin-type approximation, while enabling the use of meshes with complex geometries. Consequently, VEM can be regarded as a generalization of the FEM.

In the early development of the Virtual Element Method (VEM), its formulation was initially applied to the Poisson equation, as demonstrated by works such as those of Da Veiga et al. [1, 2] . This equation continues to serve as a benchmark for VEM implementations, including recent studies such as those of Moherdaui and Neto [3].

This work proposes the implementation of the Virtual Element Method (VEM) for solving the Poisson equation. Its performance is compared to the standard Finite Element Method (FEM) using a linear approximation, which serves as a benchmark for analyzing consistency and convergence. The paper begins with a theoretical review of the VEM implementation procedure and its formulation in comparison to the FEM. This is followed by a discussion on the convergence behavior and an analysis of the methods' performance across a diverse set of mesh configurations.

2 Theoretical background

2.1 The Poisson equation and its variational formulation

The Poisson equation is an elliptic partial differential equation that generalizes the Laplace equation and is widely used in physics. It appears in various fields such as electromagnetism, gravitation, fluid mechanics, and heat transfer. In a domain Ω, the equation takes the form:

$$-\nabla^2 u = f \tag{1}$$

where u is the potential and f is the source term.

In order for the equation to be suitable for both the Finite Element Method (FEM) and the Virtual Element Method (VEM), it must be expressed in its variational form. The variational form of the Poisson equation is given by: find $u \in H^1(\Omega)$ such that for all $v \in H_0^1(\Omega)$:

$$\int_\Omega \nabla u \nabla v \; d\Omega = \int_\Omega fv \; d\Omega \tag{2}$$

or

$$a(u,v) = L(v) \tag{3}$$

The bilinear form $a(u,v)$ is continuous and coercive, as it has a unique solution. The proofs can be found in the works of Enabe [4] and Savarè and Chanon [5].

2.2 Numerical implementations

The implementation of the Virtual Element Method shares a procedure similar to that of the Finite Element Method. After the variational formulation of the partial differential equation is established, both methods follow the general steps outlined in Algorithm 1.

Algorithm 1 General implementation of the Finite Element Method and Virtual Element Method

```
READ MESH DATA

for 1 to number of elements do
    OBTAIN LOCAL STIFFNESS MATRIX
    OBTAIN LOCAL LOAD VECTOR

end for
ASSEMBLE SYSTEM OF EQUATIONS
APPLY BOUNDARY CONDITIONS
SOLVE THE SYSTEM OF EQUATION
```

In the Virtual Element Method (VEM), the problem domain Ω is partitioned into a polytopal mesh composed of subdomains E, termed elements. A polytopal mesh allows for elements with arbitrary polygonal or polyhedral shapes. The diameter of the element, denoted by h_E, is defined as the maximum distance between any two vertices of the element, while its area is denoted by $|E|$.

As noted in Chen and Wen [6], while FEM employs shape functions that are explicit polynomials within each element, the VEM basis functions are defined by a harmonic extension and are not explicitly known inside the element.

Given that the VEM shape functions are not explicitly known, a key feature of the Virtual Element Method is the use of a projection operator, $\Pi_{E,k}$, as highlighted by Mengolini et al. [7]. This operator maps the local VEM shape function space, $V_h(E)$, to a polynomial space, $\mathbb{P}_k(E)$, such that $\Pi_{E,k} : V_h(E) \rightarrow \mathbb{P}_k(E)$, where k denotes the polynomial order.

Stifness matrix for the FEM

In the FEM approximation, the field variable u_h is defined within the finite element space $V_{h,FEM}(E)$ given by:

$$V_{h,FEM}(E) = \{u_h \in [H^1(E)]^2 : u_{u|E} \in [\mathcal{P}_n(E)]^2 \; \forall \; E \in \Omega\} \tag{4}$$

in the form:

$$u_h = \sum_{i=1}^{N^{dof}} \varphi_i . dof_i(u_h) \tag{5}$$

where $dof_i(u_h)$ are the degrees of freedom and φ_i are the shape functions.

By substituting eq. (4) in the variational form of the Poisson equation in eq. (2), and letting the test function $v = \varphi$, the FEM stiffness matrix is obtained by:

$$K_{ij} = \int \nabla\varphi_i \nabla\varphi_j d\Omega \quad i,j = 1..N^{dof} \tag{6}$$

For a linear (3 node) two-dimensional triangular element, Rao [8] demonstrates that when using a triangular coordinate system, shape functions can be expressed in terms of natural coordinates in the form:

$$\varphi_i = \frac{A_i}{|E|} \tag{7}$$

where A_i is the area of the triangular subregion formed by a point P internal to the element and the two vertices opposite to the vertex V_i.

Since the gradients of these linear shape functions are constant within the element, the FEM stiffness matrix K can be directly calculated by:

$$K_{ij} = |E| \nabla\varphi_i \nabla\varphi_j \tag{8}$$

Stifness matrix for the VEM

The Virtual Element Method (VEM) defines its function space, $V_{h,VEM}(E)$, in the form:

$$V_{h,VEM}(E) = \{u_h \in [H^1(K)]^2 : u_{u|e} \in \mathcal{P}_n(e) \,\forall\, e \in \partial E, \Delta u \in \mathcal{P}_{k-2}(K)\} \tag{9}$$

This definition reveals some important properties:

- The solution u_h is a polynomial of degree k on each edge of the element E;
- It's globally continuous on the boundary ∂E;
- And its Laplacian, Δu_h, is a polynomial of degree $k-2$.

While the Laplacian is the most natural choice for the Poisson equation, Da Veiga et al. [2] notes that the Laplacian operator in eq. (9) can be generalized to any second-order elliptical operator.

The degrees of freedom in the VEM are chosen to uniquely determine the solution within this space. These degrees of freedom usually include: the values of u_h at the vertices; the $k-1$ internal vertices of a $(k+1)$-point Gauss-Lobatto quadrature on each edge; and its internal moments given by:

$$\frac{1}{|E|} \int_E v_h m_\alpha, \quad \alpha = 1, ..., n_{k-2} \tag{10}$$

A central component of the VEM is the use of a Ritz projection, Π^∇, to connect the virtual space to a polynomial space. This projection is an elliptic operator for the Poisson problem, defined by two conditions given by:

$$\begin{cases} (\nabla(\Pi^\nabla u_h - u_h)), \nabla p_k)_{0,E} = 0 \quad \forall p \in \mathbb{P}_E \\ P_0(\Pi^\nabla u_h - u_h) = 0 \end{cases} \tag{11}$$

The first condition states that the gradient of the complement, $(\Pi^\nabla u_h - u_h)$, must be orthogonal to the gradient of any polynomial function p_k. Since this condition only holds up to a constant, a second operator, P_0, is required to ensure that the averages of the functions are equivalent.

The specific formulation of the second operator, P_0, depends on the order of the polynomial approximation. For the linear case, the operator is defined as the average of the function value at the element's vertices V_i:

$$P_0 u_h := \frac{1}{N^V} \sum_{i=1}^{N^V} u_h(V_i) \tag{12}$$

where N^V is the number of vertices.

To compute the stiffness matrix in VEM, a projection of the virtual solution onto a space of explicit polynomials is required. In the VEM literature, the use of scaled monomials is a common practice, as emphasized by Moherdaui and Neto [3]. These monomials, given by:

$$m_\alpha := \left(\frac{x - x_E}{h_E}\right)^\alpha \tag{13}$$

form a basis centered at the baricenter (x_E, y_E) of the element and are scaled by it's diameter h_E. The set of all scaled monomials with degree less than or equal to k, $\mathcal{M}_k(\Omega)$, form a basis for the polynomial space $\mathcal{P}_k(E)$.

As explained by Wriggers et al. [9], this formulation offers advantages in evaluating element matrices and leads to smaller conditioning values.

The Ritz projection, Π^{∇}, is defined by the orthogonality conditions given in eq. (11). As this projection maps to a polynomial space, it can be written in the scaled monomial basis, as:

$$\Pi^{\nabla} u_i = \sum_{\alpha=1}^{n_k} s_i^{\alpha} m_{\alpha} \tag{14}$$

By adopting the scaled monomials as the weight function and applying them to the orthogonality relation of the gradients from eq. (11):

$$\sum_{\alpha=1}^{n_k} s_i^{\alpha} (\nabla m_{\alpha}, \nabla m_{\beta})_{0,\Omega} = (\nabla m_{\alpha}, \nabla u_h)_{0,\Omega} \tag{15}$$

This relationship can be expressed in matrix form. The field variable u_h has the same Lagrangean identity as eq. (4). The operator Π_*^{∇} is then defined by solving for the coefficient vector s in the resulting system of equations:

$$sG = Bu_h \Rightarrow s = G^{-1} Bu_h = \Pi_*^{\nabla} u_h \ \therefore \ \Pi_*^{\nabla} = G^{-1} B \tag{16}$$

where G is the Gram matrix of the scaled monomials and B is the projection matrix.

The matrix B is computated by applying the divergence theorem to the integral. This separates the integral into a boundary term and a domain term. For a linear approximation, the domain term is null and the boundary term can be evaluated using the Gauss-Lobatto quadrature. This simplifies the matrix B to the form:

$$B = (\nabla m_{\alpha}, \nabla \varphi_i)_{0,\Omega} = -\int_{\Omega} \Delta m_{\alpha} \varphi_i + \int_{\partial\Omega} \frac{\partial m_{\alpha}}{\partial n} \varphi_i \tag{17}$$

$$B_{\beta,i} = \left\{ \begin{matrix} \frac{1}{N^V} & \cdots & \frac{1}{N^V} \\ \frac{|\hat{e_1}|}{2} n_{\hat{e_1}} \cdot \nabla m_{\beta} & \ldots & \frac{|e_{\hat{N^V}}|}{2} n_{e_{\hat{N^V}}} \cdot \nabla m_{\beta} \\ \frac{|\hat{e_1}|}{2} n_{\hat{e_1}} \cdot \nabla m_{\beta} & \ldots & \frac{|e_{\hat{N^V}}|}{2} n_{e_{\hat{N^V}}} \cdot \nabla m_{\beta} \end{matrix} \right\} \tag{18}$$

where $\hat{e_i}$ is the line segment between the vertices V_{i+1} and V_{i-1} and $n_{\hat{e_i}}$ is its normal vector.

Since the shape functions are defined as the canonical basis functions, with the property $dof_i(\varphi_j) = \delta_{i,j}$, the matrix B is simplified. The projector Π_*^{∇} operates on the polynomial basis. To transform it into the canonical base, a matrix D is constructed, where $D_{i\alpha} = dof_i(m_{\alpha})$. The final projector in the canonical basis is given by:

$$\Pi^{\nabla} = D\Pi_*^{\nabla} \tag{19}$$

In the work of Da Veiga et al. [1], the authors prove an identity that simplifies the calculation of the Gram matrix G in the form:

$$G = BD \tag{20}$$

As highlighted by Manzini [10], any virtual element function can be decomposed into a polynomial part, $\Pi_k^*(V_h(E))$, and a non-polynomial part $(I - \Pi_k^*)(V_h(E))$. This decomposition of the shape function φ_i is given by:

$$\varphi_i = \Pi^{\nabla}(\varphi_i) + (I - \Pi^{\nabla})\varphi_i \tag{21}$$

By substituting eq. (21) into the bilinear form from eq. (2), the local stiffness matrix $K_{ij,VEM}$ is composed of two distinct terms: a consistency and a stability term, in the form:

$$K_{ij,VEM} = (\nabla \Pi^{\nabla} \varphi_i, \nabla \Pi^{\nabla} \varphi_j)_{0,\Omega} + (\nabla (I - \Pi^{\nabla})\varphi_i, \nabla (I - \Pi^{\nabla})\varphi_j)_{0,\Omega} \tag{22}$$

The consistency term ensures that the method's behavior is accurate for polynomial solutions. This term can be computed using the Gram matrix G and the projector Π_*^{∇} as:

$$(\nabla \Pi^{\nabla} \varphi_i, \nabla \Pi^{\nabla} \varphi_j)_{0,\Omega} = \Pi_*^{\nabla} \tilde{G} \Pi_*^{\nabla} \tag{23}$$

where $\tilde{G}$ is the G matrix with its first row replaced by zeros.

The stability term ensures the method's well-posedness, even for highly distorted elements. Da Veiga et al. [2], suggests a simple but effective approximation for this term, given by:

$$\sum_{r=1}^{N^{dof}} dof_r((I-\Pi^{\nabla})\varphi_i))dof_r((I-\Pi^{\nabla})\varphi_j) = (I-\Pi^{\nabla})^T(I-\Pi^{\nabla}) \tag{24}$$

By combining these two terms, the final formulation for the VEM stiffness matrix is obtained by:

$$K_{ij,VEM} = \Pi_*^{\nabla}\tilde{G}\Pi_*^{\nabla} + (I-\Pi^{\nabla})^T(I-\Pi^{\nabla}) \tag{25}$$

Local force vector

The local force vector, which represents the right-hand side of the variational problem, is a crucial component for both the Finite Element and Virtual Element methods.

For the Finite Element Method, the test function v on the right side of eq. (2) assumes the value of the shape function, φ. The resulting integral for a given element E, is:

$$F_E = \int_E f(x,y)\varphi(x,y)dE \tag{26}$$

and can typically be evaluated using a simple one-point quadrature rule.

For the Virtual Element Method, a simplified approach is often used for linear cases. As presented by Sutton [11], the integral for the force vector can be approximated by a term that depends on the value of the source function at the element's centroid. This is represented by:

$$F_E^j = \frac{1}{|E|}\int_E \varphi_j^{\bar{E}}\Pi_0^E f dx \approx \frac{1}{|E|}\int_E \frac{f(x_E,y_E)}{N^V}dx = \frac{|E|}{N^V}f(x_E,y_E) \tag{27}$$

where N^V is the number of vertices of the element.

This is a common and practical simplification. A similar approach to that of Sutton [11] can be seen in the work of Chen and Wen [6], which demonstrates a consistent strategy for handling the local force vector in linear VEM implementations.

3 Methodology

This work employs a Fortran code that implements both the Virtual Element Method (VEM) and the standard Finite Element Method (FEM), each based on linear formulations. The performance of the methods is evaluated by comparing the results obtained for different mesh patterns through the computation of the L^2 error. The procedures to calculate the error can be found in the work of Cangiani et al. [12].

The computational domain consists of a unit square ($\Omega = [0,1] \times [0,1]$). Uniform meshes of triangular elements are considered for both methods, and several levels of mesh refinement are tested. The mesh patterns are built with the assistance of Gmsh and are illustrated in Fig. 1.

The problem under analysis consists of the Poisson equation with homogeneous Dirichlet boundary conditions, ($u = 0$ on $\partial\Omega$), subjected to a source term:

$$f(x,y) = 2\pi^2\sin(\pi x)\sin(\pi y) \tag{28}$$

with analytical solution given by:

$$u(x,y) = \sin(\pi x)\sin(\pi y) \tag{29}$$

All routines, assembly, and application of boundary conditions - except for the mesh generation - have been implemented by the authors for control and reproducibility of the analyses.

4 Results

This section presents the numerical results obtained from the implementation of both the Finite Element Method (FEM) and the Virtual Element Method (VEM). A convergence study was performed to assess the accuracy and behavior of the methods. Table 1 summarizes the computed L^2 errors for each method at various mesh sizes. These results are visually represented in Fig. 2, which illustrates the convergence of the solutions towards the analytical solution as the mesh is progressively refined.

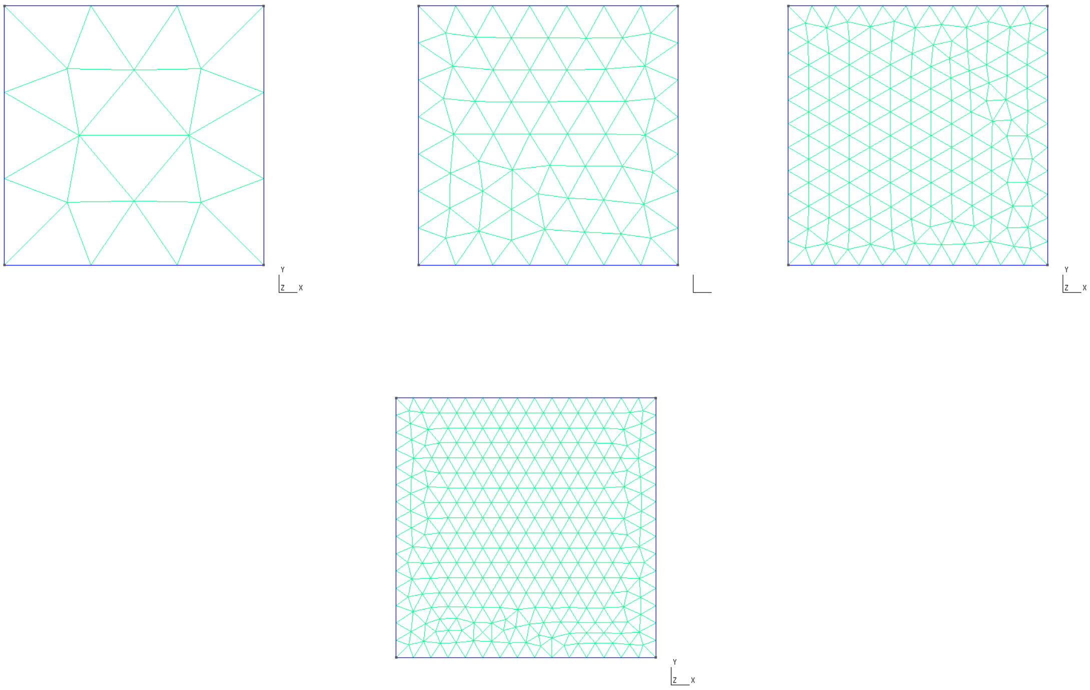
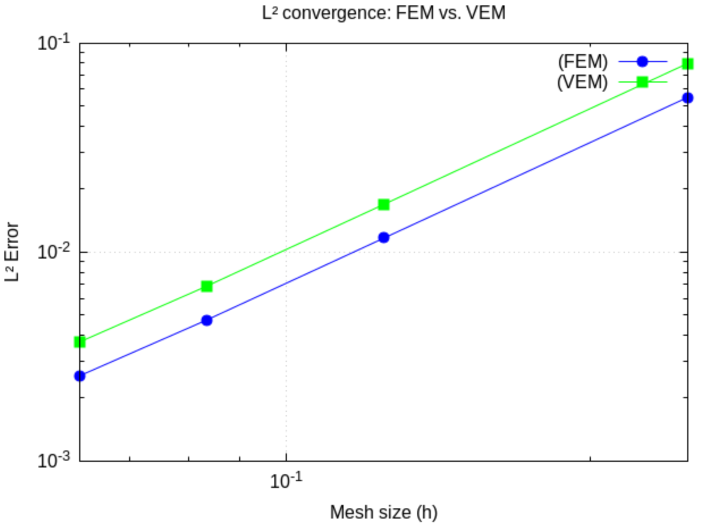


Figure 1. Meshes used in the analysis, varying from 4×4 up to 16×16 elements

Table 1. L² error

h size	FEM L² Error	VEM L² Error
$2.5000E^{-1}$	$5.4936E^{-2}$	$7.9848E^{-2}$
$1.2500E^{-1}$	$1.1662E^{-2}$	$1.6912E^{-2}$
$8.3333E^{-2}$	$4.7188E^{-3}$	$6.8514E^{-3}$
$6.2500E^{-2}$	$2.5654E^{-3}$	$3.7247E^{-3}$

Figure 2. Convergence graphic



5 Conclusions

Analyzing the results obtained by the implementation, it can be seen that both the Finite Element Method and the Virtual Element Method have a consistent convergence rate, where the decrease of the element size in the mesh results in a gradual diminishment of the error.

From Fig. 2, it can also be seen that the linear Finite Element solution has a lesser y-intercept, which shows that for this Poisson problem with triangular elements, the FEM presented a slightly better accuracy compared to the Virtual Element Method.

Acknowledgements The authors acknowledge the PPGMNE (Graduate Program in Numerical Methods in Engineering) and the UFPR (Federal University of Paraná) for their institutional support. The authors gratefully acknowledge the financial support provided by CNPq (grant number: 301802/2022-0).

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors.

References

[1] L. B. Da Veiga, F. Brezzi, L. D. Marini, and A. Russo. The Hitchhiker's Guide to the Virtual Element Method. *Mathematical Models and Methods in Applied Sciences*, vol. 24, n. 08, pp. 1541–1573, 2014.

[2] L. B. 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*, vol. 23, n. 01, pp. 199–214, 2013.

[3] T. F. Moherdaui and A. G. Neto. Virtual Element Method: An overview of formulations. *Ibero-Latin American Congress on Computational Methods in Engineering (CILAMCE)*, 2024.

[4] P. A. F. Enabe. *Virtual element method applied to the linear elastic model.* Mestrado em Engenharia de Estruturas, Universidade de São Paulo, São Paulo, 2021.

[5] S. Savarè and O. Chanon. Virtual Element Method. Master's thesis, Politecnico di Milano. Disponível em `https://github.com/deatinor/VEM3D`, 2016.

[6] L. Chen and M. Wen. Programming of linear virtual element methods. Notes. Disponível em `https://www.math.uci.edu/~chenlong/226/vemcode.pdf`, 2020.

[7] M. Mengolini, M. F. Benedetto, and A. M. Aragón. An engineering perspective to the virtual element method and its interplay with the standard finite element method. *Computer Methods in Applied Mechanics and Engineering*, vol. 350, pp. 995–1023, 2019.

[8] S. S. Rao. *The finite element method in engineering*. Butterworth-Heinemann, an imprint of Elsevier, Kidlington, Oxford, United Kingdom, sixth edition edition, 2018.

[9] P. Wriggers, F. Aldakheel, and B. Hudobivnik. *Virtual Element Methods in Engineering Sciences*. Springer International Publishing, Cham, 2024.

[10] G. Manzini. Annotations on the virtual element method for second-order elliptic problems. Technical Report LA-UR-16-29660, 1338710, 2017.

[11] O. J. Sutton. The virtual element method in 50 lines of MATLAB. *Numerical Algorithms*, vol. 75, n. 4, pp. 1141–1159, 2017.

[12] A. Cangiani, E. H. Georgoulis, T. Pryer, and O. J. Sutton. A posteriori error estimates for the virtual element method. *Numerische Mathematik*, vol. 137, n. 4, pp. 857–893, 2017.