A new quality preserving polygonal mesh refinement algorithm for Virtual Element Methods.*

Stefano Berrone\textsuperscript{a,b,*}, Alessandro D’Auria\textsuperscript{a,b}

\textsuperscript{a}Dipartimento di Scienze Matematiche, Politecnico di Torino
Corso Duca degli Abruzzi 24, Torino, 10129, Italy
\textsuperscript{b}Member of the INdAM research group GNCS

Abstract

Mesh adaptivity is a useful tool for efficient solution to partial differential equations in very complex geometries. In the present paper we discuss the use of polygonal mesh refinement in order to tackle two common issues: first, adaptively refine a provided good quality polygonal mesh preserving quality, second, improve the quality of a coarse poor quality polygonal mesh during the refinement process on very complex domains. For finite element methods and triangular meshes, convergence of a posteriori mesh refinement algorithms and optimality properties have been widely investigated, whereas convergence and optimality are still open problems for polygonal adaptive methods. In this article, we propose a new refinement method for convex cells with the aim of introducing some properties useful to tackle convergence and optimality for adaptive methods. The key issues in refining convex general polygons are: a refinement dependent only on the marked cells for refinement at each refinement step; a partial quality improvement, or, at least, a non degenerate quality of the mesh during the refinement iterations; a bound of the number of unknowns of the discrete problem with respect to the number of the cells in the mesh. Although these properties are quite common for refinement algorithms of triangular meshes, these issues are still open problems for polygonal meshes.

Keywords: Mesh adaptivity, Polygonal mesh refinement, Virtual Element Method, Convergence and Optimality

2010 MSC: 65N30, 65N50

*This research has been supported by INdAM-GNCS, by the MIUR projects PRIN 20174KLJL_004 and 20204LN5N5_003, and by the MIUR project “Dipartimenti di Eccellenza 2018-2022” (CUP E11G18000350001). Computational resources were partially provided by HPC@POLITO \texttt{http://hpc.polito.it} and SmartData@POLITO \texttt{http://smartdata.polito.it}.

*Corresponding author

Email addresses: stefano.berrone@polito.it (Stefano Berrone), alessandro.dauria@polito.it (Alessandro D’Auria)
1. Introduction

In recent years, the use of numerical methods on polygonal meshes and elements with curved edges has considerably reduced the constraints for mesh generation in domains with high geometrical complexities. The versatility and robustness of these methods has increased their use in engineering applications in the field of mechanics, thermo and fluid dynamic, and in other engineering fields. A great interest in geophysical application has been addressed to these methods, among them underground flow problems and flows in poro-fractured media, where the geometric complexities of the domain sometimes make infeasible the creation of standard triangular meshes. Moreover, many of these applications are quite critical for safety reasons and the need of reliable simulations calls for an adaptive approach. Adaptive methods have been widely investigated both from the analytical and geometric point of view for standard finite elements \[1\ 2\ 3\ 4\ 5\]. In recent years, isotropic and anisotropic estimators have been developed and analysed for polygonal methods \[6\ 7\ 8\ 9\ 10\ 11\]. The derived error estimators have many common properties with the estimators for standard finite elements, but a topic still under-investigated for polygonal methods is the refinement strategy preserving quality and with a bounded growth of unknowns. The presence in the mesh of different type of elements that can be refined independently largely increase the difficulties of obtaining a good quality mesh \[12\ 13\ 14\]. A preliminary work for general convex polygons is recently appeared \[15\].

Here we show an evolution of the algorithms presented in \[15\] that takes into account the geometric properties of the cells such that the refinement results to preserve or improve the quality of the cell. The resulting algorithms is capable to ensure some relevant issue for an optimal refinement strategy: a refinement dependent only on the marked cells at each refinement step; a common improvement of the mesh quality during the refinement iterations; a bound of the number of unknowns of the discrete problem with the number of the cells in the mesh. Moreover, the refinement algorithm has proved to automatically converge towards good quality triangular or quadrilateral meshes very suitable for the solution in the regions subject to several refinement iterations.

The article is structured in this way. In Section 3, a very brief introduction to the Virtual Element Method for 2-dimensional problems is provided with the geometrical assumptions on the polygonal elements. In Section 4 the residual estimator used for the standard Poisson problem is recalled. Virtual Element methods are introduced because the algorithm is tested in this framework, although none of the properties of these methods is assumed in
the refinement algorithm that is exclusively based on the geometrical properties of the cells. In Section 5 the geometric parameters used to check the quality of the mesh are provided. In Section 6 the new refinement algorithm based on two different splitting directions is described. In the Section 7 we show the numerical tests solving the Poisson problem in the L-shape domain starting from a set of different initial meshes. In Section 8 some other test problems concerning uniform refinement of regular polygons that easily yields to problematic configurations is presented.

2. Model Problem

In the present work we consider the numerical solution with a polygonal mesh refinement method of the simple Poisson problem with piecewise constant diffusivity coefficient $K$. Let $\Omega \subset \mathbb{R}^2$ be a bounded open set with Lipschitz boundary $\partial \Omega$; then, for a forcing term $f$ we look for a function $u$ such that

$$
\begin{cases}
-\nabla \cdot (K \nabla u) = f & \text{in } \Omega, \\
u = 0 & \text{on } \partial \Omega,
\end{cases}
$$

where $K$ is a positive function representing the diffusivity.

Let $a : H_0^1(\Omega) \times H_0^1(\Omega) \to \mathbb{R}$ be such that

$$a(w,v) := (K \nabla w, \nabla v) \quad \forall w, v \in H_0^1(\Omega),$$

where $(\cdot, \cdot)$ is the $L^2(\Omega)$ scalar product, and given $f \in L^2(\Omega)$, the variational form of (1) is: find $u \in H_0^1(\Omega)$ such that

$$a(u,v) = (f,v) \quad \forall v \in H_0^1(\Omega).$$

3. The Virtual Element Method

Let $T_\delta$ be a discretization of $\Omega \subset \mathbb{R}^2$ with open star-shaped polygons having an arbitrary, bounded, number of sides (even different from one polygon to another) and let $E_\delta$ be the set of their edges. As a mesh regularity assumption, we require that $\forall E \in T_\delta$, with diameter $H_E$, there exists a constant $\gamma > 0$ such that

- $E$ is star-shaped with respect to a ball $B_E$ of radius larger than $\gamma H_E$;
- for any two vertices $x_1, x_2 \in E$, $\|x_1 - x_2\|_{\mathbb{R}^2} \geq \gamma H_E$.

In [12] a detailed discussion on the mesh requirements and the VEM stability properties is provided. The assumptions introduced imply the existence, on each element $E \in T_\delta$, of a uniformly shape regular nested triangulation $T_{\delta,E}$ whose triangles $t$ are such that

$$\forall E \in T_\delta, \forall t \in T_{\delta,E}, \quad H_E \geq H_t \geq \gamma H_E,$$
and each of these triangles have one edge lying on \( \partial E \). This subtriangulation can be accomplished by connecting all vertices of \( E \) to the center of the ball \( B_E \), whose coordinates are \( x_E = (x_E, y_E) \), although this option does not always provide the best quality subtriangulation.

Let \( k \in \mathbb{N} \) be the polynomial order of the VEM discretization, and let \( \Pi_k^\nabla : H^1_0(\Omega) \to \mathbb{P}_k(\mathcal{T}_\delta) \) be the operator such that, \( \forall v \in H^1_0(\Omega) \) and \( \forall E \in \mathcal{T}_\delta \),

\[
(\nabla (v - \Pi_k^\nabla v), \nabla p)_E = 0, \forall p \in \mathbb{P}_k(E) \quad \text{and} \quad \begin{cases} (\Pi_k^\nabla v, 1)_{\partial E} = (v, 1)_{\partial E} & \text{if } k = 1, \\ (\Pi_k^\nabla v, 1)_E = (v, 1)_E & \text{if } k \geq 1, \end{cases}
\]

where \( \mathbb{P}_k(\omega) \) is the space of the polynomials of degree less than or equal to \( k \) on \( \omega \subset \mathbb{R}^2 \). Following [17], we introduce the finite dimensional spaces

\[
V_{\delta}^E = \{ v \in H^1(E) : \Delta v \in \mathbb{P}_k(E), v \in \mathbb{P}_k(e) \ \forall e \subset \partial E, v_{|_{\partial E}} \in C^0(\partial E) \},
\]

\[
(v, p)_E = (\Pi_k^\nabla v, p)_E \ \forall p \in \mathbb{P}_k(E)/\mathbb{P}_{k-2}(E), \quad \forall E \in \mathcal{T}_\delta,
\]

\[
V_\delta = \{ v \in C^0(\Omega) \cap H^1_0(\Omega) : v \in V_{\delta}^E \ \forall E \in \mathcal{T}_\delta \},
\]

where \( \mathbb{P}_k(\mathcal{E})/\mathbb{P}_{k-2}(\mathcal{E}) \) denotes the subspace of \( \mathbb{P}_k(\mathcal{E}) \) containing polynomials that are \( L^2(\mathcal{E}) \)-orthogonal to \( \mathbb{P}_{k-2}(\mathcal{E}) \) (see [18]; other options are possible, see, for example, [17]).

A function \( v \in V_\delta \) can be described on each polygon \( E \in \mathcal{T}_\delta \) by the following degrees of freedom:

1. the values at the vertices of the polygon;
2. if \( k \geq 2 \), for each edge \( e \subset \partial E \), the value of \( v \) at \( k - 1 \) internal points of \( e \). For practical purposes, we choose these points to be the internal Gauss – Lobatto quadrature nodes;
3. if \( k \geq 2 \), the moments \((v, m_\alpha)_E\) for all the monomials \( m_\alpha \in \mathcal{M}_{k-2}(\mathcal{E}) \) up to the order \( k - 2 \), with \( \alpha = (\alpha_1, \alpha_2), |\alpha| = \alpha_1 + \alpha_2 \leq k - 2 \), and

\[
\forall x = (x, y) \in E, \quad m_\alpha(x, y) := \frac{(x - x_E)^{\alpha_1}(y - y_E)^{\alpha_2}}{h_E^{\alpha_1+\alpha_2}}.
\]

We point out that the chosen degrees of freedom uniquely identify the polynomial expression of a function in \( V_\delta \) on each edge of the discretization, whereas inside the polygons these functions can not be directly evaluated. These degrees of freedom ensure the computability of the projection \( \Pi_k^\nabla v_\delta \), for any \( v_\delta \in V_\delta \), see [16, 18], and, once it is known, to compute the \( L^2(\Omega) \) projection of \( v_\delta \) on \( \mathbb{P}_k(\mathcal{T}_\delta) \), which is indicated by \( \Pi_k^0 v_\delta \) in the following. Similarly, \( \Pi_{k-1}^0 \nabla v_\delta \) indicates the vector containing the \( L^2(\Omega) \) projection on \( \mathbb{P}_{k-1}(\mathcal{T}_\delta) \) of the partial derivatives of \( v_\delta \), computable by the degrees of freedom.

To introduce the VEM discretization of the Poisson problem we suppose to know, for each \( E \in \mathcal{T}_\delta \), a symmetric bilinear form \( S^E : V_\delta \times V_\delta \to \mathbb{R} \) that scales like \( a^E \) on the kernel of \( \Pi_k^\nabla \), [12], i.e. \( \exists c_0, c^* > 0 \) such that

\[
\forall v_\delta \in V_\delta \text{ with } \Pi_k^\nabla v_\delta = 0, \quad c_* a^E (v_\delta, v_\delta) \leq S^E (v_\delta, v_\delta) \leq c^* a^E (v_\delta, v_\delta),
\]
where $a^E(v,w) := (K\nabla v, \nabla w)_E$. Once $S^E$ is given, we can define the following local and global discrete bilinear forms:

$$a_{\delta}^E(u_{\delta},v_{\delta}) := (K\Pi^0_{k-1}\nabla u_{\delta},\Pi^0_{k-1}\nabla v_{\delta})_E + S^E((I - \Pi^\delta_k)u_{\delta},(I - \Pi^\delta_k)v_{\delta}),$$

$$a_h(u_{\delta},v_{\delta}) := \sum_{E \in \mathcal{T}_{\delta}} a_{\delta}^E(u_{\delta},v_{\delta}).$$

With the above definitions, we can formulate the Virtual Element method as the solution to the following discrete problem: find $u_{\delta} \in V_{\delta}$ such that

$$a_h(u_{\delta},v_{\delta}) = (f_{\delta},v_{\delta}) \quad \forall v_{\delta} \in V_{\delta},$$

where $f_{\delta} := \Pi^0_k f$, that is the best approximation of $f$ that allows the computability of the scalar product with a VEM function, since $(\Pi^0_k f,v_{\delta}) = (f,\Pi^0_k v_{\delta})$ and we can compute $\Pi^0_k v_{\delta}$ using the degrees of freedom. The well-posedness of this problem simply follows by noticing that, thanks to (5), $a_h$ is coercive on $V_{\delta}$; optimal orders of convergence are proved in [18].

One possible choice for $S^E$ is the scalar product between the two vectors containing the degrees of freedom of the two functions involved, i.e., if we indicate by $\chi_r$ the operator which associates to each function in $V_{\delta}$ its $r$-th degree of freedom,

$$S^E(u_{\delta},v_{\delta}) := K_E \sum_{r=1}^{N^\text{dof}_E} \chi_r(u_{\delta})\chi_r(v_{\delta}) \quad \forall E \in \mathcal{T}_{\delta}, \forall u_{\delta},v_{\delta} \in V_{\delta},$$

where $N^\text{dof}_E$ indicates the number of degrees of freedom on element $E$.

The stabilization term can be avoided resorting to a slightly different VEM formulation [19].

4. A posteriori error estimate

The proposed refinement strategies are tested on a simple Poisson problem with constant diffusivity parameter. This simple case fits the assumptions made in [10] to prove an equivalent relation between the error $e_{\delta}^\pi = u - u_{\delta}^\pi$, where $u_{\delta}^\pi = \Pi^\pi_k u_{\delta}$, measured in the broken norm

$$|||v||| := \sup_{w \in H^1_0(\Omega)} \frac{\sum_{E \in \mathcal{T}_{\delta}} (K\nabla v, \nabla w)_E}{\sqrt{\sum_{E \in \mathcal{T}_{\delta}} (K\nabla w, \nabla w)_E}}$$

and the computable, residual-based error estimator introduced in the following. The results provided in [7, 8, 9] can be applied as well.

In order to define the error estimator mesh-size parameters are needed: let $H_E$ be the longest edge of the element $E \in \mathcal{T}_{\delta}$, and $h_e$ be the length of the edge $e \in \mathcal{E}_{\delta}$. 

5
For each $e \in E_\delta$, let $R, L \in T_\delta$ be the two polygons sharing $e$ and $N_e := \{R, L\}$ the set of the elements (right and left) sharing the edge $e$.

**Definition 1.** For any internal edge $e \in E_\delta$ let us define a unit normal vector $n_e$ as the outward unit normal vector for the element on the right of $e$ ($n_e = n_R$) and the jump of the co-normal derivative of $u^\pi_\delta$:

$$\left[ K \frac{\partial u^\pi_\delta}{\partial n} \right]_e = K \nabla u^\pi_\delta |_R \cdot n_R - K \nabla u^\pi_\delta |_L \cdot n_L = K \nabla u^\pi_\delta |_R \cdot n_e + K \nabla u^\pi_\delta |_L \cdot n_e.$$ 

Moreover, let $\eta_{R,E}$ the local residual estimator $\forall E \in T_\delta$:

$$\eta^2_{R,E} := \frac{H^2_E}{K} \| f_\delta + \nabla \cdot (K \nabla u^\pi_\delta) \|_E^2 + \frac{1}{2} \sum_{e \in E_{\partial E}} \frac{h_e}{K} \left\| \left[ K \frac{\partial u^\pi_\delta}{\partial n} \right]_e \right\|_e^2,$$

and finally, we define the error estimator for the computable projection $u^\pi_\delta$ of the VEM solution $u_\delta$:

$$\eta_R := \left\{ \sum_{E \in T_\delta} \eta^2_{R,E} \right\}^{\frac{1}{2}}. \quad (9)$$

In the following, the relation $a \lesssim b$ means that there exists a constant $C$ independent of $a, b$ such that $a \leq C b$.

**Theorem 1.** [10] Let $u$ be the solution to (2), $u^\pi_\delta = \Pi^\nabla_k u_\delta$, and $f_\delta = \Pi^0_k f$. Then,

$$|||u - u^\pi_\delta||| \lesssim \left\{ \sum_{E \in T_\delta} \left( \eta^2_R + \sum_{e \in E_{\partial E}} \frac{H^2_E}{K} \| f - f_\delta \|_E^2 \right) \right\}^{\frac{1}{2}}, \quad (10)$$

being the constant independent of the meshsize and dependent on the mesh quality and on the VEM order $k$.

**Theorem 2.** [10] Let $u$ be the solution to (2), $u^\pi_\delta = \Pi^\nabla_k u_\delta$, and $f_\delta = \Pi^0_k f$. Then,

$$\eta_R \lesssim \left\{ |||u - u^\pi_\delta|||^2 + \sum_{E \in T_\delta} (1 + H^2_E) \frac{H^2_E}{K} \| f - f_\delta \|_E^2 \right\}^{\frac{1}{2}}. \quad (11)$$

Results contained in Theorems 1 and 2 provide an equivalent relation between the error estimator $\eta_R$ and the error $|||u - u^\pi_\delta|||$ up to the approximation $\| f - f_\delta \|_E$ of the right hand side $f$, they are proved in [10] provided some assumption on the polygonal mesh.
In order to evaluate the quality, reliability and efficiency of the estimator the effectivity index e.i. is commonly used:

\[ e.i. := \frac{err}{\eta_R} \text{ where } err := \left\{ \sum_{E \in T_3} \left\| \sqrt{K} \nabla (u - u_\pi^E) \right\|_E^2 \right\}^{\frac{1}{2}}. \]  \tag{12}

Being the effectivity index strongly influenced by several aspects of mesh quality, in the following we will investigate its behaviour during the refinement process and we will analyse how the different refinement strategies considered can impact on it.

5. Definition of mesh quality parameters

In order to evaluate and check the evolution of the mesh quality during the refinement process we introduce a set of geometric parameters based on the following geometric quantities, \( \forall E \in T_3 \) and \( e \in E \), let us define:

- \( H_E \) the longest edge of the element \( E \);
- \( h_E \) the shortest edge of the element \( E \);
- \( h_e \) the length of the edge \( e \);
- \( C_E \) the centroid of the element \( E \);
- \( R_E \) the largest distance between the centroid and the vertices of \( E \);
- \( r_E \) the smallest distance between the centroid and the edges of \( E \).

The quality of the mesh elements is analysed through the following parameters:

- \( AR^{Br}_E = R_E / r_E \);
- \( AR^{Hr}_E = H_E / r_E \);
- \( AR^{Hh}_E = H_E / h_E \);
- \( AR^{edge}_E = \max_{e=1}^{N_E} \max_N \{ h_e, h_{(e+1)\%N_E} \} \min_N \{ h_e, h_{(e+1)\%N_E} \} \);
- \( K_{E}^{\nabla} = \kappa_2(\Pi_{E}^{\nabla}) \), condition number of the matrix \( \Pi_{E}^{\nabla} \in \mathbb{R}^{\dim(\mathbb{P}_k(E)) \times \#dofs_E} \).

The matrix \( \Pi_{E}^{\nabla} \) is computed using the scaled monomial basis for \( \mathbb{P}_k(E) \) introduce in Section3, this quantity is relevant only for VEM discretization.

Some of these parameters provide similar information, and we will discuss which of them can be considered significant in order to discuss the quality of the elements of a polygonal partition for a VEM discretization.
These parameters are selected because their computation does not require the solution of complex or expensive problems. For example, instead of using $R_E$ and $r_E$ the computation of the radius of the smallest circumscribed circle and of the largest inscribed circle (the ball $B_E$ for which the convex element $E$ is star shaped) should be more significant, but their computation for a generic convex polygon can be quite expensive. The quantities $R_E$ and $r_E$ can possibly replace these quantities in a cheaper way. The quality of the cells depends on the best sub-triangular mesh, that is non-trivially constructed as well.

Moreover, we consider some parameter to characterize the mesh in term of number of points, cells, and number of vertices of the polygons.

- $\#\mathcal{T}_\delta$: number of cells of the mesh $\mathcal{T}_\delta$;
- $\#P_{\mathcal{T}_\delta}$: number of points of the mesh $\mathcal{T}_\delta$;
- $\#\triangle_{\mathcal{T}_\delta}$: number of cells of $\mathcal{T}_\delta$ with three vertices;
- $\#\diamond_{\mathcal{T}_\delta}$: number of cells of $\mathcal{T}_\delta$ with four vertices;
- $E_{\#P_{\mathcal{T}_\delta}} = \#P_{\mathcal{T}_\delta}/\#\mathcal{T}_\delta$;
- $R_{\triangle_{\mathcal{T}_\delta}} = \#\triangle_{\mathcal{T}_\delta}/\#\mathcal{T}_\delta$;
- $R_{\diamond_{\mathcal{T}_\delta}} = \#\diamond_{\mathcal{T}_\delta}/\#\mathcal{T}_\delta$.

In the following sometimes we will use the previous symbols $\#\triangle_{\mathcal{T}_\delta}$, $\#\diamond_{\mathcal{T}_\delta}$, $R_{\triangle_{\mathcal{T}_\delta}}$, $R_{\diamond_{\mathcal{T}_\delta}}$ dropping the mesh symbol $\mathcal{T}_\delta$.

For the Virtual Element Methods, the parameter $E_{\#P_{\mathcal{T}_\delta}}$ can be considered as a sort of cost parameter in the sense that each cell is providing an approximation to the solution with a fixed polynomial degree, the degree of the VEM elements, at a cost that is increasing with the number of the points of the polygon. Considering this aspect, the most efficient mesh for approximating the solution with a piecewise polynomial function (the function $u_{\mathcal{T}_\delta}$ in the VEM a posteriori error estimator) is the triangular mesh displaying the minimum number of dofs per cell required by the polynomial degree $k$ and for which $E_{\#P_{\mathcal{T}_\delta}} \approx 0.5$.

6. Refinement algorithms

In this section we introduce the refinement algorithms used for refining marked convex cells generating two new convex sub-cells. In the algorithms we assume that each edge of the original mesh is provided with a marker uniquely identifying the edge, it is inherited by the children edges generated by the refining algorithm. This marker is used in order to identify aligned edges of a cell produced by the refinement of some neighbouring cell without
any additional computational cost. We do not consider aligned edges of the initial mesh present in the same element.

Triangular cells are always refined with the longest edge refinement criterion. This refining criterion is also applied when a cell is recognised as a triangle in spite of the presence of more than three vertices when aligned edges are present. For other polygons the first refinement algorithm presented is based on the idea introduced in the article [15] of using the maximum inertia moment as cutting direction of the element (Algorithm 1). The second one, is a natural extension of the longest edge refinement applied to polygons (Algorithm 2) considering the longest diagonal of the cell. This second option does not require the computation of integrals on the element but the (squared) distance between all the couple of vertices.

In the Algorithm 3 we describe how we generate an optimal cutting direction. The algorithm consists of a double loop over the edges of the cell to prevent the collapsing of the cutting line Cl on an edge of the element when a change of Cl is suitable (collapsing on a vertex).

In the Algorithm 4 we describe the procedure to refine the polygonal cells with a cut along the inertia axis with the maximum moment improved with respect to [15]. In order to prevent the creation of small edges in the refined cells a small change of the refining direction was proposed in [15] such that new edges shorter than a fraction of the cut edge were collapsed. In this new algorithm we define an adaptive collapsing tolerance that depends on the shape and the edges of the refining polygon M and its neighbouring cells N_M. For each edge e ∈ E_M we define the set of the neighbouring cells N_e. In order to preserve or improve the quality of the produced cells we define ρ_E = \min\{h_E, r_E\} for E ∈ T_δ, and then we define for each edge e ∈ E_δ

\[ \rho_e = \max\{\rho_E : E \in N_e\}. \]  

Let M_δ ⊂ T_δ the set of the marked elements for refinement, let us consider M ∈ M_δ and let e ∈ E_δ,M an edge of the element M that should be refined for refining M (the edge intersected by the cutting line Cl). The parameter \rho_e is designed in order to prevent the formation of small edges in the refinement of the edge e that could degenerate the quality of the mesh either reducing the quality of the children of M or of the neighbouring element of M faced to the edge e. The refinement algorithm is based on four main rules:

1. New edges obtained refining the edge e are created only if they are larger then \( c_p \rho_e \) for some arbitrary parameter \( c_p \geq 0 \), otherwise the intersection is collapsed to the closest vertex.
2. Regardless of the intersection point between Cl and e, the edges is cut in its mid-point if the intersection does not collapse in one of its vertices.
3. If the intersected edge is part of a set of aligned edges (descendant of an edge of the initial mesh) of the element M, the set of aligned edges
of $M$ is cut in the mid-point of the set. We remark that due to the chosen procedure this point is always a point of $M$.

4. A cell with three edges or three sets of aligned edges (a geometric triangle, regardless of the number of vertices) is cut with the Longest Diagonal rule.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{example_polygon.png}
\caption{Example of polygon producing a child with more vertices, $M = P_0 - P_6$, children $P_0P_5P_5$ and $P_5P_1P_1 - P_4$.}
\end{figure}

\begin{algorithm}
\caption{Compute Maximum Moment Direction}
\begin{algorithmic}[1]
\State Given a convex polygon $M \in \mathcal{M}_3$
\State 1: Compute the centroid $C_M$
\State 2: Compute the inertia tensor respect to $C_M$
\State 3: Compute eigenvalues and eigenvectors
\State 4: Set the cut-line $Cl$ to the line parallel to eigenvector corresponding to the maximum eigenvalue trough $C_M$
\end{algorithmic}
\end{algorithm}

\begin{algorithm}
\caption{Compute Longest Diagonal Direction}
\begin{algorithmic}[1]
\State Given a convex polygon $M \in \mathcal{M}_3$
\State 1: Compute the longest diagonal $D_M$
\State 2: Set the cut-line $Cl$ to the line orthogonal to the longest diagonal trough $C_M$
\end{algorithmic}
\end{algorithm}

Remark 1. The main differences between the proposed algorithm and the one presented in [15] is that in deciding if an edge can be cut we involve the effects of the cut on the neighbouring element sharing the edge. If the cut of an edge is suitable for the quality of the element that is performing the cut, but negative for the quality of the other facing element the cut is not performed and the intersection between the cutting line and the edge is collapsed to the closest vertex. Moreover, the cut, if performed, always occurs in the middle of the edge in this way an edge is cut always in the same way independently of the element that first processes it if the two sharing elements may cut the same edge. This simple improvement prevent some bad situations for the quality of the produced cells.
Algorithm 3 Smoothing Direction
Given a convex polygon $M \in \mathcal{M}_\delta$ and the cut-line $Cl$

1: Define the set $\mathcal{N}_M$ of the elements sharing an edge with $M$
2: Compute $\rho_E = \min\{h_E, r_E\} \ \ \forall E \in \mathcal{N}_M$
3: for Each edges $e$ of the cell do
4: Compute the intersection between $e$ and $Cl$
5: if There is an intersection then
6: Set $\rho_e = \max \rho_E \ \ E \in \mathcal{N}_e$
7: if $e$ has some neighbour aligned edges of $M$ generated by a previous refinement then
8: Find the mid-point of the ancestor largest edge $\hat{e} \supset e$ of $M$
9: else
10: Find the mid-point of $\hat{e} = e$ and set it as standard cutting point.
11: end if
12: if $|\hat{e}|/2 <= c_\rho \rho_e$ then
13: Set the closest vertex as candidate collapsed cutting point
14: end if
15: end if
16: end for
17: if We have two standard cutting points then
18: Split the cell with the segment through the two standard cutting points
19: else if We have two non-consecutive candidate collapsing points then
20: Split the cell with the segment through the two collapsed cutting points
21: else if We have a standard cutting point and a candidate collapsing point then
22: Set the cutting line $Cl$ as the line passing through the collapsed point and the centroid. Compute the intersection between $Cl$ and the edges nonadjacent to the collapsed point. Repeat points 5-15
23: else if We have two consecutive candidate collapsing points then
24: Set the cutting line $Cl$ as the line passing through the collapsed point closest to the original intersection between the two intersected edges and the centroid. Compute the intersection between $Cl$ and the edges nonadjacent to the collapsed point. Repeat points 5-15
25: end if
Figure 2: Example of refining conditioned by the neighbouring cell: bottom square $\rho_E = 0.5$, upper trapezoid $\rho_E = 0.5224$, the square is not cut by the vertical dotted red segment because this cut would produce an edge $|\hat{e}|/2 < 0.5224$ not viable for the upper cell, so the cut is collapsed on the first vertex of the edge and a new $Cl$ is set passing from this vertex and the centroid. The new $Cl$ is passing through a vertex and the new cut results in the green continuous segment. This example is valid with $c_p \geq 0.5/0.5224$.

**Algorithm 4** Refinement algorithm

Given a convex polygon $M \in \mathcal{M}_5$

1. if $M$ is a triangle then
2. Refine the cell with the longest edge refinement
3. else
4. Compute the centroid $C_M$
5. Compute the cut-line $Cl$
6. Use the Smoothing Direction Algorithm [3]
7. Cut the cells in the two children cells
8. Update the neighbourhood elements
9. end if

**Remark 2.** The presented refinement algorithm can produce new children cells with at most the number of vertices of the father cell increased by one. A child cell have more vertices than the father cell when the cut refines to consecutive edges of the father cell, see Figure [7]. In all the other cases the vertices of the children cells are less or equal to the number of vertices of the refined cell.

**Remark 3.** The absence of a conformity recovering step needed, for example, to get a conforming triangular mesh in the refinement of a triangular grid implies that the number of new cells generated by the refinement algorithm that splits each refined cell in two children cells is exactly the number of marked cells. Neighbouring cells of refined cells may change due to the refined edges shared with refined cells. In order to prevent the growing of number of vertices for non refined cells, after the refinement of all the marked cells a refinement of the un-refined cells modified with an increment of the number vertices larger than a given number or a refinement of the cells with many aligned edges can be considered. We have not implemented this.
facility and by our experiments we did not notice any problem concerning the growing of vertices per cell in the mesh.

Remark 4. Being the longest diagonal the maximum distance between two points of the cells, the Longest Diagonal criterion applied to a cell with a triangular shape independently of the number of vertices is equivalent to the longest edge refinement applied to the same cell but with glued aligned edges.

7. Numerical Results: convergence and mesh quality

In this section we analyse the performances and the overall behaviour of the refinement Algorithm applied to the solution of the Laplace problem in the $L$-shape domain $\Omega = (-1, 1)^2 \setminus (-1, 0)^2$ (see Figure 3) with exact solution

$$u(r, \theta) = r^{\frac{2}{3}} \sin\left(\frac{2}{3}(\theta + \frac{\pi}{2})\right),$$

where $r, \theta$ are the polar coordinates. The forcing function is $f = 0$, homogeneous Dirichlet boundary conditions are imposed on the edges which generate the re-entrant corner, and suitable non homogeneous Dirichlet boundary conditions obtained by the evaluation of the exact solution are set on the other edges. This is a quite common to test problem for a posteriori error estimates and mesh adaptive methods due to the bounded regularity of the solution [20].

The adaptive algorithm is based on the standard loop:

$$SOLVE \rightarrow ESTIMATE \rightarrow MARK \rightarrow REFINE.$$  \hfill (14)

The marking strategy follows from [5], Section 5, with the parameter $\theta = 0.5$ using the error estimator introduced in Section 4 and the stopping criterion

$$\frac{\eta_R^2}{\|\sqrt{K} \nabla u_h^3\|^2} \leq 1.0E^{-4},$$  \hfill (15)
Figure 4: Error Estimator $\eta_R$ versus #dofs

Figure 5: Error Estimator $\eta_R$ versus $\#T_h^k - \#T_h^0$
Figure 6: Effectivity Index e.i. versus refinement step
with a maximum number of iterations set to 100.

We consider different starting meshes and analyse the behaviour of the algorithm with VEM of order one. The notation used to define the meshes and the refinement options is:

- **TRI MM-LE**
  - Starting mesh: triangular mesh, Figure 3 (left).
  - Refinement strategy: Maximum Moment (Algorithm 1) with quality check ($c_\rho=0, 1, 1.5$).
  - Refinement strategy (FVem): Longest Edge with conformity recovery for comparison with FEM P1.

- **TRAP_MM**
  - Starting mesh: trapezoidal mesh, Figure 3 (center).
  - Refinement strategy: Maximum Moment (Algorithm 1) with quality check ($c_\rho=0, 1, 1.5$).

- **POLY MM**
  - Starting mesh: polygonal mesh, Figure 3 (right).
  - Refinement strategy: Maximum Moment (Algorithm 1) with quality check ($c_\rho=0, 1, 1.5$).

- **POLY_LD**
  - Starting mesh: polygonal, Figure 3 (right).
  - Refinement strategy: Longest Diagonal (Algorithm 2) with quality check ($c_\rho=0, 1, 1.5$).

The triangular mesh is generated by the Triangle library [21, 22], the polygonal one is generated by PolyMesher [23]. For the Maximum Moment strategy (Algorithm 1) and for the Longest Diagonal strategy (Algorithm 2) we consider three different values of the coefficient $c_\rho = 0.0, 1.0, 1.5$ (Line 12 in Algorithm 3). The case with $c_\rho = 0$ is the case in which we always cut the intersected edge in the mid-point, or, in case of the intersection with aligned edges, we cut the set of aligned edges in the mid-point of the set.

In all the cases we analyse the rate of convergence of the *a posteriori* error estimator $\eta_R$, the effectivity index $e.i.$ and the ratio between the number of mesh nodes and the number of cells in the mesh $E^#_P/\#_{T_h}$. We also analyse the quality of the mesh by observing the parameters $AR^r_E$, $AR^{edge}_E$, $AR^H_H$, $AR^{R_L}_L$, $K_E^{P}$. Finally, we report the fraction of cells that are triangle (with exactly three vertices, i.e. without aligned edges) $R^T_{T_h}$, and the fraction of cells that are quadrilateral polygons (with exactly four vertices) $R^Q_{T_h}$. 
With the test TRI\_MM-LE we compare adaptive results obtained starting from a triangular mesh using the maximum moment refinement with the results obtained by the longest edge refinement with triangular conformity recovery (case denoted by FVem in the figures), i.e. with the results obtained on a suitable mesh by conforming FEM elements P1.

In Figure 4 we report the convergence behaviour of the estimator $\eta_R$ with respect to the \#dofs. We note that the behaviour is essentially the same on the different meshes and using Maximum Moment and Longest Diagonal strategies. We remark that the reference rate of convergence $\alpha = -0.5$ is the largest possible with elements of order one. We also remark that all the collapsing strategies ($c_\rho = 0.0, 1.0, 1.5$) behaves similarly and in the case TRI\_MM-LE behaves essentially like the FEM of order one on all conforming triangular meshes. We remark that in Figure 4b the case $c_\rho = 1.0$ is overlapped to the case $c_\rho = 1.5$ (the same will happen also in the next figures concerning POLY\_MM), whereas in Figure 4c it is very close to $c_\rho = 0.0$.

In Figure 5 we report the behaviour of the estimator with respect to the added cells in the refinement process, that is also the number of cells marked for refinement. Figure 6 displays the effectivity index with respect to the refinement steps. We can observe that its behaviour is quite stable, in the case TRI\_MM-LE all the collapsing strategies produce overlapped curves, very close to the FVem case, in the other plots the different collapsing strategies may slightly impact on the values.

In Figure 7 the ratio $E/\#T_b$ between the number of mesh points and cells is reported, we can observe that this ratio is bounded and decreasing. The meshes that during refinement progressively tend to triangular meshes (see Figure 12) have this ratio approaching the value 0.5, whereas the TRAP-mesh with $c_\rho = 0$ that converge to a mesh of quadrilateral elements (see Figure 13) has a limit for this ratio that approaches 1. The boundedness of this ratio is relevant for efficiency reasons being the number of cells related to the approximation capacity of the VEM solution, whereas the number of points being related to the dimension of the linear systems to be solved. The degrowth of this ratio implies that refinement does improve approximation efficiency of the new cells.

In Figure 8 we display the behaviour of $AR_E^{\rho_r}$, in subfigure 8a-8d the maximum value of $AR_E^{\rho_r}$ for each mesh is plotted, and in Subfigures 8e and 8f the mean values over each mesh and the standard deviation are displayed. We can see that the maximum values are quite stable and during the refinement process does not increase so much being the quality of the mesh stable for all the elements.

The mean values are slightly increasing in the first iterations and then almost constant during refinement. The almost constant standard deviation and mean value suggest that, also if the maximum is increasing, the number of elements for which we have this higher values are very few and most of
Figure 8: Max $AR_{E}^{R}$: (a), (b), (c), (d), Mean $AR_{E}^{R} \pm$ standard deviation: (e), (f)
Figure 9: Max $A_{R_E^{edge}}$: (a), (b), (c), (d), Mean $A_{R_E^{edge}}$ ± standard deviation: (e), (f)
Figure 10: Max $AR^H_{E}$

Figure 11: Max $AR^H_{Eh}$

Figure 12: $R^\triangle_{Ts}$

Figure 13: $R^2_{Ts}$
Figure 14: Refinement steps POLY-MM test with $c_\rho = 0.0, 1.5$. 
Figure 15: Max $\kappa_2(\Pi^T)$: (a), (b), (c), (d), Mean $\kappa_2(\Pi^T)\pm$ standard deviation: (e), (f)
the elements of the mesh have an improved quality. In Figure 9 we display the same type of information concerning $AR_r^e$. The behaviour of this parameter is even better with respect to $AR_r^e$ and the mean values of this aspect ratio are decreasing and the standard deviation strongly decreasing denoting a very good effect of the refinement on the produced polygonal meshes. In Figures 10 and 11 we report some plot of the aspect ratios $AR_r^H$, $AR_h^H$ with a behaviour very similar to $AR_r^e$, $AR_h^e$, respectively.

In Figures 13 and 12 we compare the fraction of cells that during refinement become triangles or quadrilateral polygons for TRAP MM mesh and POLY MM and POLY LD meshes. As previously noted the TRAP mesh with $c_\rho = 0.0$ tend to produce quadrilateral cells, whereas in all the other cases the mesh is progressively and automatically transformed in a good quality triangular mesh.

In Figure 14 we display a sequence of meshes generated by the refinement algorithm. Figures 14a-f describe the behaviour of the algorithm with $c_\rho = 0.0$, whereas Figures 14g-l provide the meshes generated with $c_\rho = 1.5$. In Figure 14f we can clearly observe a large number of cells with four or three edges and that some cells have an hanging node. In Figure 14l most of the cells are triangles.

In Figure 15 we analyse the behaviour of the spectral condition number of the projector $\Pi^\nabla$. Although the maximum condition number is slightly increasing for polygonal meshes all the values are in the order of ten and can be considered stable during the refinement process. Comparing the plots of $\kappa_2(\Pi^\nabla)$ with the behaviour of $AR_r^e$ and $AR_h^e$ reported in Figures 8 and 10 respectively, we can see that $\kappa_2(\Pi^\nabla)$ is strongly related to the geometrical shape of the cell. As for the other quality parameter the mean value is almost constant as well. The condition number in the case FVem is not the identity matrix because it is the change of basis between the scaled monomial basis and the triangle Lagrangian basis.

The estimator used in the numerical experiments is equivalent to the error provided some mesh properties are assumed. In [10] these assumptions are discussed and the stability of a suitable quasi interpolation operator is numerically investigated. In [24] is proved that the VEM stabilization term, present in the estimator in [9], can be bounded by the estimator on triangular meshes with hanging nodes. In Figure 16 we report the behaviour of the ratio between the stabilization term computed with the solution and the squared estimator along the refinement iterations. The decay of this ratio confirm the negligible role of the stabilization terms in the error estimator within the proposed framework that converges to triangular or quadrilateral (including triangles with hanging nodes) meshes. In the case TRAP MM (Figure 16a) we can see a fast decay for the values of $c_\rho$ that converge to triangular meshes (see Figures 12 and 13), whereas we find a larger and stable ratio for $c_\rho = 0.0$ converging to mixed triangular-quadrilateral meshes. The same behaviour is confirmed by Figure 16b where the case $c_\rho = 1.5$ is
rapidly converging to an almost triangular mesh and the ratio $S(u_\delta, v_\delta)^2 / \eta_R^2$ is decreasing, whereas the cases $c_\rho = 0.0$ and $c_\rho = 1.0$ do not decrease due to the large number of non triangular cells. The refinement strategy based on the longest diagonal (Figure 16c) is converging towards a triangular mesh for all the values of $c_\rho$ and the ratio is decreasing for all the values.

8. Uniform refinement of a regular polygon

In this Section we analyse the behaviour of the meshes obtained by a uniform refinement of a regular polygon with 120 edges, initially not partitioned (one cell), with the strategy of the Algorithms 1 and 4. In the first refinement iterations a strong degeneration of the elements quality is unavoidable starting from a regular polygon that is cut in two cells by a diagonal, nevertheless the algorithm is capable to fix the element quality and
converge towards a good quality mesh composed by triangles and quadrilateral elements depending on the parameter $c_\rho$.

In Figure 17 we collect the behaviour of the parameters we have chosen for the characterization of the mesh quality.

Figures 17a, 17c suggest that the parameter $AR_{E}^{Rr}$ is less sensitive to the geometric degeneration of the elements in the first step with respect to $AR_{E}^{edge}$, Figures 17b, 17d. The strong jump in $AR_{E}^{edge}$ in the first iteration is due to the presence of a long edge (the diameter of the polygon) close to one of the 120 edges of the polygon. The behaviour of the parameters $AR_{E}^{Hr}$ is similar to the one of the parameter $AR_{E}^{Rr}$, and $AR_{E}^{HH}$ is similar to $AR_{E}^{edge}$.

In Figure 18 (left) we report the fraction of cells that have three vertices and in Figure 18 (right) the fraction of cells with four vertices. As noticed in the previous tests the refinement strategy characterized by the parameter $c_{\rho} = 1.5$ clearly tends to naturally converge to a good quality triangular mesh and in all the cases, after few iterations, the mesh tends to be composed by elements with three or four vertices and all the cells can be definitely defined good quality cells.

9. Conclusions

In this paper we propose a refinement algorithm for polygonal meshes with convex cells. We have tested its behaviour with an adaptive mesh refinement algorithm for virtual element methods (VEM), but it can be applied to any other polygonal method such as Discontinuous Galerkin methods, Hybrid High Order methods or Mimetic Finite Differences. The algorithm requires the selection of a suitable splitting direction for each element, the proposed strategies are based on the maximum moment direction or the longest diagonal direction that perform in a comparable way.

With respect to the algorithm proposed in [15] the main improvement is a strategy to avoid multiple splitting of an edge in the same refinement step.
that are often responsible of a degeneration of the quality of the elements. Moreover, the algorithm always avoids the creation of relative small edges during the refinement.

A set of geometrical parameters, easily computable, is tested in order to characterize the quality of the elements. Among the tested parameters we have found that the parameters $AR^{Rr}_E$ and $AR^{edge}_E$ can provide a measure of the quality of an element in all the situations. An additional important property in the VEM discretization is related to an upper bound of the ratio between the number of vertices of the mesh over the number of cells ($E^P_{#T}$), in particular, the proposed algorithm allows the increasing by one of the number of the vertices of the polygons produced by the refinement in a very particular unusual situation, in all the other situations the number of vertices of the produced polygons is less or equal to the number of vertices of the refined polygon and in the most common situations this number decreases if the number of vertices is larger than four. Depending on the parameter $c$, we have verified that, starting from general polygons, the mesh tends toward a good quality triangular, quadrilateral or mixed triangular-quadrilateral mesh. These meshes are characterized by a better efficiency (smaller $E^P_{#T}$) with respect to other meshes of convex elements.

A general advantage of a conforming polygonal method that allows aligned edges is that it does not require a conformity recovery after an adaptive refinement iteration. This property trivially bounds the number of cells produced in each refinement iteration with the number of cells marked for refinement that is a property useful for optimality analyses of an adaptive mesh refinement approach.

The advantage of the proposed refinement method relies on the fact that the generation of an initial polygonal mesh can be usually replaced by a description of the domain geometry in terms of very coarse convex cells only considering the geometric constraints. This given coarse-mesh/geometry-description can be refined by an adaptive method producing a good quality mesh and a reliable solution fully driven by the problem.

References

[1] R. Verfürth, A review of a posteriori error estimation and adaptive mesh-refinement techniques, Wiley-Teubner, 1996.

[2] W. Dörfler, A convergent adaptive algorithm for Poisson’s equation, SIAM Journal on Numerical Analysis 33 (3) (1996) 1106–1124.

[3] J. M. Cascon, C. Kreuzer, R. H. Nochetto, K. G. Siebert, Quasi-optimal convergence rate for an adaptive finite element method, SIAM J. Numer. Anal. 46 (5) (2008) 2524–2550.
[4] R. Stevenson, An optimal adaptive finite element method, SIAM Journal on Numerical Analysis 42 (5) (2005) 2188–2217.

[5] R. P. Stevenson, Optimality of a standard adaptive finite element method, Foundations of Computational Mathematics 7 (2007) 245–269.

[6] L. Beirão da Veiga, G. Manzini, An a posteriori error estimator for the mimetic finite difference approximation of elliptic problems, Internat. J. Numer. Methods Engrg. 76 (11) (2008) 1696–1723.

[7] L. Beirão da Veiga, G. Manzini, Residual a posteriori error estimation for the virtual element method for elliptic problems, ESAIM Math. Model. Numer. Anal. 49 (2) (2015) 577–599.

[8] L. Beirão da Veiga, G. Manzini, L. Mascotto, A posteriori error estimation and adaptivity in hp virtual elements, Numer. Math. 143 (1) (2019) 139–175.

[9] A. Cangiani, E. H. Georgoulis, T. Pryer, O. J. Sutton, A posteriori error estimates for the virtual element method, Numer. Math. 137 (4) (2017) 857–893.

[10] S. Berrone, A. Borio, A residual a posteriori estimate for the virtual element method, Mathematical Models and Methods in Applied Sciences 27 (08) (2017) 1423–1458.

[11] P. F. Antonietti, S. Berrone, A. Borio, A. D’Auria, M. Verani, S. Weisser, Anisotropic a posteriori error estimate for the virtual element method, IMA Journal of Numerical Analysis.

[12] L. Beirão da Veiga, C. Lovadina, A. Russo, Stability analysis for the virtual element method, Mathematical Models and Methods in Applied Sciences 27 (13) (2017) 2557–2594.

[13] T. Sorgente, S. Biasotti, G. Manzini, M. Spagnuolo, The role of mesh quality and mesh quality indicators in the virtual element method (2021).

[14] S. Berrone, A. Borio, Orthogonal polynomials in badly shaped polygonal elements for the Virtual Element Method, Finite Elements in Analysis & Design 129 (2017) 14–31.

[15] S. Berrone, A. Borio, A. D’Auria, Refinement strategies for polygonal meshes applied to adaptive vem discretization, Finite Elements in Analysis and Design 186 (2021) 103502.

[16] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, A. Russo, Basic principles of virtual element methods, Mathematical Models and Methods in Applied Sciences 23 (01) (2013) 199–214.
[17] B. Ahmad, A. Alsaedi, F. Brezzi, L. D. Marini, A. Russo, Equivalent projectors for virtual element methods, Computers & Mathematics with Applications 66 (2013) 376–391.

[18] L. Beirão da Veiga, F. Brezzi, L. D. Marini, A. Russo, Virtual element methods for general second order elliptic problems on polygonal meshes, Mathematical Models and Methods in Applied Sciences 26 (04) (2015) 729–750.

[19] S. Berrone, A. Borio, F. Marcon, Lowest order stabilization free virtual element method for the poisson equation (2021). arXiv:2103.16896

[20] P. Grisvard, Elliptic Problems in Nonsmooth Domains, Society for Industrial and Applied Mathematics, 2011.

[21] J. R. Shewchuk, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, in: M. C. Lin, D. Manocha (Eds.), Applied Computational Geometry: Towards Geometric Engineering, Vol. 1148 of Lecture Notes in Computer Science, Springer-Verlag, 1996, pp. 203–222, from the First ACM Workshop on Applied Computational Geometry.

[22] J. R. Shewchuk, Delaunay refinement algorithms for triangular mesh generation, Computational Geometry 22 (1) (2002) 21–74, 16th ACM Symposium on Computational Geometry.

[23] C. Talischi, G. Paulino, A. Pereira, I. Menezes, Polymesher: A general-purpose mesh generator for polygonal elements written in matlab, Structural and Multidisciplinary Optimization 45 (3) (2012) 309–328.

[24] L. Beirão da Veiga, C. Canuto, R. H. Nochetto, G. Vacca, M. Verani, Adaptive vem: Stabilization-free a posteriori error analysis (2021). arXiv:2111.07656