International Journal of Numerical Methods and Applications

Published Online: July 2015
http://dx.doi.org/10.17654/IJNMAJun2015_087_117
Volume 13, Number 2, 2015, Pages 87-117 **ISSN: 0975-0452**

HIGHER-ORDER VIRTUAL NODE METHOD FOR POLYGONAL ELEMENTS AND APPLICATION OF *h*-ADAPTIVITY

Hyeon Cheol Oh and Byung Chai Lee*

Department of Mechanical Engineering
Korea Advanced Institute of Science and Technology
373-1, Guseong-dong, Yuseong-gu
Daejeon 305-701, Republic of Korea
e-mail: bchlee@kaist.ac.kr

Abstract

This paper presents a higher-order approximation of polygonal elements using the generalized finite element method. The approximation is constructed using a virtual-node polygonal element based on the partition of unity coupled with the polynomial nodal approximation. Because the approximation functions are polynomials, the numerical integration can be evaluated accurately using Gauss quadrature. The proposed method passes the higher-order patch test and yields an optimal convergence rate for polygonal meshes. This higher-order method is also applied to the *h*-adaptive method on triangular quadtree mesh, which allows arbitrary-level hanging nodes. The study results demonstrate that the process of the *h*-adaptive refinement using the proposed method can deliver accuracy comparable with that of the red-green refinement while being even simpler, and this is demonstrated using several numerical tests for the Poisson problem.

Received: April 14, 2015; Revised: May 13, 2015; Accepted: May 19, 2015

2010 Mathematics Subject Classification: 65N30, 65N50.

Keywords and phrases: polygonal element, generalized finite element method, virtual node method, triangular quadtree mesh, hanging nodes.

*Corresponding author

1. Introduction

When finite element solutions to partial differential equations are used for problems with local singularities or steep gradient layers, uniform mesh refinement is not recommended due to its expensive computational costs. A clear remedy is to adaptively refine the mesh; then the local mesh resolution is varied automatically until the desired accuracy is reached. Over the past three decades, various adaptive refinement techniques have been investigated [1-3]. Among them, the *h*-adaptive refinement using subdivision has been widely used. A general method to subdivide elements is a quadtree subdivision [4-6].

In a triangular mesh with the quadtree subdivision applied, the triangles are subdivided into four congruent triangles. This method leads to recursive refinements of the mesh. The data structure of the quadtree mesh is simple and efficient for geometric operations such as locating neighbors [7, 8]. In addition, it can be easily combined with the adaptive mesh refinement. When using the quadtree mesh, however, hanging nodes on the edge of an element may be allowed due to the difference in the refinement levels between adjacent elements. In this case, inter-element compatibility is not satisfied. There are several methods to handle the hanging nodes. The first method is to eliminate the hanging nodes via transition elements [9]. This requires an additional remeshing process that is quite costly. In the second method, the approximation at the hanging nodes is constrained to neighboring corner nodes [10, 11]. This can be achieved through determining the solution at the hanging nodes as the average of the neighboring corner nodes or applying Lagrange multipliers to ensure inter-element compatibility. However, the constraint algorithms might become complicated when there are many hanging nodes on an edge. Therefore, in order to implement the code efficiently, the number of hanging nodes on an edge is generally limited to one (1-irregularity rule) [4, 12]. The third approach uses variable-node finite elements [13, 14]. They construct conforming shape functions on the 1-irregular mesh. In addition to these elements, a variable-node element that allows arbitrary-level hanging nodes has been developed [15]. For a

triangular variable node element that is degenerated from a quadrilateral variable node element, however, only one edge of the element is allowed for hanging nodes. In the fourth approach, polygonal finite elements have been used to manage hanging nodes [16-18]. This method satisfies the compatibility conditions naturally and is free from the 1-irregularity rule because polygonal elements can be regarded as non-matching elements.

Linear triangular elements (also known as constant strain triangle elements) are not recommended in general analyses because a large number of these elements are required to obtain sufficient accuracy. Similarly, higher-order (especially quadratic) elements are preferred in the *h*-adaptive refinement of the triangular quadtree mesh. If polygonal elements are used to treat the hanging nodes of the higher-order triangular element, then the polygonal elements should be approximated with the same order. Recently, methods for extension to higher-order polygonal elements have also been proposed [19]. Milbradt and Pick [20] presented higher-order interpolation functions for polygons based on natural element coordinates. Rand et al. [21] developed quadratic serendipity polygonal elements using generalized barycentric coordinates. Sukumar [19] proposed quadratic serendipity polygonal elements based on the maximum entropy principle. The shape functions of Sukumar's elements are constructed through a constrained optimization. However, the shape functions of all these elements are non-polynomials; thus, many integration points are required in order to reduce numerical integration errors when using general Gauss quadrature [17]. Efficient numerical integration techniques for the polygonal elements have also been investigated [22, 23].

Tang et al. [18] proposed the virtual node method, which is a polygonal finite element based on the partition of unity. In this method, the properties of the standard FEM shape functions are satisfied. Moreover, the special integration techniques are not required because the shape functions are polynomials. However, the higher-order extension of the virtual node polygonal element has not yet been proposed. The generalized finite element method (GFEM) [24-26] is one of many instances of the partition of

unity method (PUM) or partition of unity finite element method (PUFEM) [27]. The approximation spaces spanned by the nodal approximation can be reproduced precisely using the property of a partition of unity (PoU). Both polynomial and non-polynomial functions can be used as the nodal approximation. Thus, a higher-order element from a first order implementation can be obtained through using the polynomial nodal approximation.

This paper presents a higher-order approximation of the virtual node polygonal element (VPE) combined with the GFEM. Through applying the proposed method, an *h*-adaptive refinement on a triangular quadtree mesh with arbitrary-level hanging nodes is performed. The domain integrations can be evaluated accurately using Gauss quadrature because the shape functions are polynomials. Furthermore, the proposed method is based on the FEM framework; thus, it can be easily incorporated into an existing finite element code.

The outline of this paper is as follows. In Section 2, a higher-order approximation of the VPE method is described, together with brief reviews of the VPE method and GFEM. The *h*-adaptive refinement strategies and advantages of the *h*-adaptive strategy using the VPE method are described in Section 3. The Galerkin formulation for the Poisson equation and implementation of the proposed method are presented in Section 4. In order to demonstrate the performance of the proposed method, several numerical problems are presented in Section 5. Finally, the conclusions of this study are summarized in Section 6.

2. Higher-order Approximation of the VPE Method

2.1. VPE method

The VPE approximation is constructed using a combination of the least-squares and constant strain triangle (CST) approximations. Before introducing the VPE, consider the polygonal domains in 2D, as shown in Figure 1. Each polygon is bounded by edges and the nodes are defined where

two adjacent edges meet. As depicted in Figure 2, the polygon can be divided into N subtriangles through connecting its centroid and the nodes. In the VPE method, the subtriangles and centroid points are called *virtual triangles* and *virtual nodes*, respectively, as the nodal unknown at the centroid is eliminated from the system of equations.

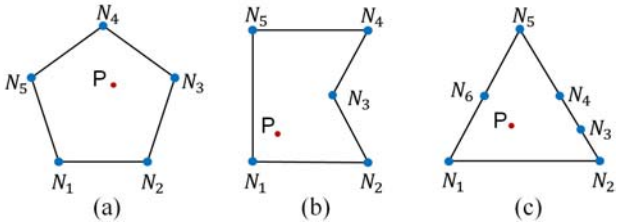


Figure 1. Polygonal domains: (a) pentagon, (b) concave pentagon and (c) triangle with side nodes.

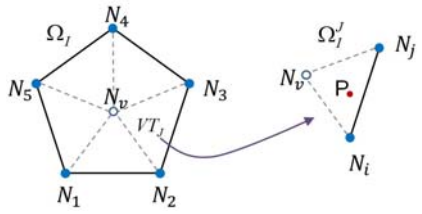


Figure 2. Illustration of dividing the polygonal domain into virtual triangles.

The unknown function $(u(\mathbf{x}))$ at any point within the virtual triangle (VT_J) can be approximated as follows:

$$u(\mathbf{x}) \approx u^h(\mathbf{x}) = R_C^J(\mathbf{x}) \cdot \hat{u}_C^J(\mathbf{x}) + R_L^J(\mathbf{x}) \cdot \hat{u}_L^J(\mathbf{x}), \tag{1}$$

where $\hat{u}_C^J(\mathbf{x})$ is the CST approximation, $\hat{u}_L^J(\mathbf{x})$ is the least-squares approximation, and $R_C^J(\mathbf{x})$ and $R_L^J(\mathbf{x})$ are weight functions [18].

In the least-squares method, the approximation function is described as:

$$\hat{u}_L(\mathbf{x}) = \sum_{I=1}^{N} \phi_I(\mathbf{x}) \tilde{u}_I = \mathbf{\Phi}^{\mathbf{T}} \tilde{\mathbf{u}}, \tag{2}$$

where N is the number of polygonal nodes and $\mathbf{\Phi}$ is the least-squares shape function, which can be defined using the polynomial terms $(\mathbf{p}(\mathbf{x}))$ that occur in Pascal's triangle as follows [28]:

$$\mathbf{\Phi}^{\mathbf{T}} = \mathbf{p}^{\mathbf{T}}(\mathbf{x})\mathbf{H}^{-1}\mathbf{g}, \tag{3}$$

$$\mathbf{H} = \sum_{I=1}^{N} \mathbf{p}(\mathbf{x}_I)\mathbf{p}^{\mathbf{T}}(\mathbf{x}_I), \tag{4}$$

$$\mathbf{g} = [\mathbf{p}(\mathbf{x}_1)\ \mathbf{p}(\mathbf{x}_2) \cdots \mathbf{p}(\mathbf{x}_N)]. \tag{5}$$

The CST approximation is given by

$$\hat{u}_C^J(\mathbf{x}) = \sum_I \varphi_I^J \tilde{u}_I = \varphi_i^J \tilde{u}_i + \varphi_j^J \tilde{u}_j + \varphi_{\mathrm{v}}^J \tilde{u}_{\mathrm{v}}, \tag{6}$$

where φ_I^J is the shape function in terms of the area coordinates and $\tilde{u}_I$ is the nodal unknown. Because the nodal unknown at the centroid $(\tilde{u}_{\mathrm{v}})$ is equal to the least-squares approximation at the centroid, equation (6) can be reconstructed without $\tilde{u}_{\mathrm{v}}$. In order to satisfy the properties such as the partition of unity and piecewise linear approximation on the polygonal boundary, the weight functions $R_C^J(\mathbf{x})$ and $R_L^J(\mathbf{x})$ should be represented as follows:

$$\begin{aligned} R_C^J(\mathbf{x}) &= \varphi_i^J(\mathbf{x}) + \varphi_j^J(\mathbf{x}), \\ R_L^J(\mathbf{x}) &= \varphi_{\mathrm{v}}^J(\mathbf{x}), \end{aligned} \tag{7}$$

where φ_i^J, φ_j^J and φ_{v}^J are the shape functions of the CST. Therefore, the VPE approximation can be rewritten as follows:

$$u^h(\mathbf{x}) = \sum_{I=1}^{N} N_I(\mathbf{x})\tilde{u}_I, \tag{8}$$

$$N_I(\mathbf{x}) = R_C(\mathbf{x})\{(\delta_{Ii} + \delta_{Ij})\varphi_I(\mathbf{x}) + \varphi_{\mathrm{v}}(\mathbf{x})\phi_I(\mathbf{x}_{\mathrm{v}})\} + R_L(\mathbf{x})\phi_I(\mathbf{x}), \tag{9}$$

where $N_I(\mathbf{x})$ is the VPE shape function of node I, as shown in Figure 3:

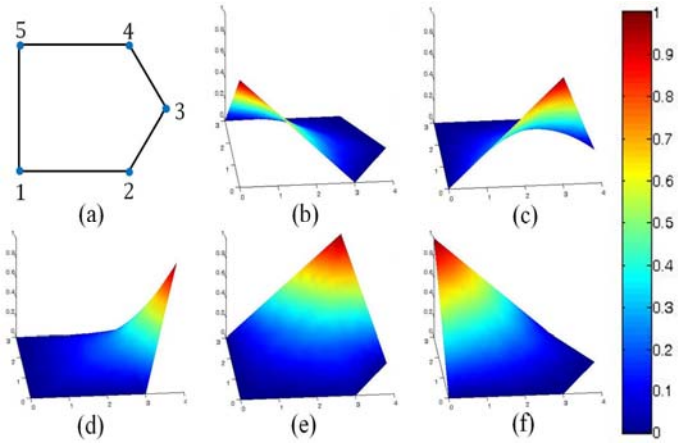


Figure 3. VPE shape functions: (a) polygonal element, (b) node 1, (c) node 2, (d) node 3, (e) node 4 and (f) node 5.

The VPE method satisfies the desirable properties of the finite element shape functions including the partition of unity, linear completeness, Kronecker delta property, and piecewise linear on the element boundary. Furthermore, the VPE shape functions are polynomials.

2.2. GFEM

The key concept of the GFEM is the partition of unity (PoU) [24, 27]. The PoU function is used not only to impose the inter-element condition but also to reproduce the local approximation function. In the GFEM, a domain (Ω) consists of an overlap of the domain (or cover) ω_i, which is defined by the union of elements sharing the node *i*. This domain ω_i, referred to as the node patch of node *i* [29], is illustrated in Figure 4. The functions that satisfy equation (10) form a PoU subordinate to the node patch of node *i*, ω_i,

$$\sum_i \varphi_i(\mathbf{x}) = 1, \quad \forall \mathbf{x} \in \Omega. \tag{10}$$

The linear Lagrangian FE shape functions are typically chosen as the PoU functions in the standard GFEM. The GFEM shape functions are constructed using the product of the PoU functions and local functions. The local functions are also called the *nodal approximation*. These functions can

be either polynomials or non-polynomials (e.g., harmonic functions and singular functions).

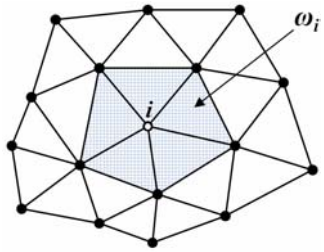


Figure 4. Node patch of node *i*.

Notable advantages of the GFEM are that the implementation of the essential boundary conditions is easy; the selection of an appropriate nodal approximation enables the good approximation of the solution; and *p*-adaptivity can be implemented simply. In particular, unlike the conventional FEM, a higher-order approximation can be obtained without needing to add nodes on an element edge or face. For more details of the GFEM, please refer to the literatures [24, 25, 27].

2.3. Enriched VPE

As mentioned in Subsection 2.1, the VPE shape functions satisfy the PoU, and these can be used as the PoU functions of the GFEM. Because this study focuses on the higher-order approximation on polygons, uniform polynomials are chosen as the nodal approximation. The VPE approximation of order *p* applying the GFEM is constructed as follows:

$$u^h(\mathbf{x}) = \sum_{i=1}^{n} \varphi_i(\mathbf{x}) \widetilde{u}_i(\mathbf{x})$$

$$= \sum_{i=1}^{n} \varphi_i(\mathbf{x}) \sum_{j=1}^{D_L} L_{ji}(\mathbf{x}) a_{ji} = \sum_{i=1}^{n} \sum_{j=1}^{D_L} \phi_{ji}(\mathbf{x}) a_{ji}, \tag{11}$$

where $\varphi_i(\mathbf{x})$ is the VPE shape function of node *i*, $\widetilde{u}_i(\mathbf{x})$ is a local approximation of *u* over ω_i and *n* is the total number of the nodes in the

discretized domain. $\phi_{ji}(\mathbf{x})$ is an enriched shape function of order p and a_{ji} are nodal degrees of freedom (dofs). D_L is the number of terms of the polynomial nodal approximation $L_{ji}(\mathbf{x})$ of degrees less than or equal to $p-1$. The sets $\{L_{ji}(\mathbf{x})\}_{j=1}^{D_L}$ considered in this study are given as follows:

$$\begin{aligned}\{L_{ji}(\mathbf{x})\}_{j=1}^{D_L} &= \{1\}, \text{ constant enrichment}\\ &= \{1, \bar{x}, \bar{y}\}, \text{ linear enrichment}\\ &= \{1, \bar{x}, \bar{y}, \overline{xy}, \bar{x}^2, \bar{y}^2\}, \text{ quadratic enrichment,}\end{aligned} \tag{12}$$

where $\bar{x} = (\mathbf{x} - \mathbf{x}_i)/h_i$ are normalized coordinates and h_i is a scaling factor [25]. This shifted function enables the direct imposition of the Dirichlet boundary conditions. This element is named the "enriched VPE". Thus, the enriched VPE of order $p = 2$ indicates a quadratic element using the linear enrichment. Also, the enriched VPE of order $p = 1$ is exactly the same as the VPE.

Figure 5 presents the enriched shape function of the VPE at node 3 as described in Figure 3(d). As illustrated in Figure 5, the VPE shape function inherits the properties of both the compact support of the PoU function and the approximate features of the nodal approximation.

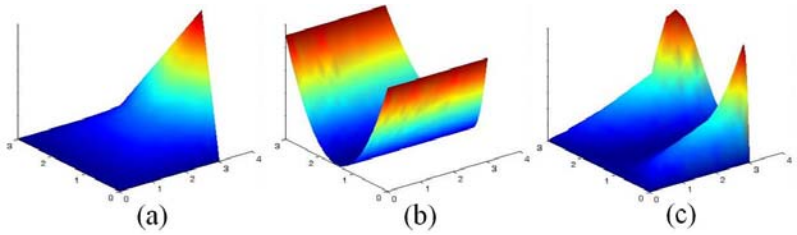


Figure 5. Construction of an enriched VPE shape function: (a) PoU function $\varphi_3(\mathbf{x})$, (b) nodal approximation $L(\mathbf{x}) = (y - y_3)^2$ and (c) enriched VPE shape function $\phi_3(\mathbf{x}) = \varphi_3(\mathbf{x})L(\mathbf{x})$.

3. *h*-adaptive Refinement Strategy on Triangular Quadtree Meshes

3.1. Triangular quadtree data structure

h-adaptive refinements are widely used for the efficient improvement of the FE solution. The element subdivision and mesh regeneration are representative methods of the *h*-refinement, and the former is more commonly used. The subdivision for a triangular mesh is generally a quadtree subdivision in which the triangle is split into four subtriangles [5].

As shown in Figure 6(a), each of these can be divided into new four congruent subtriangles. This successive process is regarded as a triangular quadtree refinement. This refinement process can be represented as the hierarchical tree structure, as depicted in Figure 6(b). Each node of the tree corresponds to generated or unrefined elements during the refinement process. In Figure 6(b), the divided triangle (e.g., element 2) and the new subtriangles (e.g., elements 6-9) are called the *father* and *sons*, respectively, [4]. The level number of the triangle is defined as the number of subdivisions from an initial mesh; thus, the level number of the father element is one higher than that of its son elements.

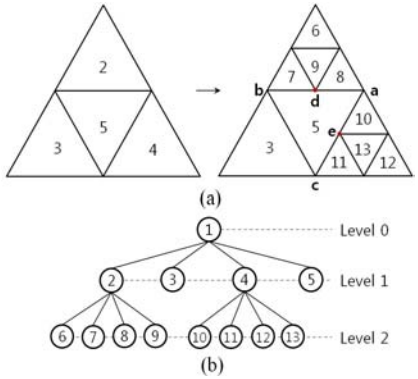


Figure 6. Triangular quadtree refinement: (a) successive quadtree meshes and (b) their hierarchical tree structure.

The tree data structure enables fast information retrieval that is necessary for the refinement. Furthermore, the mesh data, in which the element connectivity as fundamental information and the subdivision level of the

element and data pointers to the father element and son elements are included, can be stored and maintained in an efficient tree. By virtue of these features of the tree data structure, the *h*-adaptive refinement becomes more effective.

During the refinement process, irregular meshes with hanging nodes can be created depending on the difference in the level among the neighbor elements. For triangular meshes, the red-green refinement strategy is often used in the *h*-adaptive refinement in order to maintain the regularity of the mesh.

3.2. Red-green refinement

An algorithm for the red-green refinement consists of two refinements: one divides an element into four congruent elements (referred to as a "red refinement") and the other bisects an element with one hanging node (referred to as a "green refinement") as shown in Figure 7 [4, 30]. In this strategy, the refinement region to treat the hanging nodes is very local compared with the longest edge refinement [31]. Furthermore, the minimum angle of the mesh after the refinement steps is bounded to be a half of the minimum angle of the initial mesh. During the refinement process, however, coarsening of the refined elements to maintain the minimum angle and refining those elements should be repeated until the hanging nodes in the mesh are removed. This incurs additional computational costs. Moreover, the elements, neither the marked element nor its neighbor elements, can be refined. Using a polygonal element such as the VPE is an alternate approach to efficiently handle the hanging nodes.

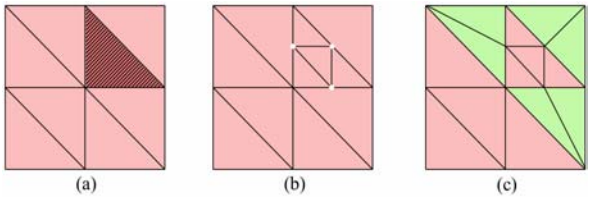


Figure 7. Red-green refinement algorithm: (a) a marked element to be refined, (b) refinement of the marked element (red refinement) and (c) additional refinements to remove hanging nodes (green refinement).

3.3. Conforming shape functions on triangular quadtree meshes

Through replacing the transition element with the higher-order VPE, an irregular mesh with hanging nodes is generated. Before demonstrating the conformity of the higher-order VPE shape functions at the hanging nodes on the triangular quadtree meshes, the conformity of the VPE shape function is addressed. For example, if the VPE is applied to element 5 of Figure 6(a), then its shape function at hanging node *d* is piecewise linear along the edge *a*-*d*-*b*, as illustrated in Figure 8(a). In order to demonstrate the conformity of the VPE shape function, the shape functions at node *a* of the elements using the VPE and the CST elements are given in Figures 8(b) and 8(c), respectively. The resulting shape functions using the VPE (Figure 8(b)) conform, whereas those of the CST elements (Figure 8(c)) do not.

For the higher-order VPE, this conformity is naturally retained because the VPE shape functions are used as the PoU functions. As illustrated in Figure 9, through the product of the VPE shape function (Figure 9(a)) and the nodal approximation (Figure 9(b)), the higher-order VPE shape function of node *d* (Figure 9(c)) conforms on edge *a*-*d*-*b*.

Therefore, if the higher-order VPE is used on triangular quadtree meshes, then the inter-element compatibility can be retained without the transition element or constrained approximation. In addition, because arbitrary-level hanging nodes are allowed, an additional process to maintain the irregularity rule is not required. The quadtree data structure can also be utilized without requiring special modification.

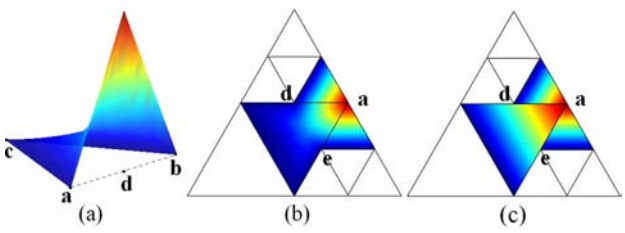


Figure 8. (a) A VPE shape function of the hanging node *d* and shape functions at node *a* of the elements applying the (b) VPE and (c) CST elements.

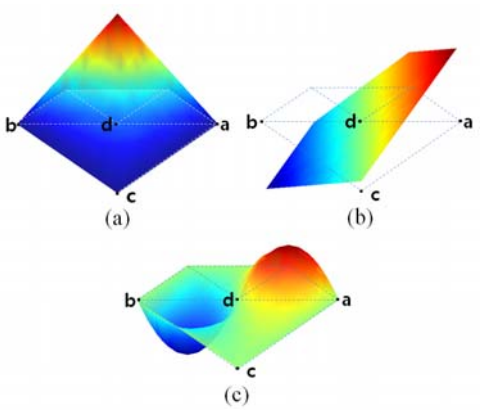


Figure 9. Functions of the hanging node *d*: (a) a VPE shape function $\varphi_d(x)$, (b) nodal approximation $L(x) = x - x_d$ and (c) enriched VPE shape function $\phi_d(x) = \varphi_d(x)L(x)$.

3.4. *h*-adaptive strategy

The discretization error is defined as follows:

$$e = u - u^h, \tag{13}$$

where u is the exact solution and u^h is the finite element approximation. The above error is a function form and thus, a norm is introduced to measure the magnitude of the error. The energy norm is most commonly used. The error in the energy norm can be written as follows:

$$\| e_i \|_E = \left[\int_{\Omega_i} \nabla e_i^T \nabla e_i \, d\Omega \right]^{1/2}, \tag{14}$$

$$\| e \|_E = \left[\sum_{i=1}^{N_e} \| e_i \|_E^2 \right]^{1/2}, \tag{15}$$

where N_e is the total number of elements, $\| e \|_E$ is the error in the energy norm in the entire domain Ω, and $\| e_i \|_E$ is the error in the energy norm in the element domain Ω_i. $\| e_i \|_E$ is also considered as error contributions from

element *i*. The relative error in the energy norm is defined as:

$$R_E = \frac{\| e \|_E}{\| u \|_E}, \tag{16}$$

where $\| u \|_E$ is the exact solution in the energy norm. This relative error is a factor that relatively defines the discretization error using the finite element approximation. The *h*-adaptive refinement procedure is repeated until the following equation is satisfied:

$$R_E \leq R_a, \tag{17}$$

where R_a is the predetermined permissible relative error [32]. Equation (17) is called a *stopping criterion*. In this procedure, whether each element is refined or not is determined based on the elemental error contributions (η_i).

In order to select the elements to be refined, various refinement criteria have been reported in the literatures [33, 34]. Two well-known criteria are discussed here. The first criterion is to refine a fixed fraction (θ) of the total elements with the largest error contributions (η_i), called the *fixed fraction criterion*. This is useful for controlling the number of refined elements, but it is insensitive for capturing the local properties such as singularities and concentrations in the problem. Thus, the mesh is overly refined when the problem has local properties. The second criterion is to refine the elements that have an η_i value larger than or equal to a fraction (θ) of the largest error contributions, called the *maximum criterion*. This is much closer to the adaptive refinement than the former criterion. That is, the elements to be refined are highly concentrated in the region near the local properties when it is present. In this study, which considers numerical problems that have local properties, the maximum criterion is used as the refinement criterion: we refine the elements for which

$$\eta_i \geq \theta \cdot \max_i \eta_i. \tag{18}$$

4. Galerkin Formulation and Implementation

Applications of the Poisson equation are investigated in order to assess the performance of the enriched VPE. Consider the following elliptic boundary value problem:

$$-\nabla^2 u(\mathbf{x}) = Q(\mathbf{x}) \text{ in } \Omega, \tag{19}$$

$$\frac{\partial u}{\partial n} = f \text{ on } \Gamma_N, \tag{20}$$

$$u = g \text{ on } \Gamma_D, \tag{21}$$

where $Q(\mathbf{x})$ is the source term obtained by the exact solution $u(\mathbf{x})$ of the problem, and Ω is the problem domain. f is the prescribed flux on the Neumann boundary (Γ_N) and g is the prescribed solution on the Dirichlet boundary (Γ_D). The Galerkin weak form of these equations can be expressed as follows:

$$\int_\Omega (\nabla \delta u)^T (\nabla u) d\Omega = \int_\Omega \delta u^T Q d\Omega + \int_{\Gamma_N} \delta u^T f d\Gamma. \tag{22}$$

We now consider the finite element discretization of the Poisson equation. Discretized system equations are obtained through substituting the approximation function (equation (11)) for the trial and test function into the weak form (equation (22)), and the arbitrariness of the variations is considered. The resulting equations can be expressed as the following matrix form:

$$\mathbf{Kd} = \mathbf{f}, \tag{23}$$

$$\mathbf{K}_{ij} = \sum \mathbf{K}_{ij}^e, \quad \mathbf{f}_i = \sum \mathbf{f}_i^e, \quad \mathbf{d} = \begin{Bmatrix} \mathbf{u} \\ \mathbf{a} \end{Bmatrix}, \tag{24}$$

$$\mathbf{K}_{ij}^e = \int_{\Omega_e} \nabla \phi_i^T \nabla \phi_j d\Omega,$$

$$\mathbf{f}_i^e = \int_{\Omega_e} \phi_i^T Q d\Omega + \int_{\Gamma_N} \phi_i^T f d\Gamma, \tag{25}$$

where **d** is the nodal unknown vector, **K** is the global stiffness matrix and **f** is the global source vector.

As in the standard FEM, numerical integration is used to evaluate $\mathbf{K}_{ij}^{e}$ and $\mathbf{f}_{i}^{e}$ (equation (25)). The numerical integration over the polygonal domain is performed for each virtual triangle as follows:

$$\int_{\Omega_I} f d\Omega = \sum_{J=1}^{N} \int_{\Omega_I^J} f d\Omega. \tag{26}$$

The shape functions of the traditional polygonal finite elements such as Wachspress coordinates and mean value coordinates are rational polynomials [17]. Therefore, numerous integration points are used to reduce the numerical integration errors. However, the shape functions of the proposed method are polynomials because both the VPE shape functions used as the PoU functions and the nodal approximation are polynomials. Thus, the domain integrations can be evaluated accurately using Gauss quadrature for the triangle (or Hammer quadrature).

When both the PoU functions and the nodal approximation are polynomials, the set of the shape functions may be linear dependent. This results in the system of equations potentially being positive semi-definite. This system can be successfully solved using the iterative method proposed by Babuška and his colleagues [35].

5. Numerical Results

In this section, several numerical examples are investigated in order to test the accuracy and convergence of the higher-order enriched VPE and the validation of its application for the *h*-adaptivity. First, the patch tests for the Laplace equation are performed on different types of mesh. Second, the convergence test for the Poisson problem is conducted on polygonal meshes. Then the *h*-adaptive strategy using the proposed method is applied to solve three Poisson problems that have a smooth solution with local singularities or steep gradient layers. For the *h*-adaptive refinement, the fixed fraction θ of the maximum criterion is set to 0.7.

The relative errors in the energy norm (equation (16)) and in the displacement norm are used to evaluate the accuracy and convergence. The relative error in the displacement norm is defined as follows:

$$R_{L^2} = \left[\frac{\sum_{i=1}^{N_e} \int_{\Omega} e_i^T e_i d\Omega}{\sum_{i=1}^{N_e} \int_{\Omega} u_i^T u_i d\Omega} \right]^{1/2} . \tag{27}$$

5.1. Patch tests

Linear and quadratic patch tests are performed for the Laplace equation $-\nabla^2 u(\mathbf{x}) = 0$ in $\Omega = (0, 1)^2$.

The Dirichlet boundary conditions are imposed on the boundary of the domain Ω and the exact solutions of the patch tests are:

$$u(\mathbf{x}) = x + y \text{ for linear patch test,}$$

$$u(\mathbf{x}) = 1 - x + 5y - 2xy - 4x^2 + 4y^2 \text{ for quadratic patch test.} \tag{28}$$

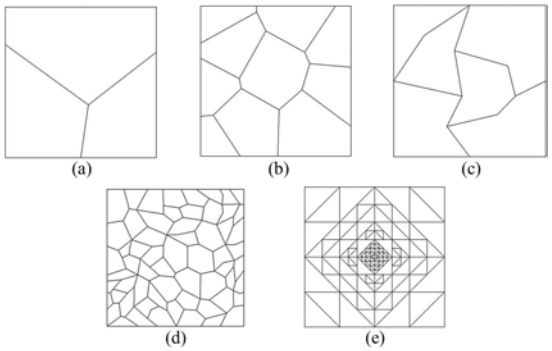


Figure 10. Patch tests for various types of the mesh: (a)-(b) convex polygonal meshes, (c) concave polygonal meshes, (d) mixed polygonal mesh and (e) triangular quadtree mesh.

As shown in Figure 10, various types of mesh, including convex polygonal meshes, concave polygonal meshes, mixed polygonal mesh

and triangular quadtree mesh are considered. The relative error in the displacement norm for the patch tests is presented in Table 1. It is demonstrated that the enriched VPE with orders of $p = 1$ and $p = 2$ pass the linear and quadratic patch tests near the machine precision, respectively.

Table 1. Relative error in the displacement norm for the linear and quadratic patch tests

Meshes	Linear patch test	Quadratic patch test	
	Shape functions of order $p = 1$	Shape functions of order $p = 1$	Shape functions of order $p = 2$
(a)	1.53×10^{-16}	8.63×10^{-2}	2.96×10^{-16}
(b)	4.57×10^{-16}	1.85×10^{-2}	4.39×10^{-16}
(c)	2.29×10^{-16}	2.46×10^{-2}	9.97×10^{-16}
(d)	8.52×10^{-16}	3.05×10^{-3}	1.00×10^{-15}
(e)	4.75×10^{-16}	5.46×10^{-3}	4.96×10^{-16}

5.2. Convergence test

The convergence of the enriched VPE is investigated for the Poisson problem. The following Poisson equation is solved using the Dirichlet boundary conditions:

$$-\nabla^2 u(\mathbf{x}) = Q(\mathbf{x}) \text{ in } \Omega,$$
$$u = g \text{ on } \partial\Omega. \tag{29}$$

The domain of interest is a bi-square, $\Omega = (-1, 1)^2$. The exact solution $u(\mathbf{x})$ and source term $Q(\mathbf{x})$ are:

$$u(\mathbf{x}) = \sin(2\pi x)\sin(\pi y),$$
$$Q(\mathbf{x}) = 5\pi^2 \sin(2\pi x)\sin(\pi y). \tag{30}$$

In this section, the uniform refinement on the convex and concave meshes presented in Figure 11 is used to examine the convergence of the proposed method. Three convex meshes with 78, 210 and 736 nodes are considered. Similarly, three concave meshes with 61, 217 and 817 nodes are used. Figure 9 presents the rate of convergence of the relative error in the displacement norm and energy norm for each type of mesh. r denotes the average slope of the convergence curve, which indicates the rate of convergence. As seen in Figure 12, the enriched VPE of order $p = 1$ achieves the theoretical convergence rate of the relative error in $R_{L^2} = 1.0$ and $R_E = 0.5$. The enriched VPE of order $p = 2$ also yields it in $R_{L^2} = 1.5$ and $R_E = 1.0$. Furthermore, it is observed that the rate of convergence does not affect the type of polygonal mesh, i.e., convex or concave mesh.

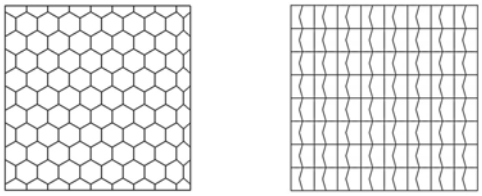
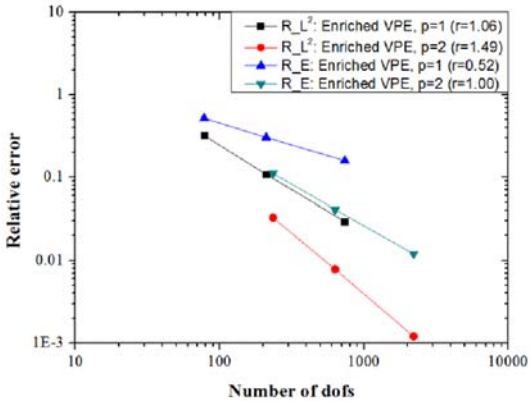
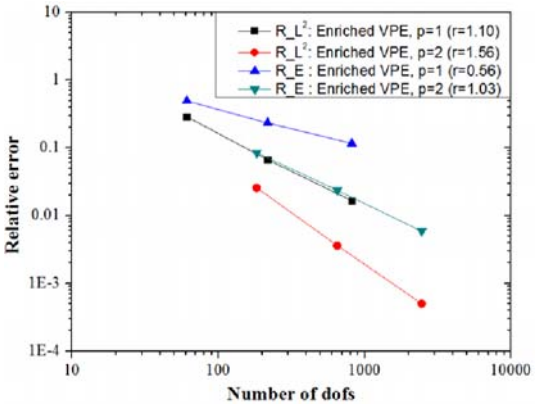


Figure 11. Convergence test: (a) convex polygonal mesh and (b) concave polygonal mesh.



(a)



(b)

Figure 12. Convergence test: the convergence rate of the relative error in the displacement norm (R_{L^2}) and energy norm (R_E): (a) convex polygonal mesh and (b) concave polygonal mesh.

5.3. *L*-shaped domain problem

In this section, the convergence of the *h*-adaptive refinement strategy based on the triangular quadtree mesh with the application of the proposed method is compared with the *h*-uniform refinement. The problem to be considered is the Laplace equation in the *L*-shaped domain $-\nabla^2 u = 0$ in $\Omega = (-1, 1)^2 \setminus (0, 1) \times (-1, 0)$. The exact solution [36] imposed on the domain boundary $\partial\Omega$ as a Dirichlet boundary conditions is:

$$u(\mathbf{x}) = r(\mathbf{x})^{2/3} \sin\left(\frac{2\theta(\mathbf{x})}{3}\right). \tag{31}$$

The solution has a local singularity at the re-entrant corner. The *h*-adaptive strategy applying the enriched VPE $(p = 1 \text{ and } p = 2)$ and the *h*-uniform refinement with the CST $(p = 1)$ and linear strain triangle (LST) $(p = 2)$ element are performed. The initial mesh consists of twelve regular triangles. For comparison of the convergence, the refinement procedure is repeated until 1000 dofs are reached.

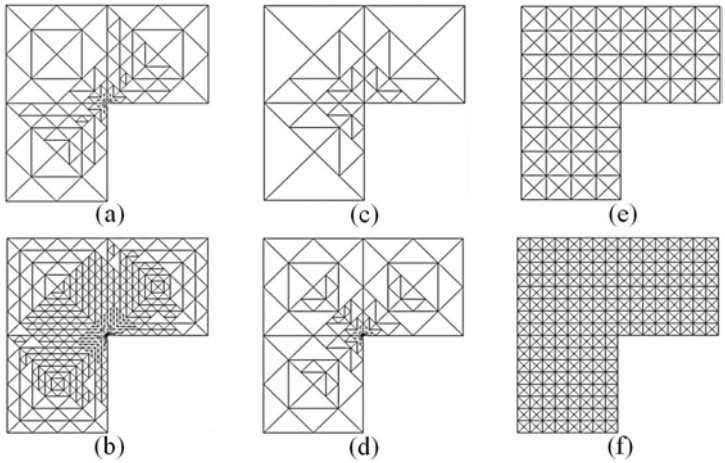


Figure 13. *L*-shaped domain problem: *h*-adaptive refinement using enriched VPE $(p = 1)$ with meshes after the (a) 5th step (109 dofs) and (b) 9th step (430 dofs); *h*-adaptive refinement using the enriched VPE $(p = 2)$ with meshes after the (c) 2nd step (111 dofs) and (d) 9th step (450 dofs); and the *h*-uniform refinement using the CST element with meshes after the (e) 2nd step (113 dofs) and (f) 3rd step (417 dofs).

Some steps of the refinement procedure using the enriched VPE $(p = 1, 2)$ and CST element are shown successively in Figure 13. Triangular quadtree meshes with hanging nodes can be seen in Figures 13(a) to 13(d). In order to capture the singular behavior, meshes using the *h*-adaptive strategy are densely refined near the re-entrant corner. Comparison of the convergence rate is illustrated in Figure 14. Despite the singularity within the domain, the *h*-adaptive refinement using the enriched VPE constructs optimal meshes. That is, this achieves the theoretical convergence rate of the relative error in $R_E = p/2,$ in contrast with the *h*-uniform refinement where the rate of convergence is limited to 1/3. It can be seen clearly that applying the second-order enriched VPE leads to better accuracy and a faster convergence rate than the first-order one.

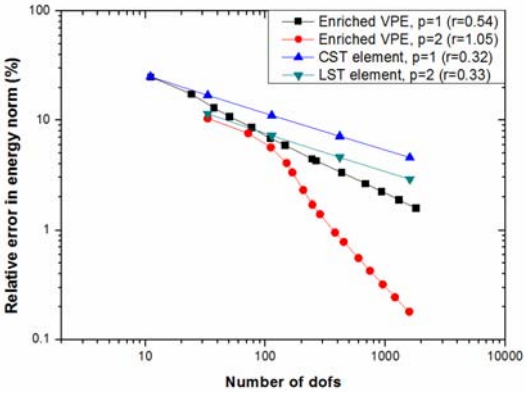


Figure 14. *L*-shaped domain problem: the convergence rate of the relative error in the energy norm.

5.4. Internal layer problem

The performance of the *h*-adaptive refinement strategy using the proposed method is compared with the red-green refinement strategy. In order to achieve this, the Poisson equation with mixed boundary conditions (equations (19)-(21)) is solved on the unit square. This problem has a smooth solution with a steep interior layer. The geometry of the problem and imposed parts of the Dirichlet and Neumann boundary conditions are seen in Figure 15(a). $Q(\mathbf{x})$, f and g in equations (19)-(21) are chosen according to the exact solution to the problem [3]:

$$u(x,\ y) = a\tan(S\sqrt{(x-1.25)^2 + (y+0.25)^2} - \pi/3), \tag{32}$$

where S is a slope of the layer; $S = 60$ is used in this study.

The triangular quadtree refinement using the second-order enriched VPE and the red-green refinement based on the LST element were conducted. The initial mesh for both refinement strategies is presented in Figure 15(b). The predetermined permissible relative error (R_a) is set to 1% in this problem.

Figures 16 and 17 illustrate the meshes and their corresponding approximate solutions after the refinement step using the proposed method and red-green strategy, respectively. As the refinement step is iterated,

the mesh refinements in both strategies are gradually concentrated near the internal layer.

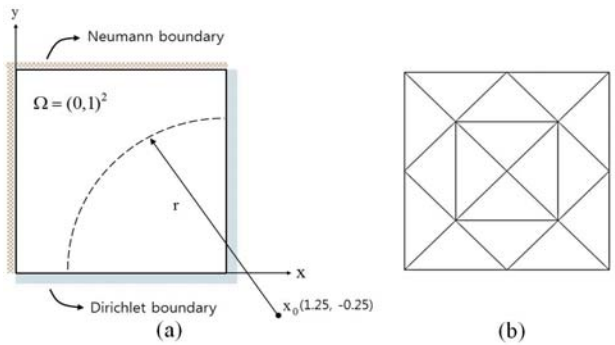


Figure 15. Internal layer problem: (a) geometry with imposed parts of the boundary and (b) initial mesh.

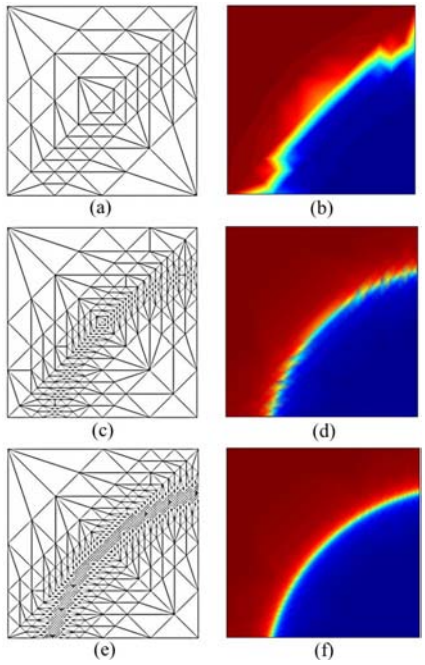


Figure 16. Internal layer problem with the *h*-adaptive refinement using the LST element $(p = 2)$: meshes after the (a) 3rd, (c) 7th and (e) 13th step and their corresponding approximate solutions in (b), (d) and (f).

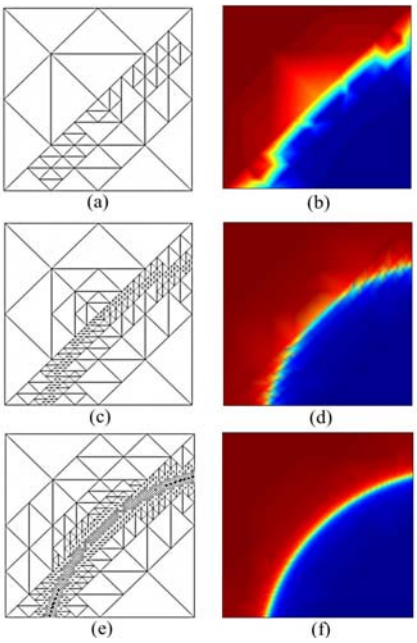


Figure 17. Internal layer problem of the *h*-adaptive refinement using the enriched VPE of order $p = 2$: meshes after the (a) 3rd, (c) 7th and (e) 13th step and their corresponding approximate solutions in (b), (d) and (f).

In the red-green strategy, in order to maintain not only the regularity of the mesh but also the minimum angle of the element, additional works to divide the unrefined elements are required. These elements may be slightly separated from the internal layer. In the refinement strategy using the proposed method on a triangular quadtree mesh, a locally refined mesh with hanging nodes is constructed. As illustrated in Figure 17, arbitrary-level hanging nodes are allowed in the proposed method.

The convergence curves of the relative error in the energy norm are illustrated in Figure 18. The stopping criterion is achieved within the nineteenth step for the red-green refinement strategy and the eighteenth step for the *h*-adaptive refinement strategy using the proposed method. Although the refinement process using the proposed method is simpler than the red-green strategy and no additional remeshing is required, it yields slightly better results.

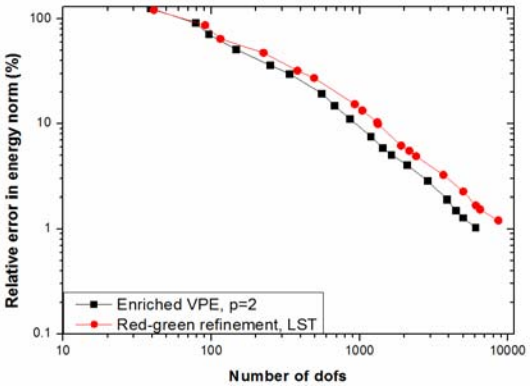


Figure 18. Internal layer problem: the convergence rate of the *h*-adaptive strategies.

5.5. Locally sharp gradient problem

In order to examine the performance of the *h*-adaptive strategy with the higher-order VPE, the Poisson problem that has a smooth solution with sharp gradient at the center is considered. The computational domain is $\Omega = (-1, 1)^2$ and the Dirichlet boundary conditions on $\partial\Omega$ correspond to the exact solution as follows:

$$u = \sin(x)\sin(y) + \exp(-100(x^2 + y^2)). \tag{33}$$

Figure 19 presents a 2D plot of the solution and initial mesh for the *h*-adaptive refinement. Due to the sharp behavior at the center, intensive mesh refinement near the center could be expected. First, the triangular quadtree refinement using the enriched VPE $(p = 2)$ and the red-green refinement based on the LST element are performed. In this case, the predetermined permissible relative error (R_a) was set to 1%.

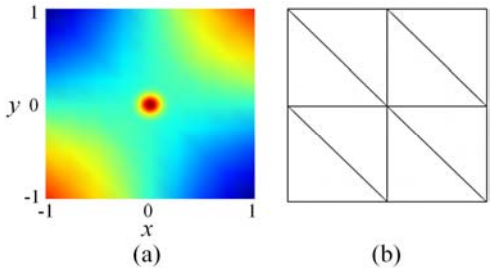


Figure 19. Locally sharp gradient problem: (a) 2D plot of the solution and (b) the initial mesh.

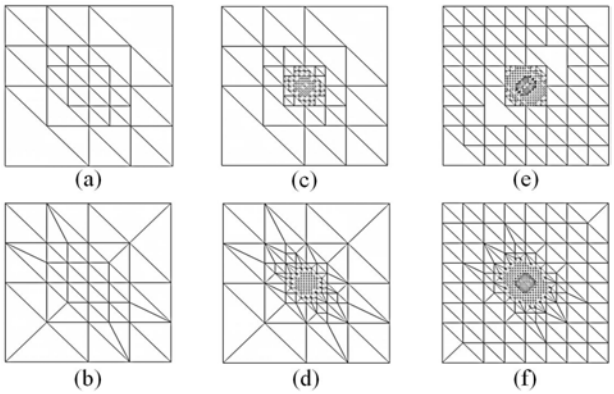


Figure 20. Locally sharp gradient problem: *h*-adaptive refinement meshes after the (a), (b) 2nd, (c), (d) 7th and (e), (f) 11th steps using enriched VPE of order $p = 2$ (top images) and LST element (bottom images).

Figure 20 presents several meshes after the triangular quadtree and red-green refinement steps. As expected, the mesh density significantly increases in the vicinity of the center. In particular, the *h*-adaptive refinement strategy using the proposed method generates a fine mesh with an arbitrary number of hanging nodes as seen in Figures 20(a) to 20(c). From the convergence curves (Figure 21(a)), it can be observed that the triangular quadtree refinement using the enriched VPE $(p = 2)$ provides a comparable level of accuracy to the red-green refinement based on the LST element.

In order to investigate the higher-order performance of the refinement using the proposed method, the *h*-adaptive refinements using the enriched

VPE with orders of 1, 2 and 3 are performed. The comparison results are presented in Figure 21(b). In the preasymptotic range of each convergence curve, the *h*-adaptive refinement using the lower-order enriched VPE can yield better results. This is related to meshes in the preasymptotic range being insufficient to reflect the solution near the high gradient region. After this range, as the approximate order increases, the convergence rate is faster and relative errors occur less. An *hp*-adaptive strategy using the proposed method can be a desirable solution in order to reduce the preasymptotic range and yield better convergence rates. This will be investigated in future work.

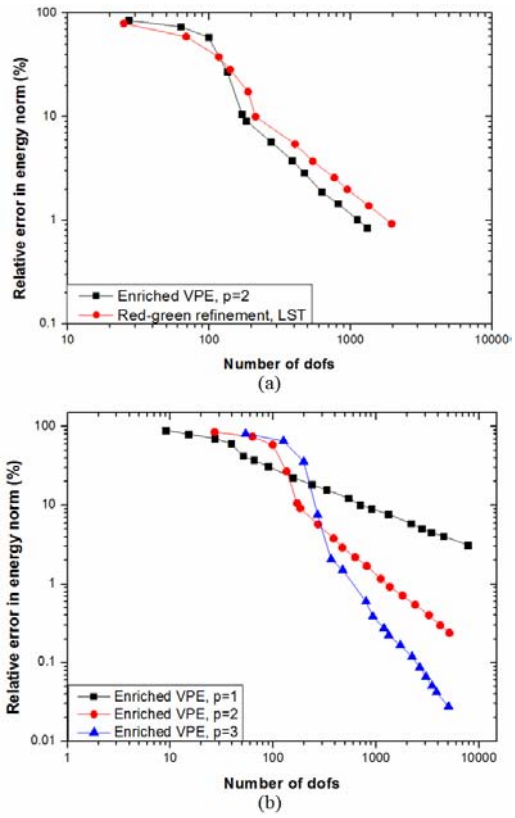


Figure 21. Locally sharp gradient problem: (a) the convergence rate of the *h*-adaptive strategies using the enriched VPE of order $p = 2$ and the LST element and (b) the comparison results among the higher-order enriched VPE.

6. Conclusions

A higher-order approximation for the virtual node polygonal element has been proposed based on the GFEM. The higher-order shape functions are constructed using the polynomial nodal approximation coupled with the VPE shape functions that satisfy the partition of unity. Because the functions are polynomials, precise numerical integration is possible via Gauss quadrature. The formulation of the proposed method is simple and the inter-element compatibility with arbitrary-level hanging nodes on element edges can be easily satisfied without special treatment. The method has been applied to the *h*-adaptive refinement on a triangular quadtree mesh, which allows arbitrary-level hanging nodes. The higher-order enriched VPE passed the patch tests within the machine precision and yielded the optimal convergence rate. The efficiency and accuracy of the proposed method were demonstrated through several numerical problems with *h*-adaptive refinements. Consequently, the accuracy of the proposed method is similar or slightly better than the red-green refinement. Applications in three-dimensional problems including crack propagations and non-matching mesh will be investigated in future research. The implementation of the *hp*-adaptive refinement using the proposed method is currently being undertaken.

Acknowledgement

This research was supported by the EDISON Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (No. 2014M3C1A6038853).

References

[1] I. Babuška and W. C. Rheinboldt, A-posteriori error estimates for the finite element method, International Journal for Numerical Methods in Engineering 12(10) (1978), 1597-1615.

[2] I. Babuška, Accuracy Estimates and Adaptive Refinements in Finite Element Computations, John Wiley & Sons, 1986.

[3] L. Demkowicz, W. Rachowicz and P. Devloo, A fully automatic *hp*-adaptivity, J. Sci. Comput. 17(1-4) (2002), 117-142.

[4] R. E. Bank, A. H. Sherman and A. Weiser, Some refinement algorithms and data structures for regular local mesh refinement, Scientific Computing, Applications of Mathematics and Computing to the Physical Sciences 1 (1983), 3-17.

[5] S. Wille, A structured tri-tree search method for generation of optimal unstructured finite element grids in two and three dimensions, International Journal for Numerical Methods in Fluids 14(7) (1992), 861-881.

[6] P. Solin, K. Segeth and I. Dolezel, Higher-order Finite Element Methods, CRC Press, 2004.

[7] G. F. Carey, M. Sharma and K. C. Wang, A class of data structures for 2-D and 3-D adaptive mesh refinement, International Journal for Numerical Methods in Engineering 26(12) (1988), 2607-2622.

[8] H. Samet, The quadtree and related hierarchical data structures, ACM Computing Surveys (CSUR) 16(2) (1984), 187-260.

[9] M.-C. Rivara, Local modification of meshes for adaptive and/or multigrid finite-element methods, J. Comput. Appl. Math. 36(1) (1991), 79-89.

[10] M. Ainsworth and B. Senior, *hp*-finite Element Procedures on Non-uniform Geometric Meshes: Adaptivity and Constrained Approximation, Grid Generation and Adaptive Algorithms, Springer, 1999, pp. 1-27.

[11] L. Demkowicz, Computing with *hp*-adaptive Finite Elements: Volume 1 One and Two Dimensional Elliptic and Maxwell Problems, Chapman & Hall/CRC, Boca Raton, 2006.

[12] P. Šolın and L. Demkowicz, Goal-oriented *hp*-adaptivity for elliptic problems, Comput. Methods Appl. Mech. Engrg. 193(6) (2004), 449-468.

[13] A. Gupta, A finite element for transition from a fine to a coarse grid, International Journal for Numerical Methods in Engineering 12(1) (1978), 35-45.

[14] C.-K. Choi and N.-H. Lee, Three dimensional transition solid elements for adaptive mesh gradation, Structural Engineering and Mechanics 1(1) (1993), 61-74.

[15] Y.-S. Cho and S. Im, MLS-based variable-node elements compatible with quadratic interpolation, Part I: formulation and application for non-matching meshes, International Journal for Numerical Methods in Engineering 65(4) (2006), 494-516.

[16] A. Tabarraei and N. Sukumar, Adaptive computations on conforming quadtree meshes, Finite Elem. Anal. Des. 41(7) (2005), 686-702.

[17] N. Sukumar and E. A. Malsch, Recent advances in the construction of polygonal finite element interpolants, Arch. Comput. Methods Engrg. 13(1) (2006), 129-163.

[18] X.-H. Tang, S.-C. Wu, C. Zheng and J.-H. Zhang, A novel virtual node method for polygonal elements, Appl. Math. Mech. 30(10) (2009), 1233-1246.

[19] N. Sukumar, Quadratic maximum-entropy serendipity shape functions for arbitrary planar polygons, Comput. Methods Appl. Mech. Engrg. 263 (2013), 27-41.

[20] P. Milbradt and T. Pick, Polytope finite elements, International Journal for Numerical Methods in Engineering 73(12) (2008), 1811-1835.

[21] A. Rand, A. Gillette and C. Bajaj, Quadratic serendipity finite elements on polygons using generalized barycentric coordinates, 2011, arXiv preprint arXiv: 1109.3259

[22] S. Natarajan, S. Bordas and D. Roy Mahapatra, Numerical integration over arbitrary polygonal domains based on Schwarz-Christoffel conformal mapping, International Journal for Numerical Methods in Engineering 80(1) (2009), 103-134.

[23] S. Mousavi, H. Xiao and N. Sukumar, Generalized Gaussian quadrature rules on arbitrary polygons, International Journal for Numerical Methods in Engineering 82(1) (2010), 99-113.

[24] J. Oden, C. Duarte and O. Zienkiewicz, A new cloud-based *hp* finite element method, Comput. Methods Appl. Mech. Engrg. 153(1) (1998), 117-126.

[25] C. Duarte, I. Babuška and J. Oden, Generalized finite element methods for three-dimensional structural mechanics problems, Comput. & Structures 77(2) (2000), 215-232.

[26] T. Strouboulis, K. Copps and I. Babuška, The generalized finite element method, Comput. Methods Appl. Mech. Engrg. 190(32) (2001), 4081-4193.

[27] J. M. Melenk and I. Babuška, The partition of unity finite element method: basic theory and applications, Comput. Methods Appl. Mech. Engrg. 139(1) (1996), 289-314.

[28] S. Rajendran and B. Zhang, A "FE-meshfree" QUAD4 element based on partition of unity, Comput. Methods Appl. Mech. Engrg. 197(1) (2007), 128-147.

[29] Y. Yang, X. Tang and H. Zheng, A three-node triangular element with continuous nodal stress, Comput. & Structures 141 (2014), 46-58.

[30] X. Zhao, S. Mao and Z. Shi, Adaptive finite element methods on quadrilateral meshes without hanging nodes, SIAM J. Sci. Comput. 32(4) (2010), 2099-2120.

[31] A. Plaza and G. F. Carey, Local refinement of simplicial grids based on the skeleton, Appl. Numer. Math. 32(2) (2000), 195-218.

[32] O. C. Zienkiewicz and R. L. Taylor, The Finite Element Method for Solid and Structural Mechanics, Butterworth-Heinemann, 2005.

[33] P. Heintz and K. Samuelsson, On adaptive strategies and error control in fracture mechanics, Comput. & Structures 82(6) (2004), 485-497.

[34] A. Byfut and A. Schröder, *hp*-adaptive extended finite element method, International Journal for Numerical Methods in Engineering 89(11) (2012), 1392-1418.

[35] I. Babuška, K. Copps, S. K. Gangaraj and C. S. Upadhyay, A-posteriori Error Estimation for Finite Element and Generalized Finite Element Method, The University of Texas at Austin, 1998.

[36] W. F. Mitchell, The *hp*-multigrid method applied to *hp*-adaptive refinement of triangular grids, Numer. Linear Algebra Appl. 17(2-3) (2010), 211-228.