TẠP CHÍ KHOA HỌC & CÔNG NGHỆ ĐẠI HỌC DUY TÂN
DTU Journal of Science and Technology 5(54) (2022) 50-57

TRƯỜNG ĐẠI HỌC DUY TÂN
DUYTAN UNIVERSITY

h-adaptive refinement strategies for triangular finite element meshes

Các thuật toán làm mịn h-thích nghi cho lưới phần tử hữu hạn tam giác

Nguyen Trung Hieu[a,b,*]
Nguyễn Trung Hiếu[a,b,*]

[a] *Institute of Research and Development, Duy Tan University, Da Nang, 550000, Vietnam*
[a] *Viện Nghiên cứu và Phát triển Công nghệ Cao, Trường Đại học Duy Tân, Đà Nẵng, Việt Nam*
[b] *Faculty of Natural Sciences, Duy Tan University, Da Nang, 550000, Vietnam*
[b] *Khoa Khoa học Tự nhiên, Trường Đại học Duy Tân, Đà Nẵng, Việt Nam*

(Ngày nhận bài: 09/05/2022, ngày phản biện xong: 11/05/2022, ngày chấp nhận đăng: 15/08/2022)

Abstract

In this paper, we present two different approaches to h-adaptively refine triangular finite element messes. These two strategies are designed to keep the shape regularity of the meshes almost the same and to preserve the sparsity pattern of the resulting system of equations.

Keywords: h-adaptive finite elements; red-green refinement; longest edge bisection

Tóm tắt

Trong bài báo này, chúng tôi giới thiệu hai cách làm mịn thích nghi lưới phần tử hữu hạn dạng tam giác. Hai cách tiếp cận này được thiết kế để giữ độ chuẩn hóa hình dạng của các lưới gần như không đổi và bảo toàn cấu trúc thưa của các hệ phương trình cần giải.

Từ khóa: Phần tử hữu hạn loại h; chia lưới xanh đỏ; chia đôi theo cạnh dài nhất.

1. Introduction

The finite element method (FEM) is a popular method for solving partial differential equations (PDEs) [1, 2]. In FEM, the physical domain of the PDE is split into a finite number of elements. Together, they form a computational mesh. For problems whose solutions change rapidly, for example, those with singularities or sharp fronts, adaptive meshes are a must to provide good accuracy. This kind of meshes is built gradually using adaptive refinement [3, 4]. In this paper, we will present two h-refinement strategies.

In h-refinement, one refines an element in two or more children elements of smaller sizes while keeping the degree of the new elements (degrees of the basis functions associated with these elements) the same as their father's.

Based on the range of influence, h-refinements can be categorized in three different strategies: global refinement, semi-global refine-

* *Corresponding Author*: Nguyen Trung Hieu; Institute of Research and Development, Duy Tan University, Da Nang, 550000, Vietnam; Faculty of Natural Sciences, Duy Tan University, Da Nang, 550000, Vietnam.
Email: nguyentrunghieu14@duytan.edu.vn

ment, and local refinement.

In *global mesh refinement*, every element in the mesh is refined (usually in the same way) to obtain a finer mesh. Clearly, this is the simplest strategy to implement. However, it is also the most expensive strategy since many elements are generated away from areas of interest. Sometimes, global mesh refinement is referred to as uniform refinement.

A variation of global mesh refinement is *semi-global mesh refinement*, in which elements in one or more selected cross-sections of the mesh are refined. In certain cases, this strategy maybe implemented as easily as global refinement and may be less wasteful. Nevertheless, this strategy does not always work and is still considered uneconomical.

In the rest of this paper, we will review two different approaches of (adaptive) *local mesh refinement*, in which only a selected group of elements is refined. This is a very attractive strategy especially for problems with singularities or sharp fronts since the refinement can be restricted to those portions of the domain where it is needed.

Requirement 1.1. *For local mesh refinement to be efficient, it is necessary that:*

(i) elements to be refined can be determined cheaply

(ii) the sparsity of the resulting systems of linear equations is preserved as the mesh is refined.

(iii) the adaptive local mesh refinement procedure can be implemented cheaply

In this paper, we assume (i) and focus only on (ii) and (iii).

2. Red-Green Mesh Refinement

In this section, we present the work of Bank, Sherman and Weiser in [5].

In h-adaptive meshing, to preserve the quality of the current mesh, one can use *red refinement*[1], which is sometimes called *bisection-type* mesh refinement. In this type of refinement, a triangular element t is subdivided into four triangles called *sons* of t, by pairwise connecting the midpoints of the three edges of t. Figure 1 (a) and (b) illustrate an element t and its children after a red refinement.

Obviously, in red refinement, the children elements are geometrically similar to their farther. Therefore, they have the same shape regularity quality as their father. This is an advantage of this type of refinement. However, new vertices introduced in red refinement usually break the conformity of the triangulation. This can be seen from Figure 1 (c) where an element t is red refined several times. In the figure, except for vertices of t, all other vertices are non-conforming.

These non-conforming vertices are usually called irregular vertices and are rigorously defined as follows.

Definition 2.1. *A vertex is said to be regular if it is a corner of each element it touches. A vertex is said to be irregular if it is not regular.*

Definition 2.2. *The irregular index of a mesh is the maximum number of irregular vertices on a side of any element in the mesh. A k-irregular mesh is a mesh with irregular index k*

Figure 2 (a) shows an example of a 2-irregular mesh.

Remark 2.3. *Note that all boundary vertices should be regular.*

In general, it is advantageous to "regularize" a mesh by restricting the number of irregular vertices on each edge. There are several reasons for that: simplifying computations such as matrix assembly and mesh refinement, increasing approximation power by ensuring that neighboring elements are not too different in sizes, and guaranteeing that each element is in the support of a bounded number of basis functions. There

[1] The name "red refinement" came after the name "green refinement" which is discussed later.

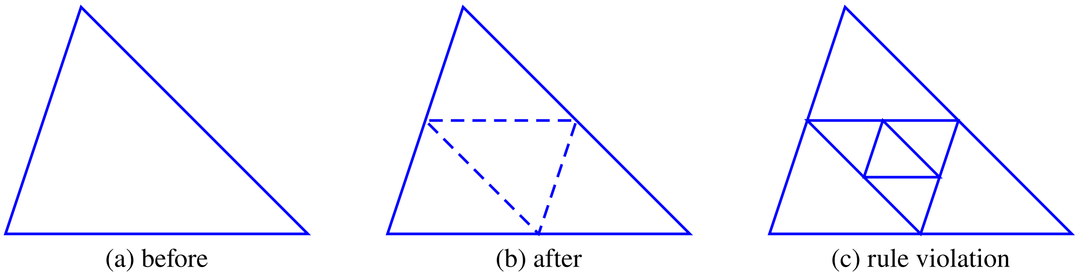


Figure 1. Red refinement: before, after, and 1-irregular rule violation.

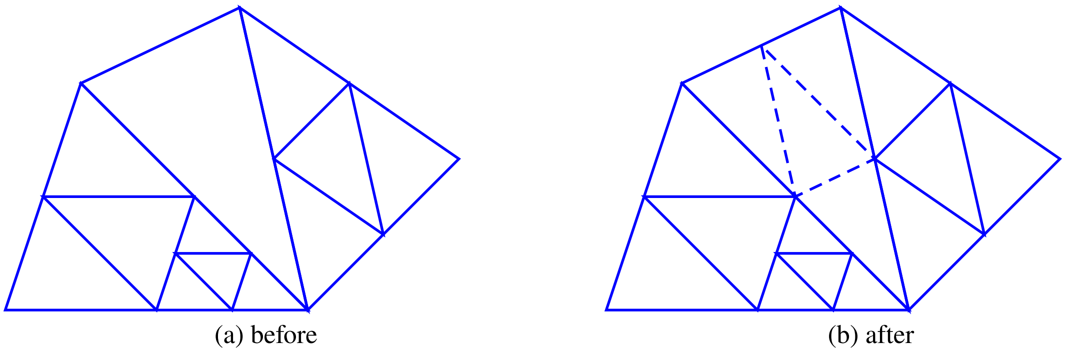


Figure 2. A mesh before and after fixing rule violations.

are several ways to achieve this regularization. In [5], Bank et al. suggested using the following *1-irregular rule* and some of its variants.

Rule 2.4. 1-Irregular Rule: *Keeping the number of irregular vertices on any edge of any element in the triangulation be at most one. In other words, refine any element for which any of its edges contains more than one irregular vertex.*

Figure 2 (a) can also serve as an example of a mesh with a violation of 1-irregular rule. The mesh after fixing the violation is shown in Figure 2 (b).

In order to monitor the number of irregular vertices on an edge of an element, one could use the information of its level and neighbors.

Definition 2.5. *The level ℓ_{t_i} of an element t_i is defined inductively as follows*

$$\ell_{t_i} = \begin{cases} 1 & \text{if } t_i \in \mathscr{T}_0 \\ \ell_{t_f} + 1 & \text{if } t_i \notin \mathscr{T}_0 \end{cases},$$

where t_f is the father of t_i and $\mathscr{T}_0$ is the geometrically admissible initial mesh.

Definition 2.6. *The neighbor t_i^j of element t_i across its jth edge e_i^j is the smallest element with one edge completely overlapping e_i^j.*

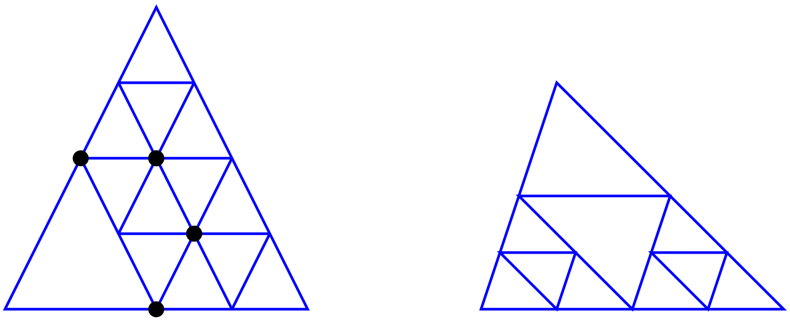
Clearly, the number of irregular vertices on an edge of an element is related to the difference of its level and the level of one of its neighbors across that edge.

Let $\mathscr{T}$ be a geometrically admissible mesh. Assume that some elements in $\mathscr{T}$ are selected to be red-refined owing to, for example, having large errors. These refinements, in turn, introduce some irregular vertices. During the refinement process, 1-irregular rule is applied as often as possible to accomplish a regularized mesh which, according to [5], has the following properties:

Proposition 2.7. *Let $\mathscr{T}'$ be the mesh obtained from $\mathscr{T}$ after some red refinements, and regularization using 1-irregular rule. Then*

(i) $\mathscr{T}'$ has irregular index 1.

(ii) $\mathscr{T}'$ uniquely contains the fewest elements of any 1-irregular mesh that can be obtained by refining $\mathscr{T}$.



(a) Nonzero basis functions in t (b) 2-neighbor rule violation

Figure 3. Nonzero basis functions (left) and 2-neighbor rule violation (right).

(iii) $|\mathcal{T}| \leq 13|\mathcal{T}'|$.

Remark 2.8. *The property (iii) of proposition (2.7) is usually pessimistic (see remark 2.11).*

Besides the nice properties above, $\mathcal{T}'$ is still not a geometrically admissible mesh owing to the presence of irregular vertices. In addition, a triangulation of irregular index 1 does not guarantee that the number of nonzero basis functions[2] in each element are exactly three. An example is illustrated in Figure 3, where the triangulation satisfies 1-irregular rule , but the four basis functions corresponding to the vertices marked by dots are nonzero in t.

To fix these issues, Bank et al. proposed using *green refinement*[3], in which a vertex is connected to the midpoint of the opposite edge of the element we want to refine (see Figure 4 (a)). The use of green refinement is determined by the *green rule* described as follows

Rule 2.9. *Green Rule: With as few elements as possible, green refine any element with an irregular vertex on one or more of its edges.*

For 1-irregular meshes, there are three cases in which green rule can be applied. These cases are shown in Figure 5.

Proposition 2.10. *Let $\mathcal{T}'$ be an 1-irregular mesh, for example, the resulting mesh in Proportion 2.7. Assume that $\mathcal{T}''$ is generated from $\mathcal{T}'$ by applying the green rule wherever possible. Then the following hold:*

(i) For any element t'' in $\mathcal{T}''$, there are at most three basis functions having supports in it. In addition, the restrictions of these basis functions in t'' are linearly independent.

(ii) In $\mathcal{T}''$, the support of a basis function intersects with those of at most twelve other basis functions.

(iii) $|\mathcal{T}''| \leq 2|\mathcal{T}'|$.

Remark 2.11. *The properties (i) and (iii) of proposition 2.10 are usually pessimistic. The most common number of non-zeros in a row of the stiffness matrix is seven, and for most meshes encountered in practice, $\mathcal{T}''$ contains fewer than twice as many elements as $\mathcal{T}$. Here, $\mathcal{T}'$ is obtained from $\mathcal{T}$ after some red refinement and 1-irregular regularization.*

In addition, one could use a more aggressive refinement strategy by applying, in conjunction with 1-irregular rule and green rule, the following 2-neighbor rule.

Rule 2.12. *2-Neighbor Rule: Red refine any element t with two neighbors that have been red refined.*

An example of a mesh with a violation of 2-neighbor rule is shown in Figure 3 (b).

When the 2-neighbor rule is used together with the 1-irregular rule, one gets an 1-irregular mesh in which each remaining irregular vertices is located at the midpoint of an edge of a unique

[2]Here we only consider linear basis functions.

[3]The name "green refinement" came from graph theory, where sometimes special edges are distinguished by color.

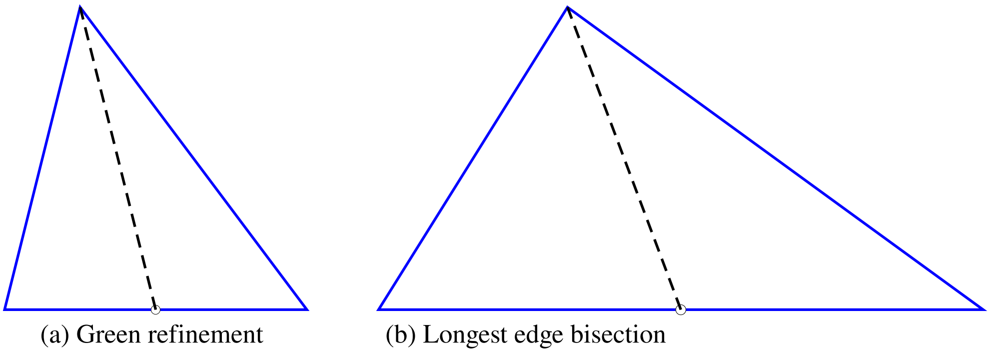


Figure 4. Refinement strategies.

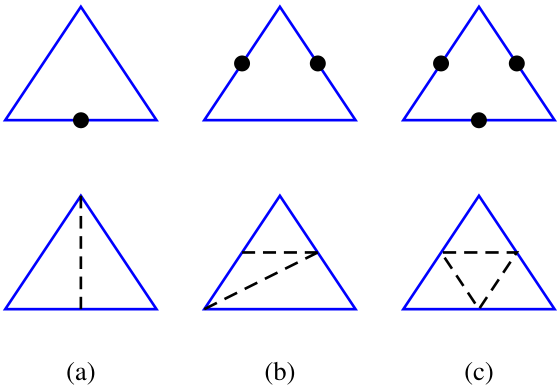


Figure 5. Cases when Green rule can be applied.

element. This implies that for such a mesh, only the case in Figure 5 (a) occurs when green rule is applied. For the resulting mesh, analogs of properties (ii) and (iii) in proposition 2.10 hold, but the constants are usually bigger.

Algorithm 1 is an algorithm implementing 1-irregular rule in conjunction with 2-neighbor rule.

Here we assume that a Boolean-valued function $DVTEST$, which decides whether an element should be refined, is available. Usually $DVTEST$ is the output of a self-adaptive mechanism within the code that uses local error indicators. Sometimes, $DVTEST$ can be a user specification of a fixed refinement pattern. An element in the mesh may be refined either because $DVTEST$ indicates it should be refined, or because it violates the 1-irregular rule or 2-neighbor rule. It is also possible that an element satisfies both rules at the beginning but violates one of them later in the refinement process owing to the refinement of one of its neighbors. This implies that an element can be examined by the algorithm more than once.

Note that in Algorithm 1, elements are processed in the order they are created, newly created elements are placed at the end of the working list. In particular, when an element is examined, its neighbors are tested against 1-irregular and 2-neighbor rules before it is checked by $DVTEST$. This guarantees that these rules are satisfied by meshes generated by Algorithm 1, and that remaining irregular vertices are the sole edge midpoints in some elements.

Since a given element has at most three neighbors, we test (and possibly refine) at most four elements at any step of Algorithm 1. Hence, the complexity of Algorithm 1 is linear in the number of elements.

3. Longest Edge Bisection

In this section we discuss *longest edge bisection*, another approach of h-adaptive meshing.

Algorithm 1 Local Meshing Procedure For Red Refinement

Procedure REFINE
  while $(i \le nt)$ **do**
    for $j = 1$ **to** 3 **do**
      if t_i^j *is unrefined* **then**
        if t_i^j *has more than one neighbor* **or** $\ell_i > \ell_{t_i^j} + 1$ **then**
          $DIVIDE(t_i^j)$;
        end if
      end if
    end for
    if $DVTEST(t_i)$ **then**
      $DIVIDE(t_i)$;
    end if
    $i \leftarrow i + 1$;
  end while
End
Procedure DIVIDE
  $s_i \leftarrow nt + 1$; $nt \leftarrow nt + 4$;
  for $j = 0$ **to** 3 **do**
    create t_{s_i+j};
  end for
End

In longest edge bisection, an element is refined into two smaller elements by connecting the midpoint of its longest edge with the opposite vertex as shown in Figure 4 (b).

Obviously, one chooses to bisect the longest edge to maintain the shape regularity quality of the mesh. In an element, the angle opposite to the longest edge is the biggest one. Therefore, refinement by dividing that angle would reduce the chance to have elements with small angles. However, bisecting an element introduces an irregular (nonconforming) vertex. This leads to further refinement. The question is whether the process terminates in finite steps and whether the resulting mesh has some control of the smallest angles.

The following theorem on "a lower bound on the angles of triangulation constructed by bisecting the longest edge" was given by Rosenberg and Stenger in 1975.[4]

Theorem 3.1. *Let α_0 be the smallest interior angle of $\mathscr{T}_0$, a given initial geometrically admissible triangulation. If α_j is the smallest angle of the triangulation $\mathscr{T}_j$ obtained by the jth iterative bisection of all the triangles generated from $\mathscr{T}_0$, then $\alpha_j \ge \alpha_0$, for all j.*

Later, in 1984, Rivara introduced several algorithms using longest edge bisection and gave a proof of their finiteness. The following is the simplest version of her algorithms for local refinement discussed in [7]. Figure 6 shows an example of using longest edge bisection algorithm, in which newly created edges are labeled in the order they are created.

Remark 3.2. *Even though theorem 3.1 guarantees a lower bound on the angles of triangulation constructed by bisecting the longest edge, the shape regularity quality of the mesh could be reduced significantly after several refinements. To*

[4]The original result of Rosenberg and Stenger in [6] was stated slightly different. Here we use the version of Rivara used in [7].

Algorithm 2 Local Mesh Refinement Using Longest Edge Bisection

For each $t \in \mathscr{S}_0$*, bisect* t *by its longest edge.*
$k \leftarrow 1$;
while $I_k \neq \emptyset$ **do**
  for $t \in I_k$ *with irregular vertex* P **do**
    Bisect t by its longest edge.
    if P *is not on the longest edge* **then**
      Join P *with the midpoint of the longest edge of* t.
    end if
  end for
  $k \leftarrow k+1$;
end while

Here S_0 *is the set of elements to be refined and* I_k *is the set of elements with irregular vertices at step* k*.*

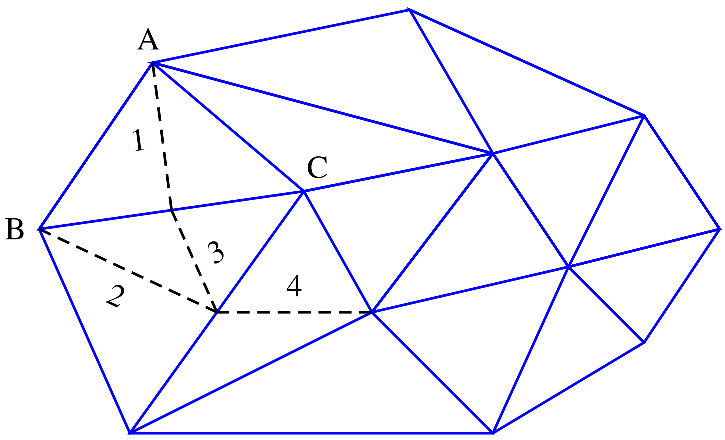


Figure 6. Local application of Algorithm 2 for refining element ABC.

deal with this issue, we apply the mesh smoothing technique, edge flipping, etc., to improve the quality of the mesh generated from h-adaptive meshing.

4. Conclusion

In this paper, we have presented two common approaches to adaptively refine triangular finite element meshes. These approaches are attractive in the sense that they preserve the shape regularity of meshes and keep the sparsity pattern of resulting system of linear equation almost the same. These qualities are essential to have a robust h-refinement algorithms.

References

[1] Barna Szabó and Ivo Babuška. *Finite element analysis*. A Wiley-Interscience Publication. John Wiley & Sons Inc., New York, 1991.

[2] Susanne C. Brenner and L. Ridgway Scott. *The mathematical theory of finite element methods*, volume 15 of *Texts in Applied Mathematics*. Springer, New York, third edition, 2008.

[3] Hieu Nguyen. *p- and fully automatic hp- adaptive finite element methods for elliptic Partial Differential Equations methods*. PhD thesis, University of California, San Diego, 2010.

[4] W. Gui and I. Babuška. The h, p and h-p versions of the finite element method in 1 dimension. *Parts 1, 2, 3, Numerische Mathematik*, 49(6):577–683, November 1986.

[5] Randolph E. Bank, Andrew H. Sherman, and Alan Weiser. Refinement algorithms and data structures for regular local mesh refinement. In *Scientific computing (Montreal, Que., 1982)*, IMACS Trans. Sci. Comput., I, pages 3–17. IMACS, New Brunswick, NJ, 1983.

[6] Ivo G. Rosenberg and Frank Stenger. A lower bound on the angles of triangles constructed by bisecting the longest side. *Math. Comp.*, 29:390–395, 1975.

[7] M.-Cecilia Rivara. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. *Internat. J. Numer. Methods Engrg.*, 20(4):745–756, 1984.