TẠP CHÍ KHOA HỌC & CÔNG NGHỆ ĐẠI HỌC DUY TÂN
DTU Journal of Science and Technology 5(54) (2022) 105-109

TRƯỜNG ĐẠI HỌC DUY TÂN
DUYTAN UNIVERSITY

p-adaptive meshing for triangular nodal finite elements meshes

Các thuật toán thích nghi loại p 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: 16/05/2022, ngày phản biện xong: 26/05/2022, ngày chấp nhận đăng: 30/08/2022)

Abstract

In this paper, we present p-adaptive meshing algorithms (p-refinement and unrefinement) for triangular finite element meshes. The algorithms are designed to keep the computation cost low, and the neighboring elements are not too different in degree. In addition, we can prove that if we start with an admissible mesh, then each element can be refined at most once using the proposed algorithms. This is important because after a p-refinement there is no way to estimate its error without resolving the whole problem.

Keywords: Finite elements; p-adaptive finite elements; p-adaptive meshing

Tóm tắt

Trong bài báo này, chúng tôi giới thiệu các thuật toán thích nghi dạng p (tăng và giảm bậc xấp xỉ) cho lưới phần tử hữu hạn dạng tam giác. Các thuật toán này được thiết kế để chi phí không quá cao, các phần tử liền kề không có bậc xấp xỉ quá chênh lệch. Ngoài ra, chúng tôi còn chứng minh được rằng: nếu bắt đầu bằng một lưới được chấp nhận thì mỗi phần tử trong lưới sẽ chỉ bị tăng bậc xấp xỉ nhiều nhất một lần khi sử dụng các thuật toán được đề xuất. Điều này có ý nghĩa rất quan trọng vì sau khi tăng bậc xấp xỉ của một phần tử, chúng ta không có cách nào ước lượng sai số nếu như không giải lại toàn bộ bài toán.

Từ khóa: Phần tử hữu hạn; phần tử hữu hạn loại p; lưới thích nghi loại p

1. Introduction

The adaptive finite element method (AFEM) is an efficient method for solving partial differential equations (PDEs) [1, 2]. In addition to the traditional h-version of AFEM [3, 4], there is also the p-version using p-adaptive meshing [5, 6, 7, 8, 9, 3, 10, 11].

In p-adaptive meshing, the geometry of mesh is fixed and a better solution approximation is obtained by modifying (usually increasing) the degrees of the elements in the mesh. This is a different approach to get better approximation spaces. In h-refinement, elements of small sizes are used, while in p-refinement, elements with

*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

higher degrees are utilized.

Since the geometry of a mesh is unchanged after a p-refinement, we assume, in this paper, that all the meshes we are working with are geometrically admissible. In addition, in this paper, when we talk about a mesh we refer to its both topology and the set of degrees of freedom associated with elements in the mesh.

Similar to h-refinement, there are global and local p-refinements.

In global p-refinement, also called uniform p-refinement, the degree of every element in the mesh is increased by the same quantity. This strategy of refinement appears to be very effective for a class of problems where the exact solutions are very smooth and can be well approximated by polynomials.

In practice, however, we usually encounter problems with singularities, sharp fronts, or rapid change in part of the solutions. For these classes of problem, using uniform p-refinement is very expensive since many elements in critical areas also use high degrees. Because of this reason, local p-refinement, where only selected elements are refined, is very attractive.

The rest of this paper is devoted to local p-refinement and other local p-adaptive meshing.

2. Refinement Rules

For convenience, in this section, "refine" is understood as "p-refine", and refinement as p-refinement unless otherwise specified. Also for now, we restrict our discussion to cases in which the degree of an element is increased by one in a p-refinement.

We assume that we have a way to determine what elements should be refined cheaply. We focus only on developing an efficient algorithm for p-refinement. The followings are the goals we want to achieve when building such algorithm.

Requirement 2.1. *For local p-refinement to be efficient, it is necessary that:*

(i) Computation is not expensive.

(ii) Neighboring elements are not too different in degree.

(iii) The algorithm can be implemented cheaply.

The requirements (i) and (iii) are natural for any practical algorithm. For p-refinement, we can achieve (i) by ensuring that each element is in the support of a bounded number of basis functions. In doing so, we could preserve the sparsity of the resulting system of linear equation as the mesh is refined. In addition, we could also limit the number of special cases. Not only does this help to simplify computation, but it also makes the algorithm simpler to implement.

As of requirement (ii), the purposes are to smooth the changes of approximated solutions from element to element, and to increase approximation power.

To achieve the goals in Requirement 2.1 we propose using the following rules.

Rule 2.2. *1-Irregular Rule (p-version): The difference in degree of neighboring elements can be at most one. Refine any element t of degree p with a neighbor of degree higher than $p+1$.*

Rule 2.3. *2-Neighbor Rule (p-version): For any element, there should be no more than one neighbor of higher degree. Refine any element t of degree p with two or three neighbors of degree $p+1$.*

Remark 2.4. *These rules are inspired by those of the same names in Red-Green (h-) refinement. The only difference is that here the degrees of elements play the role of their levels.*

Definition 2.5. *A mesh is said to be admissible (in degree) if there is no violation of 1-irregular rule and 2-neighbor rule.*

Remark 2.6. *Obviously, if a mesh is admissible, then there is only one special case in which an element of degree p has one and just one neighbor of degree $p+1$ (the other neighbors if exist are of degree p). Such element is called transition element and their basis functions are especially defined in [11].*

Now there is only one concern left. As far as we know, we cannot recalculate error estimates for an element that has been p-refined before the whole problem is resolved. Therefore, unlike

h-refinement, in p-refinement we can only refine an element at most once. The question is whether this is possible when we have both refinement owing to large error estimates, and refinement owing to violations of 1-irregular rule and 2-neighbor rule. The following theorem answers this question.

Theorem 2.7. *If we start with an admissible mesh, and no element is required to refine by error more than once, then in a refinement, with enforcement of 1-irregular rule and 2-neighbor rule, each element can be refined at most once.*

Proof. This is proof by contradiction.
Let t_i be the first element in the mesh to be refined twice. Assume t_i is of degree p right before its second refinement, and therefore of degree $p-1$ in the starting mesh. Now we consider two cases:

Case 1: Violation of 1-irregular rule: Suppose the second refinement of t_i is caused by a refinement of its neighbor t_j. This implies that t_j is of degree $p+2$ or higher right before the second refinement of t_i (see Figure 1 on the right). Since t_i is the first element to be refined twice, the refinement of t_j is its first refinement. Therefore, in the starting mesh, the degree of t_j is at least $p+1$. Hence the violation of 1-irregular rule between t_i and t_j in the starting mesh (see Figure 1 on the left). This contradicts the assumption that the starting mesh is admissible.

Case 2: Violation of 2-neighbor rule: Suppose the second refinement of t_i is caused by refinements of its neighbors t_j and t_k. This implies that t_j and t_k is of degree $p+1$ right before the second refinement of t_i (see Figure 2 on the right). Since t_i is the first element to be refined twice, the refinements of t_j and t_k are their first refinements. Therefore, in the starting mesh, t_j and t_k are of degree p. Hence, there is a violation of 2-neighbor rule between t_i, t_j and t_k in the starting mesh (see Figure 2 on the left). This contradicts with the assumption that the starting mesh is admissible. □

2.1. p-Adaptive Refinement

The p-adaptive refinement algorithm we are using consists of four phases, namely, *marking*, *laying out new dofs*, *interpolating*, and *updating mesh status*.

Phase 1 - Marking: A posteriori error estimates are calculated based on the current solution on the current mesh. Then, all elements are placed in a max-heap data structure according to the size of their error estimates. The element with the largest error estimate is at the top of the heap. This element is marked for refinement and put in a list. Elements in the list is examined one by one (first in first out) until the list is empty. If refinement of an element in the list causes a violation of 1-irregular rule or 2-neighbor rule, and requires refinement of one of its neighboring element which has not been marked for refinement, then that neighboring element is marked for refinement and put at the end of the list. When the list is empty, the mesh is admissible (having no rule violation).

Phase 2 - Laying out dofs: In this phase, we go over every element in the mesh and redefine their dof pointers.

Phase 3 - Interpolating: Since dofs (nodal points) of an element are changed after a p-refinement, the values of grid function associated with new dofs need to be computed in order to provide a good initial guess for the iterative solver used to solve the system of linear equations.

We seek the finite element solution as a linear combination of basis functions:

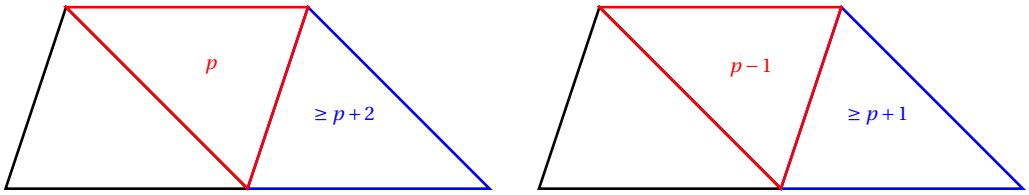
$$f_{f.e} = \sum_{i=1}^{N} c_i \, \phi_i \tag{1}$$

Here the c_i are approximated values of the exact solution at the places of dofs in the mesh. These c_i are actually the values stored in the grid function array GF.

Of course, the identity 1 still holds if it is restricted on an element t. Also note that only basis functions associated with dofs of t have supports in t. Therefore, we can write 1 as

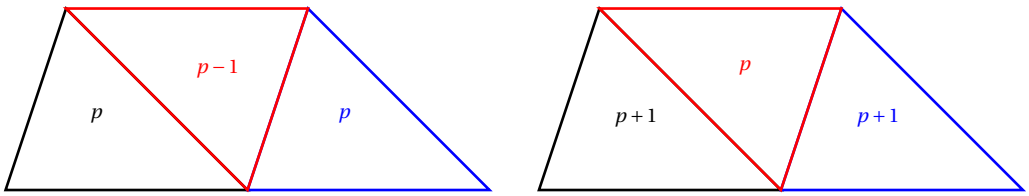
$$f_{f.e}|_t = \sum_{i=1}^{N_p} c_{n_i} \, \phi_{n_i}^{(p)}|_t, \tag{2}$$

where p is the degree of t. If t is p-refined then



(a) Before 2nd refinement of t_i (b) Starting Mesh

Figure 1. Second refinement of t_i caused by a violation of 1-irregular rule.



(a) Before 2nd refinement of t_i (b) Starting Mesh

Figure 2. Second refinement of t_i caused by 2-neighbor rule.

Algorithm 1 p-Refinement: Phase 1 - Marking

Phase 1: Marking
Calculate error estimates.
Build a max-heap H of elements according to their error estimates.
$first \leftarrow 1$; $last \leftarrow 1$;
while $ndf < ndftgrt$ **do**
  $L(first) \leftarrow H(1)$;
  repeat
    $t \leftarrow L(first)$
    Mark t as a potential element for refinement
    Update ndf
    for $j = 1$ **to** 3 **do**
      if *refinement of t required its neighbor* t^j *to be refined* **then**
        $last \leftarrow last + 1$;
        $L(last) = t^j$;
      end if
    end for
    $first \leftarrow first + 1$;
  until $first = last$
  Set errors of marked element to be zeros and update the heap
end while
End

we would want to find coefficients $\tilde{c}_{m_i}$ such that

$$\sum_{i=1}^{N_p} c_{n_i} \phi_{n_i}^{(p)}|_t = \sum_{i=1}^{N_{p+1}} \tilde{c}_{m_i} \phi_{m_i}^{(p+1)}|_t \qquad (3)$$

(Note that, for the sake of simplicity, we assume t is a regular element before and after the p-refinement.) Clearly we can compute the values of $\tilde{c}_{m_i}$ by evaluating the left hand side of equation 3 at the dof m_i.

Phase 4 - Updating Mesh Status: In this phase, using the record of elements marked for refinement and the previous status of the mesh, the new degree and especially new transition status of every element are updated.

2.2. *p*-Adaptive Unrefinement

In a crude way, one could apply Algorithm 1 (of p-adaptive refinement) to marking elements to p-unrefined with only one change: constructing H as a min-heap of errors of elements instead of a max-heap of them. With this approach, elements with small errors are marked for p-refining first. However, comparing approximate quantities of small scale is not very reliable. To avoid this problem, we currently use the same max-heap as in p-refinement. The only difference is that the "p-refinement" is carried on a reduced mesh where degree of elements, except linear ones, are reduced by one. The elements which are not marked in the "p-refinement" are the ones marked for p-unrefinement.

The other phases of p-unrefinement are very much identical to those of p-refinement except for small changes in implementation to reduce degrees of elements instead of increasing them.

3. Conclusion

In this paper, we have presented p-adaptive refinement and p-adaptive unrefinement algorithms for triangular finite element meshes. These algorithms are designed to that keep the computation cost low, and the neighboring elements are not too different in degree. In addition, we can prove that if we start with an admissible mesh, then each element can be refined at most once using the proposed algorithms. These qualities are essential to have an efficient and robust adaptive meshing algorithms.

References

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

[2] 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.

[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] Hieu Nguyen. *h*-adaptive refinement strategies for triangular finite element meshes. *DTU Journal of Science & Technolog*, to appear.

[5] I. Babuška, EP Stephan, and B.Q. Guo. The hp version of the boundary element method with geometric mesh on polygonal domains. 1989.

[6] I. Babuška and M. Suri. The *h-p* version of the finite element method with quasi-uniform meshes. *RAIRO Modél. Math. Anal. Numér.*, 21(2):199–238, 1987.

[7] L. Demkowicz, J. T. Oden, W. Rachowicz, and O. Hardy. Toward a universal h-p adaptive finite element strategy, part 1. constrained approximation and data structure. *Computer Methods in Applied Mechanics and Engineering*, 77(1-2):79 – 112, 1989.

[8] B.Q. Guo and I. Babuška. The h-p version of the finite element method - part 1: The basic approximation results. *Computational Mechanics*, 1(1):21–41, 1986.

[9] B.Q. Guo and I. Babuška. The h-p version of the finite element method - part 2: General results and applications. *Computational Mechanics*, 1(3):203–220, 1986.

[10] Hieu Nguyen, Quoc Hung Phan, and Tina Mai. Nodal basis functions in p-adaptive finte element methods. *DTU Journal of Science & Technology*, 05(42):115–119, 2020.

[11] Hieu Nguyen, Quoc Hung Phan, Tina Mai, and Xuan Linh Tran. Transition nodal basis functions in p-adaptive finte element methods. *DTU Journal of Science & Technology*, 06(49):35–39, 2021.