



Adaptive finite element methods for elliptic equations over hierarchical T-meshes[☆]

Li Tian^a, Falai Chen^{b,1}, Qiang Du^{a,*}

^a Department of Mathematics, Pennsylvania State University, University Park, PA 16802, USA

^b Department of Mathematics, University of Science and Technology of China, Hefei, Anhui 230026, PR China

ARTICLE INFO

Keywords:

Numerical PDEs
Adaptive finite element
Adaptive PHT-splines
A posteriori error estimations
Hierarchical T-meshes

ABSTRACT

Isogeometric analysis using NURBS (Non-Uniform Rational B-Splines) as basis functions gives accurate representation of the geometry and the solution but it is not well suited for local refinement. In this paper, we use the polynomial splines over hierarchical T-meshes (PHT-splines) to construct basis functions which not only share the nice smoothness properties as the B-splines, but also allow us to effectively refine meshes locally. We develop a residual-based a posteriori error estimate for the finite element discretization of elliptic equations using PHT-splines basis functions and study their approximation properties. In addition, we conduct numerical experiments to verify the theory and to demonstrate the effectiveness of the error estimate and the high order approximations provided by the numerical solution.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

The use of B-splines and NURBS (Non-Uniform Rational B-Splines) as basis functions for finite element method has been well known for a long time. The method has been applied to the study of plate and shell bending [1–3], stress analysis [4,5], shape optimization [6,7] and so on. In [8], Frederickson first mentioned the possibility of using the so-called *box-splines* for the solution of partial differential equations through the Raleigh–Galerkin approach.

To accurately describe the geometry, NURBS require the use of a large number of superfluous control points in order to achieve local refinement. To overcome this limitation, the so-called *T-splines* were introduced [9] by allowing T-junctions in the control meshes, which provide adaptivity and flexibility in modeling geometric objects. With T-splines, local refinement of NURBS-based curve and surface representations is accomplished through the knot insertion, an exact process that does not alter the shape of the curve or surface; see [10–13] for the relevant algorithms. In [14], Döfel et al. introduced local h-refinement with T-splines. For all the above methods, in addition to the control points inserted for the refinement, additional control points are also needed (in order to preserve the geometry). As seen from the examples in [14,15], the T-splines are not as local as one would expect under the refinement. In earlier studies, rational basis functions were also used for finite element spaces to guarantee partition of unity. One can refer to [16] for a recent survey on the subject.

In [17], Deng et al. introduced a new type of polynomial splines over hierarchical T-meshes, the PHT-splines, which are piecewise bicubic polynomials over a hierarchical T-mesh. The new basis functions of PHT-splines share nice properties as those of B-splines, such as nonnegativity, local support and the partition of unity. PHT-splines can be viewed as a generalization of B-splines over hierarchical T-meshes which have two main virtues in comparison with T-splines: first, all

[☆] This work is supported in part by US NSF-DMS 0712744.

* Corresponding author.

E-mail addresses: tian@math.psu.edu (L. Tian), chenfl@ustc.edu.cn (F. Chen), qdu@math.psu.edu (Q. Du).

¹ This author is supported by the 973 Program 2011CB302400, the NSF of China (No. 60873109 and 11031007).

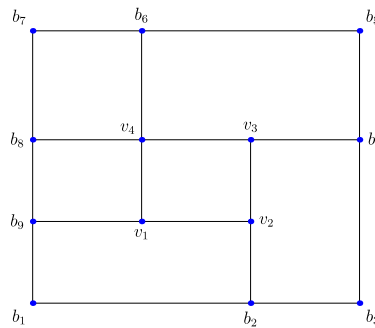


Fig. 1. An example of a T-mesh.

the basis functions are polynomials instead of rational functions, which not only make the computation more efficient but also allow analytical treatment within the conventional polynomial approximation and finite element analysis framework; second, local refinement can be effectively achieved by *cross insertion*, i.e., dividing a cell into four subcells with a cross point, with no additional points added, and the refinement is completely local. In addition, the conversion between NURBS and PHT-splines is very fast, while the conversion between NURBS and T-splines has been a bottleneck for T-splines in practical applications; see [17] for more discussions.

Given the flexibility of the T-meshes and PHT-splines, it is natural to consider the application of PHT-splines for the adaptive numerical solutions of PDEs, which is the main objective of the present work. The key steps taken here include the construction of the *a posteriori* error estimators and the adaptive refinement procedure for the T-meshes. In the numerical solution of partial differential equations (PDEs), *a posteriori* error estimates are computable quantities in terms of the approximate solutions which provide a *reliable* and *efficient* measurement for the errors of the discrete solution without the *a priori* knowledge of the exact solution. *A posteriori* error estimates have played important roles in adaptive mesh generation and algorithm design for numerical PDEs. Theoretical and systematical studies of *a posteriori* error estimators for finite element approximation began in the late 1970s [18] and have attracted a lot of attention; see [19–28] and the references cited therein. The use of T-mesh and T-splines for adaptive computations has been extensively discussed in [16]. And in [29], an adaptive mesh refinement over the so-called ‘Bogner–Fox–Schmidt’ (BFS) rectangular elements is applied to solve the Cahn–Hilliard equations. We note that the adaptive BFS elements and PHT-splines are closely related, which will be further discussed in the end of Section 3. In this paper, we develop a residual-based *a posteriori* error estimate for the finite element discretization of elliptic equations using PHT-splines basis functions, establish the approximation properties of the space of PHT-splines. Results of numerical experiments are then presented to verify the theory and demonstrate the robustness of the error estimate. The high order convergence observed from the numerical results illustrates the effective approximations provided by the adaptive splines. Though the initial study on the application of PHT-splines to the adaptive numerical solutions of PDEs carried out here is for a model two-dimensional linear second order elliptic equation, we expect greater advantages when PHT-splines are applied to solve more complex systems of equations (such as high order equations) defined on complex geometric domains (such as general Riemannian manifolds), which has been the motivation of the isogeometric analysis [14,30,16].

This paper is organized as follows: in Section 2, we introduce the PHT-splines; in Section 3, we describe the finite element discretization with PHT-splines; Section 4 presents the *a posteriori* error estimate, and both *reliability* and *efficiency* of the estimate are deduced; in Section 5, several numerical experiments are presented to support the theoretical predictions.

2. PHT-splines

Although the principle extends to higher dimensions, we focus on a two-dimensional rectangular domain in this study only. For non-rectangular domains, we can also apply a geometric map to get desired constructions [31]. Then, a *T-mesh* is basically a rectangular partition of the domain formed by grid lines parallel to the boundary of the domain that allows T-junctions (see [32,9] for more details). Each cell or facet in the mesh must be a rectangle. If a T-mesh is simply a rectangular grid with no T-junctions, the T-spline defined on it reduces to a B-spline.

A grid point formed by perpendicularly intersecting grid lines in a T-mesh is called a *vertex* of the T-mesh. If a vertex is on the boundary of the domain, then it is called a *boundary vertex*. Otherwise, it is called an *interior vertex*. Interior vertices have two types. One is *crossing*, and the other is *T-junctional*. They are called *crossing vertex* and *T-vertex* respectively. The line segment connecting two adjacent vertices on a grid line is called an *edge* of the T-mesh. Note that except for the vertices at the corners of the original domain, any vertex in the T-mesh must be on an end point to at least three different edges.

Fig. 1 provides as an example of a T-mesh: b_1, \dots, b_9 are boundary vertices; v_1, \dots, v_4 are interior vertices; v_1, v_2, v_3 are T-vertices and v_4 is a crossing vertex.

A *hierarchical T-mesh* is a special type of T-mesh which has a natural level structure. It is defined in a recursive fashion. One generally starts from a tensor-product mesh (level 0). From level k to level $k + 1$, one subdivides a cell at level k into

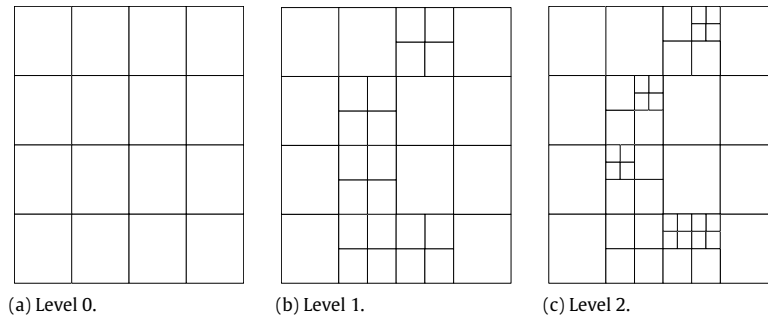


Fig. 2. Generation of a hierarchical T-mesh.

four subcells which are cells at level $k + 1$. For simplicity, we subdivide each cell by connecting the middle points of the opposite edges with two straight lines. Check Fig. 2 for an example of hierarchical T-mesh.

3. Finite element discretization

3.1. Model problem

Let $\Omega \in \mathbb{R}^2$ be a rectangular domain with boundary $\partial\Omega$. Consider the elliptic PDE with the homogeneous Dirichlet boundary condition

$$\begin{cases} -\nabla \cdot (a \nabla u) = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \quad (3.1)$$

where $f \in L^2(\Omega)$ and $a \in C^1(\Omega)$ with $a(\mathbf{x}) \geq \tilde{a} > 0$.

And we consider the weak form of (3.1): find $u \in H_0^1(\Omega)$, such that

$$B(u, v) = L(v), \quad \forall v \in H_0^1(\Omega), \quad (3.2)$$

where B is the bilinear form and L is the linear functional defined by

$$B(u, v) = \int_{\Omega} a \nabla u \cdot \nabla v dx, \quad L(v) = \int_{\Omega} f v dx, \quad \forall u, v \in H_0^1(\Omega).$$

Define the so-called *energy norm* of u by $\|u\|_E = \sqrt{B(u, u)}$, for any $u \in H_0^1(\Omega)$.

3.2. Finite element space

Denote \mathcal{T} as a T-mesh over domain Ω , then define [32]

$$S(m, n, \alpha, \beta, \mathcal{T}) := \{s(\mathbf{x}) \in C^{\alpha, \beta}(\Omega) | s(\mathbf{x}) \in \mathbb{P}_{mn} \text{ over any element } K \in \mathcal{T}\},$$

where \mathbb{P}_{mn} is the space of all the bivariate polynomials with degree (m, n) , and $C^{\alpha, \beta}$ denotes the space of all the bivariate functions which are continuous in Ω with order (α, β) . And $S(m, n, \alpha, \beta, \mathcal{T})$ is therefore a spline space over \mathcal{T} .

In [32], a dimension formula for the spline space $S(m, n, \alpha, \beta, \mathcal{T})$, with $m \geq 2\alpha + 1$ and $n \geq 2\beta + 1$, is provided. For our purpose of use, we set $m = n = 3, \alpha = \beta = 1$, so we have

$$\dim S(3, 3, 1, 1, \mathcal{T}) = 4(V^b + V^+), \quad (3.3)$$

where V^b and V^+ represent the number of boundary and crossing vertices, respectively. For simplicity, we define

$$\mathcal{U}(\mathcal{T}) := S(3, 3, 1, 1, \mathcal{T})$$

as the finite element space for mesh \mathcal{T} over Ω .

From (3.3), we can see that each boundary vertex or crossing vertex is associated with four basis functions, and we may not associate basis functions with T-vertices. So a vertex (boundary or crossing) with a basis functions defined on it is named a *basis vertex*. All the basis functions are defined in a hierarchical manner, which we sketch below.

3.3. Basis functions for PHT-splines

To construct basis functions for $\mathcal{U}(\mathcal{T})$, we adopt a level-by-level strategy (see [17]). For simplicity, we set the initial mesh to be uniform rectangular grid. For this initial level (level 0, denoted as \mathcal{T}_0), the standard bicubic tensor-product B-splines are used as basis functions. Let the grid lines be given by $\{x = x_i\}_{i=1}^s$ and $\{y = y_i\}_{i=1}^s$, there are four basis functions to be defined on any given vertex (x_i, y_j) since all the vertices are either crossing vertices or boundary vertices. Each basis

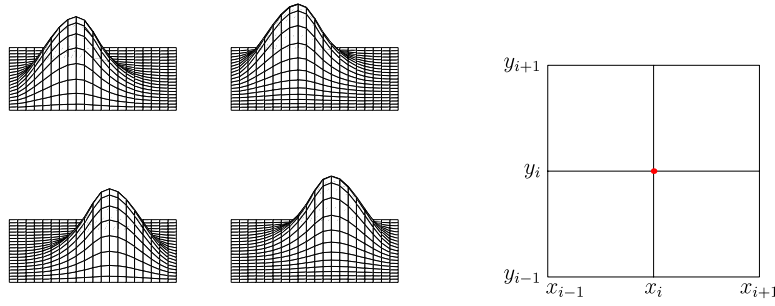


Fig. 3. Four basis functions associated with (x_i, y_j) and their support.

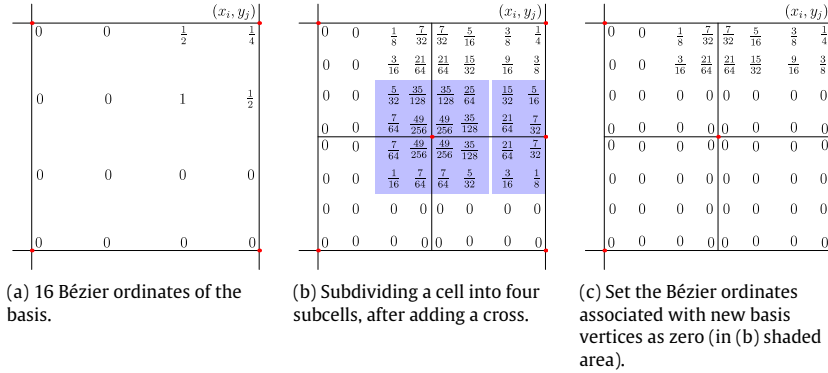


Fig. 4. Modification of a basis function at (x_i, y_j) .

function at (x_i, y_j) has its support being $[x_{i-1}, x_{i+1}] \times [y_{j-1}, y_{j+1}]$. These four basis functions are defined to be *B-spline basis functions* with knots $(x_{i-1}, x_i, x_{i+1}) \times (y_{j-1}, y_j, y_{j+1})$, $(x_{i-1}, x_i, x_{i+1}) \times (y_{j-1}, y_j, y_{j+1})$, $(x_{i-1}, x_i, x_{i+1}) \times (y_{j-1}, y_j, y_{j+1})$, $(x_{i-1}, x_i, x_{i+1}) \times (y_{j-1}, y_j, y_{j+1})$, respectively (see Fig. 3), so that their function values and derivatives vanish outside $[x_{i-1}, x_{i+1}] \times [y_{j-1}, y_{j+1}]$.

Now, assuming that some refinement is taking place on \mathcal{T}_0 , we denote the T-mesh at level k as \mathcal{T}_k . Inductively, suppose the basis functions $\{b_j^k\}$, $j = 1, \dots, d_k$, on \mathcal{T}_k have been constructed, the basis functions on \mathcal{T}_{k+1} can be constructed from two sources: some are from the modifications of the old basis functions on \mathcal{T}_k , and others are from the new basis functions associated with the new basis vertices of \mathcal{T}_{k+1} .

Notice the fact that a basis function can be represented by specifying its 16 Bézier ordinates (coefficients) in every cell within the support of the basis function. When a cell is refined by adding a cross vertex, the cell is subdivided into four subcells. Each subcell is in the support of the original basis function, and there are 16 Bézier ordinates on it. One can simply apply the formula (14.15) in [33] for this process. After this subdivision, the basis function remains the same, only that it is defined on the refined mesh. By adding a cross vertex, we get 5 new vertices, some of which are new basis vertices. Then we reset all the associated Bézier ordinates associated with the new basis vertices to zero. Fig. 4 illustrates this process.

Other than the modification of old basis functions, there are some new basis vertices (all basis vertices are colored in red). These new basis vertices may either be crossing vertices or some T-vertices from the mesh of the previous level that become basis vertices due to the subdivision of the neighboring cell(s). For these kinds of basis vertices, we simply build basis functions over their supports.

We note that during the refinement, *cross insertion* and *cross removal* are sometimes inevitable. This is one drawback of PHT-splines, although these two processes do not cost much computational resources since they are local operations. One may check [17] for more details on the construction of basis functions.

Notice that the above basis construction procedures are very different from that used in [29] for the adaptive BFS elements. In [29], there are degree-of-freedom defined on hanging nodes (T-intersections), and degree-of-freedom values on the finer mesh are decided by those on the coarser mesh in order to maintain the C^1 continuity after each refinement. Although the adaptive BFS and PHT-splines generate the same function space, their bases are different. In comparison with the procedure outlined in [29], the PHT-splines basis construction maintains the C^1 continuity automatically without post-processing, and it leads to a basis that enjoys many properties shared by the B-splines, such as the nonnegativity, local support and the partition of unity. Such properties are advantageous when the splines are used not only to solve PDEs but also to represent geometric domains in the isogeometric analysis. Moreover, they also allow one to easily extend similar constructions to higher order and smoother adaptive splines which will be discussed in forthcoming works.

3.4. Finite element discretization

Our finite element approximation $u_h \in \mathcal{U}_0(\mathcal{T})$, where $\mathcal{U}_0(\mathcal{T})$ is a subset of $\mathcal{U}(\mathcal{T})$ imposed with corresponding boundary conditions, of problem (3.1) is determined by

$$B(u_h, v_h) = L(v_h), \quad \forall v_h \in \mathcal{U}_0(\mathcal{T}). \quad (3.4)$$

On a given T-mesh \mathcal{T} over Ω , to impose essential boundary conditions, a so-called *weight function* has been introduced. This is because, if a linear combination of B-splines, say $f(\mathbf{x}) = \sum_{i=1}^d c_i b_i(\mathbf{x})$, is to vanish along the boundary $\partial\Omega$, then all coefficients c_i of B-splines whose support intersects $\partial\Omega$ should be set to 0. This will compromise the order of approximation, since B-splines outside the domain may have a zero coefficient. In this paper, since PHT-splines are defined on a rectangular domain Ω with boundary $\partial\Omega$, we set the weight function to be 1, which makes no essential difference to our analysis. We refer to [34,35] for more details. It is worth noting that this weight function is not necessary for level 0 mesh and the uniform refinement thereafter. This is because, under the definition of PHT-splines, the support of a basis vertex just consists of its neighboring cells.

Let us denote by h_K the diameter of a cell $K \in \mathcal{T}$, and define $h = \max_{K \in \mathcal{T}} h_K$ to be the mesh size of \mathcal{T} . We also assume that the exact solution $u \in H^2(\Omega)$. Let $e_h = u - u_h$ be the error of the approximation u_h , then we have the following classical a priori error estimates [36].

Theorem 1. *There exist constants C independent of a and h such that*

$$\|e_h\|_E \leq Ch^{k-1} \|\sqrt{a} |\nabla_k u|\|_{L^2(\Omega)}, \quad k = 1, 2, \quad (3.5)$$

and

$$\|e_h\|_{L^2(\Omega)} \leq h^2 C \|\sqrt{a} |\nabla_2 u|\|_{L^2(\Omega)}. \quad (3.6)$$

3.5. Interpolation and best approximation

Let \mathcal{T} be a hierarchical T-mesh, with basis functions $\{b_i(\mathbf{x}), i = 1, \dots, d\}$, constructed as in the above discussion. Then, the interpolation of $f(\mathbf{x})$, say $f^h(\mathbf{x})$, defined on $\mathcal{U}(\mathcal{T})$ may be expressed as

$$f^h(\mathbf{x}) = \sum_{i=1}^d c_i b_i(\mathbf{x}), \quad \mathbf{x} \in \Omega. \quad (3.7)$$

For any basis function $b(\mathbf{x})$, we define a *geometric information* operator by

$$\mathcal{L}b(\mathbf{x}) = (b(\mathbf{x}), b_x(\mathbf{x}), b_y(\mathbf{x}), b_{xy}(\mathbf{x}))^T. \quad (3.8)$$

It follows that $\mathcal{L}(\cdot)$ is a linear operator and satisfies the following properties:

- For any basis function $b(\mathbf{x})$ and any basis vertex \mathbf{x}_0 in \mathcal{T} , $\mathcal{L}b(\mathbf{x}_0) = 0$ holds for all the basis functions, except for the four basis functions associated with \mathbf{x}_0 .
- For a basis vertex \mathbf{x}_0 which appears in the hierarchical T-mesh since level k_0 and a basis function $b_j^k(\mathbf{x})$ constructed at level k , $\mathcal{L}b_j^k(\mathbf{x}_0)$ remains unchanged for $k \geq k_0$.

With the help of $\mathcal{L}(\cdot)$, we can determine coefficients c_i of $f^h(\mathbf{x})$. For any fixed basis vertex \mathbf{x}_0 whose four basis functions have indices i_1, i_2, i_3, i_4 , we apply $\mathcal{L}(\cdot)$ on $f^h(\mathbf{x}_0)$. Since we want to keep the geometries as well as we can, so we set

$$\mathcal{L}f(\mathbf{x}_0) = \mathcal{L}f^h(\mathbf{x}_0) = \sum_{i=1}^d c_i \mathcal{L}b_i(\mathbf{x}_0) = \sum_{j=1}^4 c_{j_i} \mathcal{L}b_{j_i}(\mathbf{x}_0) = \mathbf{B} \cdot \mathbf{c}, \quad (3.9)$$

and solve for c_i 's, where $\mathbf{B} = (\mathcal{L}b_{j_1}(\mathbf{x}_0), \mathcal{L}b_{j_2}(\mathbf{x}_0), \mathcal{L}b_{j_3}(\mathbf{x}_0), \mathcal{L}b_{j_4}(\mathbf{x}_0))$ is a 4×4 matrix, $\mathbf{c} = (c_{j_1}, c_{j_2}, c_{j_3}, c_{j_4})^T$, and $\mathcal{L}f(\mathbf{x}_0)$ is the geometric information of $f(\mathbf{x})$ at \mathbf{x}_0 .

As stated in [17], the matrix \mathbf{B} has the following form

$$\mathbf{B} = \begin{pmatrix} (1-\lambda)(1-\mu) & \lambda(1-\mu) & \lambda\mu & (1-\lambda)\mu \\ -\alpha(1-\mu) & \alpha(1-\mu) & \alpha\mu & -\alpha\mu \\ -\beta(1-\lambda) & -\beta\lambda & \beta\lambda & \beta(1-\lambda) \\ \alpha\beta & -\alpha\beta & \alpha\beta & -\alpha\beta \end{pmatrix},$$

where

$$\alpha = \frac{1}{\Delta u_1 + \Delta u_2}, \quad \beta = \frac{1}{\Delta v_1 + \Delta v_2}, \quad \lambda = \alpha \Delta u_1, \quad \beta = \Delta v_1,$$

and the four neighbor cells around \mathbf{x}_0 are with sizes $3\Delta u_1 \times 3\Delta v_1$, $3\Delta u_2 \times 3\Delta v_1$, $3\Delta u_1 \times 3\Delta v_2$, $3\Delta u_2 \times 3\Delta v_2$. Note that here \mathbf{B} is different from the one in [17]. Actually, \mathbf{B} is determined by the order of the four basis functions at each basis vertex, each column stands for the geometric information of a basis function. In our case, for basis vertex (x_i, y_j) , the control points for the four basis functions are selected in the following order: $(x_{i-1}, x_{i-1}, x_i, x_i, x_{i+1}) \times (y_{j-1}, y_{j-1}, y_j, y_j, y_{j+1})$, $(x_{i-1}, x_i, x_i, x_{i+1}, x_{i+1}) \times (y_{j-1}, y_{j-1}, y_j, y_j, y_{j+1})$, $(x_{i-1}, x_i, x_i, x_{i+1}, x_{i+1}) \times (y_{j-1}, y_j, y_j, y_{j+1}, y_{j+1})$, $(x_{i-1}, x_{i-1}, x_i, x_i, x_{i+1}) \times (y_{j-1}, y_j, y_j, y_{j+1}, y_{j+1})$.

The matrix \mathbf{B} is invertible, so we have

$$\mathbf{c} = \mathbf{B}^{-1} \cdot \mathcal{L}f(\mathbf{x}_0),$$

the interpolation is used to impose boundary conditions when solving PDEs.

Since $f(\mathbf{x})$ and $f^h(\mathbf{x})$ have the same geometric information at basis vertices, according to approximation theories for BFS rectangle in [36], if $f(\mathbf{x}) \in H^4(\Omega)$, then there exists a constant C independent of h such that

$$\|f - f^h\|_{H^1(\Omega)} \leq C h^3 \|f\|_{H^4(\Omega)}, \quad (3.10)$$

and this gives the optimal order of approximation of $f(\mathbf{x})$.

Applying the classical finite element theory and the equivalence of energy norm with H^1 norm, the following approximation result holds.

Theorem 2. Suppose $u(\mathbf{x}) \in H^4(\Omega)$ is the solution of problem (3.1), and $u^h(\mathbf{x})$ is the solution of the discrete problem (3.4), then there exists a generic constant $C > 0$ such that

$$\|u - u^h\|_E \leq C h^3 \|u\|_{H^4(\Omega)}. \quad (3.11)$$

4. Residual-based a posteriori error estimate

From (3.2) and (3.4), we have for any $v \in H_0^1(\Omega)$,

$$\begin{aligned} B(e_h, v) &= \int_{\Omega} f v dx - \int_{\Omega} (a \nabla u_h) \cdot \nabla v dx \\ &= \sum_{K \in \mathcal{T}} \left\{ \int_K f v dx - \int_K (a \nabla u_h) \cdot \nabla v dx \right\}. \end{aligned}$$

Apply integration by parts and rearrange terms, we can get

$$B(e_h, v) = \sum_{K \in \mathcal{T}} \int_K (f + \nabla \cdot (a \nabla u_h)) v dx - \sum_{\gamma \in \mathcal{E}_I} \int_{\gamma} \left((a \nabla u_h)|_{K_{\gamma_1}} \cdot \vec{\mathbf{n}}_{K_{\gamma_1}} + (a \nabla u_h)|_{K_{\gamma_2}} \cdot \vec{\mathbf{n}}_{K_{\gamma_2}} \right) v ds, \quad (4.1)$$

where \mathcal{E}_I is the set of interior edges of \mathcal{T} , K_{γ_1} and K_{γ_2} are two cells that share a common edge γ , $\vec{\mathbf{n}}_{K_{\gamma_1}}$ and $\vec{\mathbf{n}}_{K_{\gamma_2}}$ are outward unit normal vectors of K_{γ_1} and K_{γ_2} , respectively, along edge γ . Notice the second term in (4.1) vanishes due to the continuity of a and ∇u_h , so we have

$$B(e_h, v) = \sum_{K \in \mathcal{T}} \int_K (f + \nabla \cdot (a \nabla u_h)) v dx. \quad (4.2)$$

For given $v \in H_0^1(\Omega)$, let $I_h v$ be the interpolation of v on $\mathcal{U}_0(\mathcal{T})$. Then by the orthogonality property $B(e_h, I_h v) = 0$ and (4.2), we have

$$B(e_h, v) = \sum_{K \in \mathcal{T}} \int_K r(v - I_h v) dx, \quad \forall v \in H_0^1(\Omega), \quad (4.3)$$

where $r = f + \nabla \cdot (a \nabla u_h)$.

The identity (4.3) plays an important role throughout a posteriori error analysis of finite element approximations. Applying the Cauchy–Schwarz inequality gives

$$B(e_h, v) \leq \sum_{K \in \mathcal{T}} \|r\|_{L^2(K)} \|v - I_h v\|_{L^2(K)}. \quad (4.4)$$

Due to the coercivity of the bilinear form $B(\cdot, \cdot)$ and the approximation theory [19], by letting $v = e_h$, we can obtain the a posteriori error estimate: for a generic constant C independent of v and h_K ,

$$\|e_h\|_E^2 \leq C \sum_{K \in \mathcal{T}} h_K^2 \|r\|_{L^2(K)}^2. \quad (4.5)$$

Except for the constant C , all the other quantities on the right-hand side of (4.5) can be computed explicitly from the finite element solution u_h . So we get a residual-based H^1 -type [19] local error estimator η_K associated with the cell $K \in \mathcal{T}$,

defined by

$$\eta_K^2 = h_K^2 \|r\|_{L^2(K)}^2. \quad (4.6)$$

Remark 1. When using a piecewise polynomial finite element space, the basis functions do not have global smoothness so that the second term in (4.1) is usually not zero. This will lead to another term in the estimator, which evaluates the *jump* along a common edge of two adjacent cells.

The locally computed quantity η_K is the contribution from cell K to the bound for the global error $\|e\|_E$. The local quantity η_K is not supposed to provide a good estimate for the real local error in cell K , due to the possible *pollution errors* due to influences away from the local cell. An important thing is that the real error can be bounded from above in terms of η_K , i.e. when η_K is small, the real error e_h over K must also be small. This property is called the *reliability* of the a posteriori estimator.

Other than reliability, the *efficiency* of the estimator is also important for an adaptive numerical method. Generally, we want the real error could be bounded by the estimator from below, up to some constant, i.e. there exists a constant, which does not depend on the mesh size, such that an inequality like the following should hold (possibly with some extra relatively small terms)

$$\eta_K^2 \leq C \|e\|_{E(K)}^2.$$

To demonstrate the efficiency, we follow the framework outlined by Verfürth in [28]. Over a given cell K , we first need to construct a locally supported, nonnegative *bubble function*. Here we just need the *interior* bubble function, since there is no contribution from the edge in the estimator. There are various ways to define such a function with the understanding that an interior bubble function can localize residual to a single cell. For example, if the interior residual r defined above is multiplied by an interior bubble function ϕ_K , then the term $v = r\phi_K$ vanishes on the cell boundary. A simple example of an interior bubble function over the cell $K = [x_0, x_1] \times [y_0, y_1]$ is $\phi_K = (x - x_0)^2(x - x_1)^2(y - y_0)^2(y - y_1)^2$.

We have the following result about the effect of multiplication by the bubble function [19].

Theorem 3. Let ϕ_K denote the interior bubble function defined on K . There exists a constant C such that for all $v \in \mathcal{U}(\mathcal{T})$

$$C^{-1} \|v\|_{L^2(K)}^2 \leq \int_K \phi_K v^2 d\mathbf{x} \leq C \|v\|_{L^2(K)}^2 \quad (4.7)$$

and

$$C^{-1} \|v\|_{L^2(K)} \leq \|\phi_K v\|_{L^2(K)} + h_K |\phi_K v|_{H^1(K)} \leq C \|v\|_{L^2(K)} \quad (4.8)$$

where the constant C is independent of v and h_K .

Let \bar{r} be an interpolation of r onto $\mathcal{U}(\mathcal{T})$. By applying Theorem 3 we get for $K \in \mathcal{T}$ that

$$\|\bar{r}\|_{L^2(K)}^2 \leq C \int_K \phi_K \bar{r}^2 d\mathbf{x}. \quad (4.9)$$

Setting $v = \bar{r}\phi_K$ in (4.1), we get

$$B(e_h, \bar{r}\phi_K) = \int_K r \phi_K \bar{r} d\mathbf{x}, \quad (4.10)$$

and also

$$\int_K \phi_K \bar{r}^2 d\mathbf{x} = \int_K \phi_K \bar{r} (\bar{r} - r) d\mathbf{x} + B(e_h, \bar{r}\phi_K). \quad (4.11)$$

With the Cauchy–Schwarz inequality and Theorem 3, we have

$$\begin{aligned} \int_K \phi_K \bar{r} (\bar{r} - r) d\mathbf{x} &\leq \|\phi_K \bar{r}\|_{L^2(K)} \|\bar{r} - r\|_{L^2(K)} \\ &\leq C \|\bar{r}\|_{L^2(K)} \|\bar{r} - r\|_{L^2(K)}, \end{aligned} \quad (4.12)$$

and

$$\begin{aligned} B(e_h, \bar{r}\phi_K) &\leq \|e_h\|_{E(K)} \|\phi_K \bar{r}\|_{H^1(K)} \\ &\leq Ch_K^{-1} \|\bar{r}\|_{L^2(K)}. \end{aligned} \quad (4.13)$$

Combining (4.9) and (4.11)–(4.13), we have

$$\|\bar{r}\|_{L^2(K)} \leq C (h_K^{-1} \|e_h\|_{E(K)} + \|\bar{r} - r\|_{L^2(K)}).$$

Table 1
Computational results for Example 1, under uniform refinement.

| DOF | $\ e_h\ _{L^2(\Omega)}$ | CR | $\ e_h\ _{H^1(\Omega)}$ | CR |
|-------|-------------------------|------|-------------------------|------|
| 144 | 4.4814e–006 | 0.00 | 3.5925e–005 | 0.00 |
| 484 | 3.0502e–007 | 4.43 | 3.9999e–006 | 3.62 |
| 1764 | 1.9496e–008 | 4.25 | 4.7997e–007 | 3.28 |
| 6724 | 1.2265e–009 | 4.13 | 5.9329e–008 | 3.12 |
| 26244 | 7.6825e–011 | 4.07 | 7.3957e–009 | 3.06 |

Thus, with the triangle inequality, we get

$$\|r\|_{L^2(K)} \leq C \left(h_K^{-1} \|e_h\|_{E(K)} + \|\bar{r} - r\|_{L^2(K)} \right),$$

which is equivalent to

$$\eta_K^2 \leq C \left(\|e_h\|_{E(K)}^2 + h_K^2 \|\bar{r} - r\|_{L^2(K)} \right). \quad (4.14)$$

By choosing a good approximation of r , we can make the second term relatively small in comparison with the first term in (4.14). Thus the efficiency of the estimator is proved.

Remark 2. As stated before, a virtue of PHT-splines is that all the basis functions are piecewise polynomials which not only lower the computational cost in practical implementation but also make the analysis possible within standard finite element framework. When other B-splines, such as NURBS, are used, the analysis requires more effort.

5. Numerical experiment

In this section, we present computational experiments to demonstrate the practicality of the PHT-splines in the numerical solution of PDEs and to verify the theoretical analysis presented above. For illustration, we consider a relatively simple diffusion–reaction problem

$$\begin{cases} -\nabla \cdot (a \nabla u) + bu = f & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega, \end{cases} \quad (5.1)$$

where $\Omega = [0, 1] \times [0, 1]$ and $b \in L^\infty$ with $b \geq 0$. The model is slightly different from (3.1), and correspondingly, instead of setting $r = f + \nabla \cdot (a \nabla u_h)$, we let $r = f + \nabla \cdot (a \nabla u_h) - bu_h$, whereas the above theoretical analysis can be trivially extended to cover the present case.

In the numerical simulations, the initial meshes are taken to be uniform rectangular meshes, and the refinement at each level is achieved by performing the marking strategy used in [37] with a parameter θ , which is used to control the refinement process [38].

The convergence rate CR with respect to the norm $\|\cdot\|$ at the refinement level l is roughly computed by

$$CR = \frac{2 \log(\|e_{h,l}\| / \|e_{h,l-1}\|)}{\log(n_{l-1} / n_l)}, \quad (5.2)$$

where n_l denotes the number of basis vertices and $e_{h,l}$ denotes the error $u - u_h$ at refinement level l .

In the following examples, we consider problems where the exact solutions are explicitly given in order to more easily assess the qualities of the numerical solutions. All the exact solutions have concentrated regions of large spatial gradients either near some points or near some curves so that proper adaptive refinement would enhance the efficiency of the numerical solution. For simplicity, all the graphical presentations of the solutions given in the figures are generated only by the solution values on the mesh nodes. One should note that the solution should actually be C^1 smooth.

Boundary conditions are imposed with the help of interpolation. Since not all the geometric information on boundary nodes are available, we cannot get all the four coefficients on each boundary nodes and consider the rest as unknowns.

Example 1. In the first experiment, global refinement is performed to verify the convergence rate presented in Theorem 2. We choose an exact solution $u(\mathbf{x}) = e^{x(1-x)y(1-y)} - 1$, and set $a(\mathbf{x}) = x + y$, $b(\mathbf{x}) = \sin(x + y)$, f and g are determined from u by (5.1).

Implementing the finite element with globally refined meshes and the PHT-spline basis functions, we get numerical approximations with 5 different levels of resolution, i.e., uniform 5×5 , 10×10 , 20×20 , 40×40 and 80×80 mesh, as shown in Table 1 (DOF = degree of freedom). Some discrete solutions are depicted in Fig. 5. It is obvious, from the data, that the convergence rate is consistent with our analysis for both L^2 and H^1 norms.

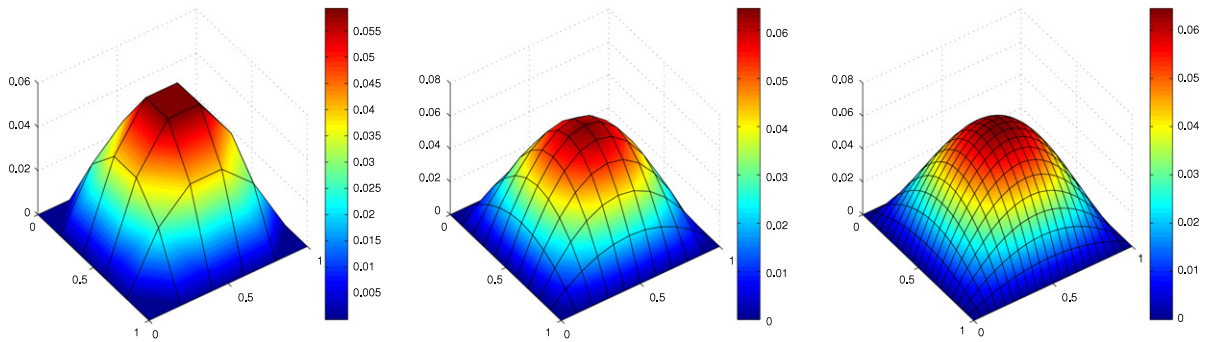


Fig. 5. Discrete solution u^h for Example 1, with degrees of freedom 144, 484 and 1764, respectively.

Table 2

Mesh qualities, energy norm errors, a posteriori error estimates for Example 2 (with 8 levels).

| DOF | $\ e_h\ _E$ | CR | η_T | CR | $\eta_T / \ e_h\ _E$ |
|-------|-------------|-------|-----------|------|----------------------|
| 484 | 3.531500 | 0.00 | 10.514000 | 0.00 | 2.977 |
| 796 | 0.275340 | 10.26 | 2.661000 | 5.52 | 9.664 |
| 1408 | 0.077305 | 4.45 | 0.822620 | 4.12 | 10.641 |
| 2680 | 0.031617 | 2.78 | 0.334950 | 2.79 | 10.594 |
| 5028 | 0.013027 | 2.82 | 0.131760 | 2.97 | 10.114 |
| 9700 | 0.004719 | 3.09 | 0.050146 | 2.94 | 10.626 |
| 18412 | 0.001977 | 2.72 | 0.019929 | 2.88 | 10.080 |
| 35064 | 0.000729 | 3.10 | 0.007436 | 3.06 | 10.203 |

Example 2. We set $a(\mathbf{x}) = 10 \cdot \cos(y)$, $b(\mathbf{x}) = x^2 + y^2$ and choose the exact solution to be

$$u(\mathbf{x}) = \frac{1.0}{|\mathbf{x} - (0.5, 0.5)|^2 + 0.02}, \quad (5.3)$$

with f and g being determined from u by Eq. (5.1).

The exact solution $u = u(\mathbf{x})$ defined in (5.3) is a smooth function, it reaches a peak value of 50 at the point (0.5, 0.5) then decays quickly away from the peak, thus it has a large gradient near (0.5, 0.5). Note that the data $a(\mathbf{x})$ and $f(\mathbf{x})$ also have relatively rapid variations over Ω .

The initial mesh consists of a 10×10 square uniform mesh, and the refining parameter θ is set to be 0.4. Fig. 6 shows three levels of the refinement (with number of basis vertices 484, 796, 1408, respectively). It can be easily seen that the meshes around the peak of u get significantly refined. In Fig. 7, we graph the energy norm errors of the approximate solution u_h and the a posteriori error estimates at all levels, along with some reference slopes. Table 2 contains the data used to produce Fig. 7, as well as the convergence rates for each level and the ratios between a posteriori estimate η_T and actual energy norm error $\|e_h\|_E$. It can be concluded that our a posteriori estimate has the same convergence rate as the energy norm error, which implies the reliability and efficiency of our estimate. In fact, it can be seen that the ratio $\frac{\eta_T}{\|e_h\|_E}$ is roughly around the value 10.

Example 3. We set $a(\mathbf{x}) = 1$, $b(\mathbf{x}) = 2$ and choose the exact solution to be

$$u(\mathbf{x}) = \frac{1.0}{|\mathbf{x} - (0.25, 0.25)|^2 + 0.01} - \frac{1.0}{|\mathbf{x} - (0.75, 0.75)|^2 + 0.01}, \quad (5.4)$$

with f and g being determined from $u = u(\mathbf{x})$ via Eq. (5.1).

In this case, the exact solution (5.4) is a smooth function with a maximum value $99 \frac{101}{201}$ near the point (0.25, 0.25) and a minimum value $-99 \frac{101}{201}$ near the point (0.75, 0.75), it again shows significant variations near its extrema.

We set the initial mesh to be a 10×16 uniform rectangular mesh over Ω and let $\theta = 0.3$. As seen from Fig. 8 which shows three levels of the refinement with the number of basis vertices being 748, 1300 and 2296 respectively, it is obvious that meshes around the two extrema are heavily refined, while elements away from the extrema are nearly unchanged or with only slight refinement. Fig. 9 and Table 3 give the data on the energy norm errors of the approximation as well as the a posteriori error estimates. The consistency of the two can be observed. And from Table 3 we can see that the ratio between the a posteriori estimate and the actual error is nearly a constant around the value 9.

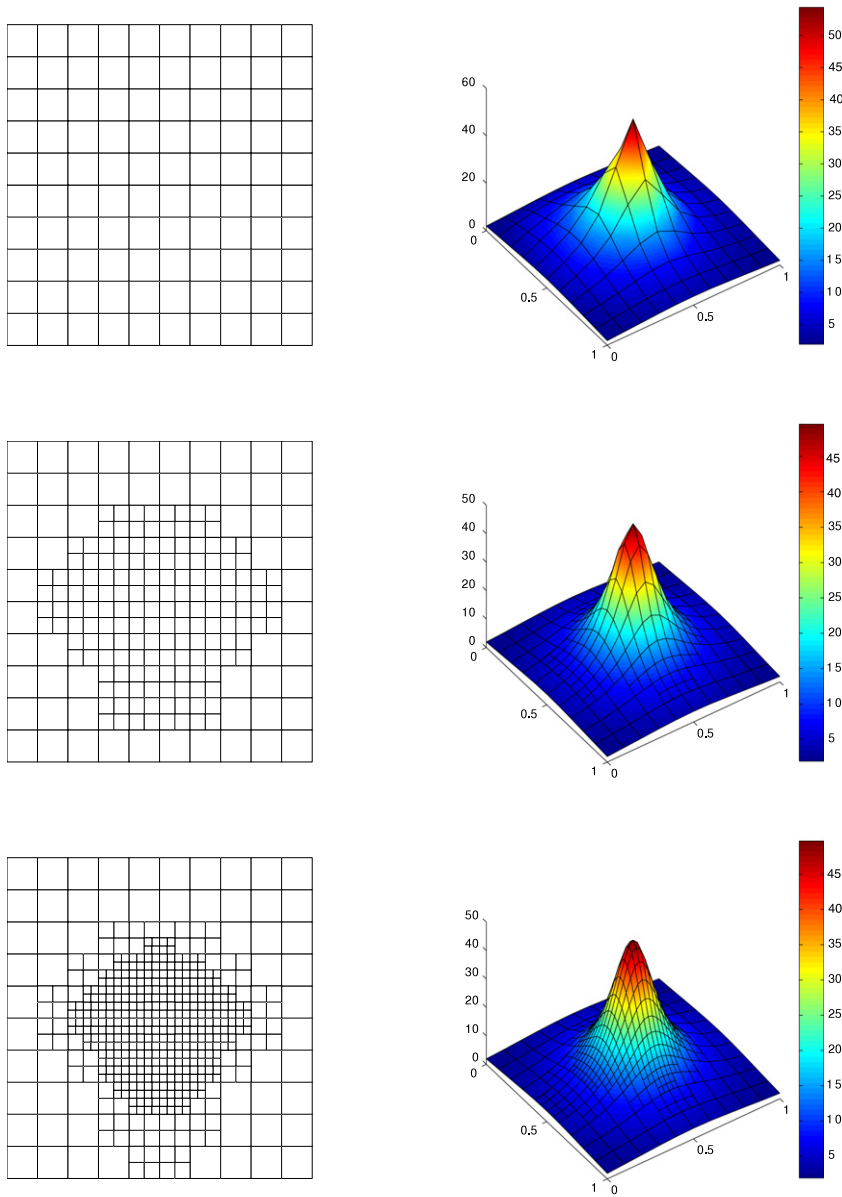


Fig. 6. Process of refinement for *Example 2*, with degrees of freedom 484, 900 and 1392.

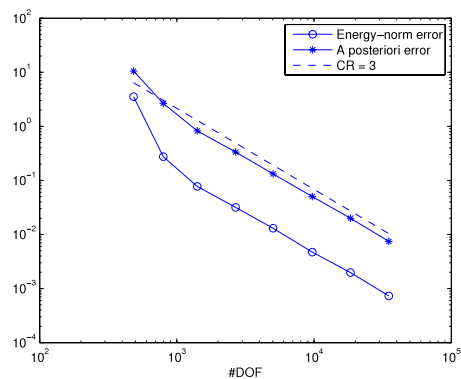


Fig. 7. Comparison of the energy norm errors and the a posteriori estimates at all 8 levels for *Example 2*.

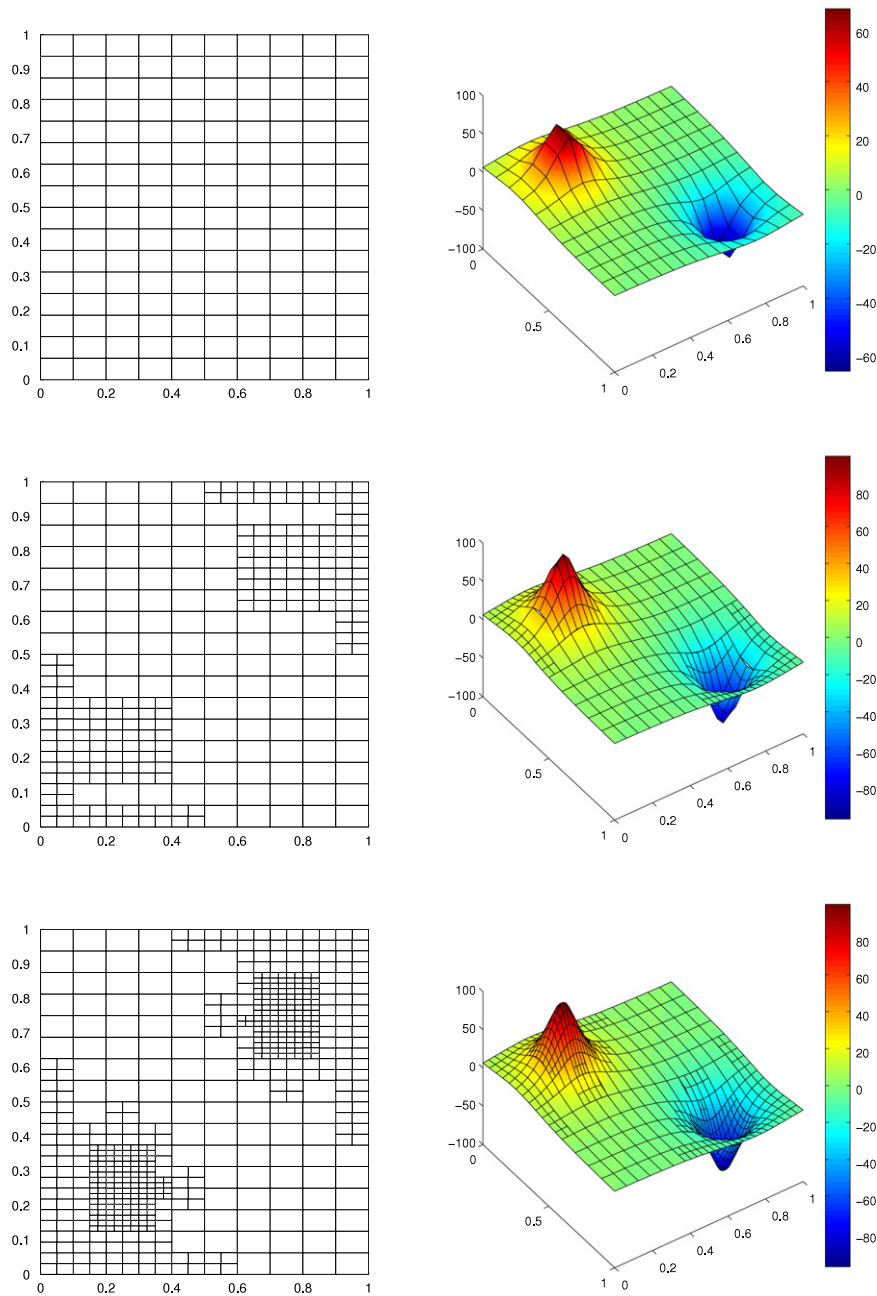


Fig. 8. Process of refinement for Example 3, with degrees of freedom 748, 1300 and 2296.

Example 4. We set $a(\mathbf{x}) = 2$, $b(\mathbf{x}) = 0$ and choose the exact solution to be

$$u(\mathbf{x}) = \tanh\left(\frac{0.25 - |\mathbf{x} - (0.5, 0.5)|}{0.03}\right), \quad (5.5)$$

and f and g are determined from $u(\mathbf{x})$ by (5.1).

The exact solution defined by (5.5) has value 1 near the point $(0, 0)$, value -1 away from $(0, 0)$, the decay is very fast around the circle $|\mathbf{x}| = 0.25$, so it has very large gradients there. We set $\theta = 0.2$. Fig. 10 shows the initial 10×10 uniform square-cell mesh (with a degree of freedom being 484), and the third and fifth levels of refinement (with degrees of freedom being 1480 and 3712, respectively). One can easily observe significant refinement around the large gradient area. The same conclusion about the convergence, similar to that in the first two experiments, can be obtained by examining Table 4 and Fig. 11. The a posteriori error estimate maintains the similar convergence rate as energy norm error, and the ratio between

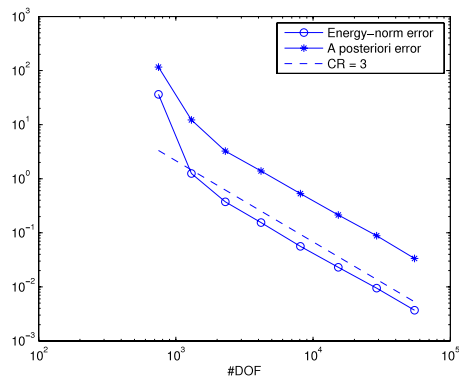


Fig. 9. Comparison of the energy norm errors and the a posteriori estimates at all 8 levels for Example 3.

Table 3

Mesh qualities, energy norm errors, a posteriori error estimates for Example 3 (with 8 levels).

| DOF | $\ e_h\ _E$ | CR | η_T | CR | $\eta_T / \ e_h\ _E$ |
|-------|-------------|-------|------------|------|----------------------|
| 748 | 36.374000 | 0.00 | 115.970000 | 0.00 | 3.188 |
| 1300 | 1.247800 | 12.20 | 12.204000 | 8.15 | 9.780 |
| 2296 | 0.374210 | 4.23 | 3.237600 | 4.67 | 8.652 |
| 4180 | 0.154410 | 2.95 | 1.389300 | 2.82 | 8.997 |
| 8092 | 0.056109 | 3.06 | 0.529130 | 2.92 | 9.430 |
| 15276 | 0.023028 | 2.80 | 0.213500 | 2.86 | 9.271 |
| 29048 | 0.009490 | 2.76 | 0.088221 | 2.75 | 9.296 |
| 54888 | 0.003701 | 2.96 | 0.033558 | 3.04 | 9.066 |

Table 4

Mesh qualities, energy norm errors, a posteriori error estimates for Example 4 (with 11 levels).

| DOF | $\ e_h\ _E$ | CR | η_T | CR | $\eta_T / \ e_h\ _E$ |
|-------|-------------|-------|-----------|-------|----------------------|
| 484 | 12.079000 | 0.00 | 38.697000 | 0.00 | 3.204 |
| 644 | 1.312500 | 15.54 | 6.600500 | 12.38 | 5.029 |
| 948 | 0.632130 | 3.78 | 3.975200 | 2.62 | 6.289 |
| 1480 | 0.346870 | 2.69 | 1.517700 | 4.32 | 4.375 |
| 2296 | 0.110040 | 5.23 | 0.643340 | 3.91 | 5.846 |
| 3712 | 0.053256 | 3.02 | 0.241700 | 4.08 | 4.538 |
| 6216 | 0.021070 | 3.60 | 0.112450 | 2.97 | 5.337 |
| 9540 | 0.009561 | 3.69 | 0.056950 | 3.18 | 5.956 |
| 15948 | 0.003516 | 3.89 | 0.021926 | 3.72 | 6.237 |
| 25620 | 0.002005 | 2.37 | 0.011858 | 2.59 | 5.915 |
| 40628 | 0.000867 | 3.64 | 0.005892 | 3.03 | 6.797 |

the two is around 6. It is worthy noting that the discrete solution over the initial mesh is very far from being well resolved, and as the refinement proceeds, cells with large gradients are captured very effectively, and we get excellent results after several refinements.

Remark 3. From the above numerical experiments, we can observe that the new basis functions for the PHT-splines can be effectively used to perform adaptive refinement. The refinement based on the a posteriori error estimations indeed happens progressively around the region where the solution has large gradients.

6. Conclusion

In this work, we developed an adaptive finite element method for the numerical solution of partial differential equations. The method is based on the new PHT-splines basis functions. The refinement is guided by a residual-based a posteriori error estimator for the finite element discretization. Since the PHT-splines use piecewise polynomials as basis functions, we were also able to use well-known results in the finite element literature to analyze their approximation properties. We presented the results of numerical experiments carried out for some model linear elliptic equations. The results are very encouraging as they not only serve to verify the theory but also demonstrate the robustness of the error estimators and the high order approximations provided by the adaptive spline space. Given the flexibility of the T-mesh and the smoothness of the

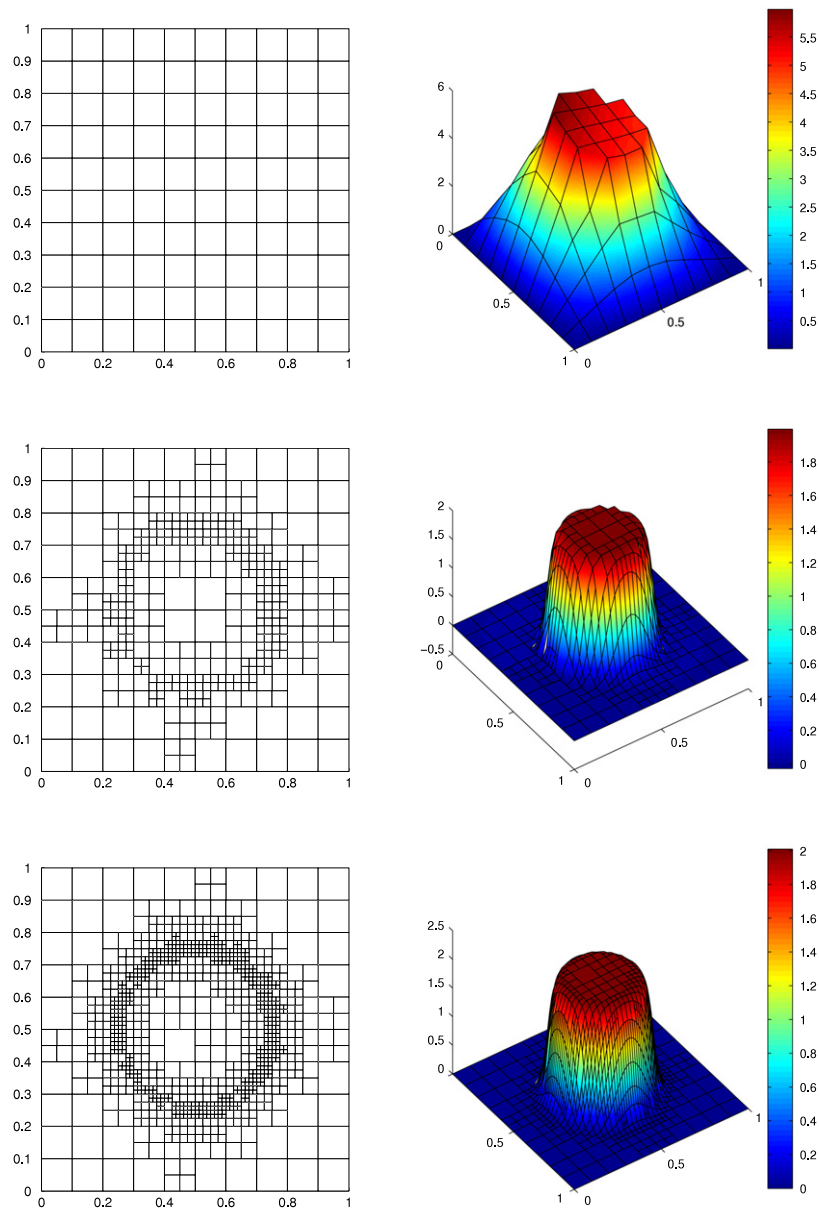


Fig. 10. Process of refinement for Example 4, with degrees of freedom 484, 1480 and 3712.

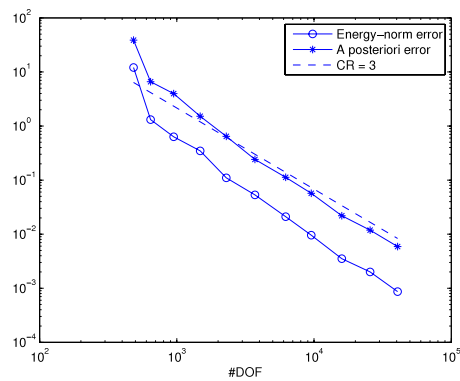


Fig. 11. With initial 10×10 mesh and $\theta = 0.1$. Left: Comparison of the energy norm errors and the a posteriori estimates at all levels for Example 3 (22 levels). Right: refined mesh after 15 levels.

PHT-splines, it is natural to consider their applications to the numerical solution of problems defined in complicated geometric domains, higher order equations and other complex nonlinear physical systems. These issues will be pursued in the future.

References

- [1] H. Antes, Bicubic fundamental splines in plate bending, *IJNME* 8 (1974).
- [2] R. Kao, Approximate solutions by utilizing hill functions, *Comput. Struct.* 3 (1973) 397–412.
- [3] R. Kao, Application of hill functions to two-dimensional plate problems, *Solids Struct.* 11 (1975) 21–31.
- [4] J.C. Boyer, P. Trompette, An improvement of the stress calculations in plane elasticity by use of spline functions, *Trans. ASME* 48 (1981) 972–974.
- [5] G. Loubignac, G. Cantin, G. Touzot, Continuous stress fields in finite element analysis, *AIAA J.* 15 (1977) 1645–1647.
- [6] V. Braibant, C. Fleury, Shape optimal design using B-splines, *Comp. Meth. Appl. Mech. Eng.* 44 (1984) 247–267.
- [7] U. Schramm, W. Pilkey, The coupling of geometric descriptions and finite elements using nurbs, *Finite Elem. Anal. Des.* 15 (1993) 11–34.
- [8] P.O. Frederickson, Triangular spline interpolation, Mathematics report 6-70 Dept of Mathematics, Lakehead University, 1970.
- [9] T.W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and t-nurcs, *ACM Trans. Graph.* 22 (3) (2003) 161–172.
- [10] W. Boehm, Inserting new knots into b-spline curves, *Comput. Aided Des.* 12 (1980) 199–201.
- [11] E. Cohem, T. Lyche, R.F. Riesenfeld, Discrete b-spline subdivision techniques in computer-aided geometric design and computer graphics, *Comput. Graph. Image Process.* 14 (1980) 87–111.
- [12] R.N. Goldman, T. Lyche, Knot Insertion and Deletion Algorithms for B-spline Curves and Surfaces, SIAM, Philadelphia, 1993.
- [13] H.-P. Seidel, Knot insertion from a blossoming point of view, *Comput. Aided Geom. Des.* 5 (1988) 81–86.
- [14] Michael R. Döfel, Bert Jüttler, Bernd Simeon, Adaptive isogeometric analysis by local h-refinement with t-splines, *Comput. Methods Appl. Mech. Eng.* 199 (5–8) (2010) 264–275. *Comput. Geom. Anal.*
- [15] T.W. Sederberg, D.L. Cardon, G.T. Finnigan, N.S. North, J. Zheng, T. Lyche, T-spline simplification and local refinement, *ACM Trans. Graph.* 23 (3) (2004) 276–283.
- [16] Y. Bazilevs, V. Calo, J. Cottrell, J. Evans, T. Hughes, S. Lipton, M. Scott, T. Sederberg, Isogeometric analysis using t-splines, *Comput. Methods Appl. Mech. Eng.* 199 (2009) 229–263.
- [17] Jiansong Deng, Falai Chen, Xin Li, Changqi Hu, Weihua Tong, Zhouwang Yang, Yuyu Feng, Polynomial splines over hierarchical t-meshes, *Graph. Models* 70 (4) (2008) 76–86.
- [18] I. Babuska, W.C. Rheinboldt, Error estimates for adaptive finite element computations, *SIAM J. Numer. Anal.* 15 (1987) 736–754.
- [19] M. Ainsworth, J. Oden, A Posteriori Error Estimation in Finite Element Analysis, Wiley-Interscience, New York, 2002.
- [20] I. Babuska, A. Miller, A feedback finite element method with a posteriori error estimates, *Comput. Methods Appl. Mech. Engrg.* 61 (1987) 1–40.
- [21] I. Babuska, W.C. Rheinboldt, A posteriori error estimates for the finite element method, *Internat. J. Numer. Methods Engrg.* 12 (1978) 1597–1615.
- [22] I. Babuska, W.C. Rheinboldt, A posteriori error analysis of finite element solutions for one-dimensional problems, *SIAM J. Numer. Anal.* 18 (1981) 565–589.
- [23] S. Bartels, C. Carstensen, G. Dolzmann, Inhomogeneous dirichlet conditions in a priori and a posteriori finite element error analysis, *Numer. Math.* 99 (2004) 1–24.
- [24] C. Carstensen, All first-order averaging techniques for a posteriori finite element error control on unstructured grids are efficient and reliable, *Math. Comp.* 73 (2003) 1153–1165.
- [25] A. Demlow, Local a posteriori estimates for pointwise gradient errors in finite element methods for elliptic problem, *Math. Comp.* 76 (2007) 19–42.
- [26] L. Ju, M. Gunzburger, W. Zhao, Adaptive finite element methods for elliptic pdes based on conforming centroidal voronoi-delaunay triangulations, *SIAM J. Sci. Comput.* 28 (2003) 2023–2053.
- [27] L. Ju, L. Tian, D. Wang, A posteriori error estimates for finite volume approximations of elliptic equations on general surfaces, *Comput. Methods Appl. Mech. Engrg.* 198 (2009) 716–726.
- [28] R. Verfürth, A Review of a Posteriori Error Estimation and Adaptive Mesh-refinement Techniques, Wiley-Teubner, Chichester, UK, 1996.
- [29] Roy H. Stogner, Graham F. Carey, Bruce T. Murray, Approximation of cahn-hilliard diffuse interface models using parallel adaptive mesh refinement and coarsening with c^1 elements, *Internat. J. Numer. Methods Engrg.* 64 (2006) 1–19.
- [30] H. Gómez, V. Calo, Y. Bazilevs, T. Hughes, Isogeometric analysis of the Cahn–Hilliard phase-field model, *Comput. Methods Appl. Mech. Eng.* 197 (2008) 4333–4352.
- [31] Y. Bazilevs, L. Beirão da Veiga, J.A. Cottrell, T.J.R. Hughes, G. Sangalli, Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes, *Math. Models Methods Appl. Sci.* 16 (7) (2006) 1031–1090.
- [32] J. Deng, F. Chen, Y. Feng, Dimensions of spline spaces over t-meshes, *J. Comput. Appl. Math.* 194 (2) (2006) 267–283.
- [33] G. Farin, Curves and Surfaces for CAGD—A Practical Guide, 5 edition, Morgan Kaufmann Publishers, 2002.
- [34] L.W. Kantorowitsch, W.I. Krylow, Näherungsmethoden der Höheren Analysis, VEB Deutscher Verlag der Wissenschaften, Berlin, 1956.
- [35] V.L. Rvachev, T.I. Sheiko, R-functions in boundary value problems in mechanics, *Appl. Mech. Rev.* 48 (4) (1995) 151–188.
- [36] P. Ciarlet, The Finite Element Method For Elliptic Problems, SIAM, 2002.
- [37] A. Schmidt, K. Siebert, Design of Adaptive Finite Element Software: The Finite Element Toolbox ALBERTA, Springer, New York, 2005.
- [38] W. Döfler, A convergent adaptive algorithm for poisson's equation, *SIAM J. Numer. Anal.* 33 (1996) 1106–1124.