Computer-Aided Design 43 (2011) 1438–1448

Contents lists available at SciVerse ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Adaptive isogeometric analysis using rational PHT-splines

Ping Wang [a,b], Jinlan Xu [a], Jiansong Deng [a,*], Falai Chen [a]

[a] Department of Mathematics, University of Science and Technology of China, Hefei, Anhui 230026, PR China

[b] Department of Mathematics, Soochow University, Suzhou, Jiangsu 215006, PR China

ARTICLE INFO

Keywords:
Isogeometric analysis (IGA)
FEA
PHT-splines
RPHT-splines
Residual-based a posteriori error estimator

ABSTRACT

Polynomial splines over hierarchical T-meshes (PHT-splines) have an efficient and simple local refinement algorithm, but fail to represent exactly certain complex engineering geometries. In this paper, based on the current isogeometric framework, we overcome the drawbacks of PHT-splines by extending these to Rational PHT-splines (RPHT-splines), and explore RPHT-splines as the basis for analysis. A residual-based posteriori error estimator using RPHT-splines basis functions is derived to guide the local refinement process adaptively. Numerical examples show the potential of RPHT-splines as the basis for the adaptive isogeometric analysis.

1. Introduction

Isogeometric analysis (IGA) introduced in [1], and later described in detail in [2], was motivated by the existing gaps between Computer Aided Design (CAD) and Finite Element Analysis (FEA). In traditional FEA, the geometry described in CAD needs to be translated to an analysis-suitable geometry and then replaced by finite element meshes for analysis. These meshes are usually an approximation of the geometry. If an accurate solution is to be obtained through a series of *h*-refinements, a link must be established between the meshes and the refinement step. Mesh generation, the link between the meshes and the refinement are severe bottlenecks in practical engineering analysis. The goal of IGA is to address these problems by using CAD basis functions for analysis, resulting in a direct design-to-analysis without the intermediate step of mesh generation. The geometry is exact at the coarsest level, thus eliminating the geometrical errors, and can be refined by knot insertion and order elevation. Refinement at any level can take place completely within the analysis framework, which eliminates the necessity to communicate with the geometry.

IGA based on NURBS has been successfully studied [3–7,1]. Both theoretical and practical results have shown that IGA methods are advantageous compared to traditional FEA with respect to convergence and accuracy of the results. However, the tensor-product topology structure of NURBS makes local refinement impossible. Using multiple patches solves this problem [6,2], but requires the use of constraint equations on those patches sharing a common interface, making it inconvenient to implement, and refinement still propagates throughout a given patch. Furthermore, it is impossible to represent most shapes using a single, watertight NURBS surface. Gaps and overlaps at intersections of surfaces cannot be avoided, making these inappropriate for analysis.

T-splines, proposed by Sederberg [8,9], a generalization of NURBS that allows T-junctions, make local refinement possible. Dörfel et al. [10], and Bazilevs et al. [11] discussed T-splines-based IGA. The refinement process of T-splines depends on the structure of the T-meshes. Analysis-suitable T-splines [12,13], a mildly restricted subset of T-splines, maintain the important mathematical properties of NURBS [14], such as linear independence [12,15] and partition of unity under some restrictions on the T-mesh [16], while providing an efficient and highly localized refinement capability [13].

PHT-splines [17,18], a generalization of B-splines over hierarchical T-meshes, have the beneficial properties of T-splines. PHT-splines are always polynomial instead of rational, and have a set of basis functions with desirable properties such as linear independence, non-negativity, partition of unity, compact support, global smoothness, and an efficient and simple local refinement algorithm, making them attractive for analysis. However, since PHT-splines are polynomial, they cannot exactly represent common engineering shapes such as circles, cylinders, spheres, ellipsoids, etc. Based on the current IGA framework, we will first extend PHT-splines to rational form in the paper.

The simple and efficient local refinement algorithm for PHT-splines is very desirable in the adaptive analysis process. In traditional FEA, a posteriori error estimator, based on the solution of the discrete problem, is a powerful tool for steering the adaptive process, since it provides control of the global error,

* Corresponding author.
E-mail address: dengjs@ustc.edu.cn (J. Deng).

0010-4485/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2011.08.026

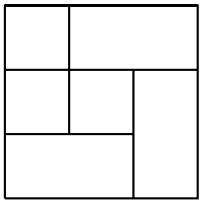


Fig. 1. Example of a 2D T-mesh.

while at the same time providing information about the error distribution. In the IGA literature, Bazilevs et al. [3] discussed in detail the approximation, stability and priori error estimates for h-refinement meshes in detail. Simeon [10] presented an implicit posteriori error estimation technique by solving auxiliary boundary value problems, the solutions of which are notionally approximations of the actual error functions. Theories on a posteriori error estimation are very mature in FEA, but little work has been done in IGA. In this paper, we will combine the error estimation technique by Verfürth [19] and Ainsworth [20] to derive a residual-based posteriori error estimator using rational PHT-splines basis functions within the current existing IGA framework.

The rest of the paper is organized as follows. In Section 2, we review pertinent notations for PHT-splines. The definition of Rational PHT-splines (RPHT-spines) and its properties are given in Section 3. Using RPHT-splines for analysis is explored in Section 4. A residual-based posteriori error estimator based on a model problem is derived and the implementation procedure for an adaptive refinement scheme is discussed in Section 5. Some two dimensional numerical examples are given and analyzed in Section 6. Section 7 concludes the paper with a summary and some future research topics.

2. Review of PHT-splines

In this section we briefly review some of the notations and the properties of PHT-splines, for more details see [18].

2.1. T-meshes and hierarchical T-meshes

Given a rectangle domain in 2D space, a T-mesh is a rectangular partition of the domain, where T-junctions are allowed. The end points of each grid line in the T-mesh must be on two other grid lines, and each cell in the grid must be a rectangle. Fig. 1 gives an example of a 2D T-mesh.

A hierarchical T-mesh starts with a tensor-product mesh $\mathscr{T}_0$. Let the T-mesh at level k be $\mathscr{T}_k$. For any $k \geqslant 0$, some selected cells at level k are subdivide equally into four subcells that are labeled at level $k+1$. Fig. 2 illustrates a 2D hierarchical T-mesh.

2.2. PHT-splines

Given a 2D T-mesh $\mathscr{T}$, let Φ be the set of all cells in $\mathscr{T}$. A polynomial spline space of all the bivariate functions that are continuous in Ω with order α in the x-direction and order β in the y-direction over $\mathscr{T}$ is defined as

$$\mathbb{S}(m, n, \alpha, \beta, \mathscr{T}) : = \{s(x, y) \in C^{\alpha,\beta}(\Omega) : s(x, y)|_{\phi} \in \mathbb{P}_{mn},\ \forall \phi \in \Phi\},$$

where $\mathbb{P}_{mn}$ is the space of all the bivariate polynomials with bi-degree (m, n). $\mathbb{S}(m, n, \alpha, \beta, \mathscr{T})$ is a linear space, with dimensions as given in [18].

A PHT-spline is a polynomial spline surface $\mathbb{S}(m, n, \alpha, \beta, \mathscr{T})$ defined over a hierarchical T-mesh $\mathscr{T}$. In this paper, we concentrate on the PHT-spline space $\mathbb{S}(3, 3, 1, 1, \mathscr{T})$.

The basis functions of a PHT-spline space $\mathbb{S}(3, 3, 1, 1, \mathscr{T})$ have properties such as linear independence, non-negativity, partition of unity and compact support. These features are most desirable in FEA, to ensure a well-conditioned and sparse matrix. But basis functions of a PHT-spline are not interpolating at the basis vertices, since a basis vertex associates four non-zero basis functions that form the partition of unity, which makes a bit more complicated treatment with essential boundary conditions than in traditional FEA.

3. RPHT-splines

The necessity to extend the PHT-splines to rational form and how to extend are discussed in this section.

PHT-splines have many important properties such as adaptivity and locality. Two or more PHT-splines can be stitched together [18]. PHT-splines are polynomial instead of rational, compared to NURBS and T-splines, this is a very promising feature in geometric modeling, which makes geometric computation less costly. However, within the current framework of IGA, the engineering geometry for analysis must first be exactly represented. We thus need to extend the PHT-splines to a rational form, since PHT-splines cannot exactly represent some common engineering shapes such as circles, cylinders, spheres, ellipsoids, etc.

Given a hierarchical T-mesh $\mathscr{T}$, we can define a PHT-spline space $\mathbb{S}(3, 3, 1, 1, \mathscr{T})$ over $\mathscr{T}$. Let $\{b_i(s, t)\}_{i=1}^m$ be the basis functions of the hierarchical T-mesh $\mathscr{T}$, and $\{P_i\}_{i=1}^m \in \mathbb{R}^d$ be the corresponding control points. Then, a PHT-spline can be defined as

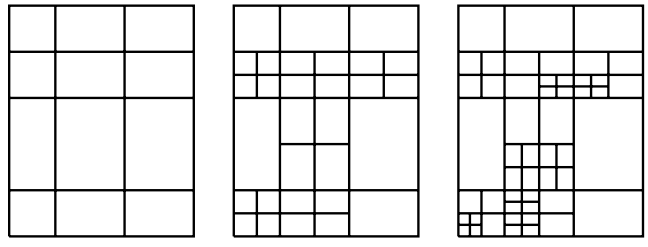
$$S(s, t) = \sum_{i=1}^{m} P_i b_i(s, t).$$

Given each control point a corresponding weight, $w_i > 0$, $w_i \in \mathbb{R}$, $i = 1, 2, \ldots, m$, then we can define control points $(w_i P_i, w_i) \in \mathbb{R}^{d+1}$, $i = 1, 2, \ldots, m$, denoted by $\{P_i^w\}_{i=1}^m$. A PHT-spline in $\mathbb{R}^{d+1}$ can be defined by the formula

$$S^w(s, t) = \sum_{i=1}^{m} P_i^w b_i(s, t).$$

By projecting this into the $\mathbb{R}^d$ space, a Rational PHT-spline (RPHT-spline) can be defined as

$$S(s, t) = \sum_{i=1}^{m} P_i R_i(s, t),$$



a. Level 0. b. Level 1. c. Level 2.

Fig. 2. A 2D hierarchical T-mesh.

where

$$R_i(s,t) = \frac{w_i b_i(s,t)}{\sum_{i=1}^{m} w_i b_i(s,t)}, \quad i = 1, 2, \ldots, m \tag{1}$$

are called the *RPHT-spline basis functions*.

If all the weights are equal to one, an RPHT-spline reduces to a PHT-spline.

It is obvious that the basis functions of an RPHT-spline have the same properties as those of the PHT-spline, such as linear independence, non-negativity, partition of unity, local compact support and C^1 global smoothness. These desirable properties make the RPHT-spline attractive for analysis.

4. IGA based on RPHT-splines

IGA based on NURBS and T-splines has been studied extensively. IGA consists in the framework of traditional FEA, except that Lagrange interpolating polynomial shape functions in FEA are now replaced by NURBS or T-splines basis functions. In this section, we will explore RPHT-splines for IGA within the current framework of IGA.

4.1. Building an RPHT-based geometry

Within the current framework of IGA, what we first need to do is to parameterize the physical domain Ω by a global geometry function G,

$G : \Omega_0 = [0, 1]^2 \to \Omega$,

$(s, t) \in \Omega_0 \to (x, y) \in \Omega$.

In this paper, we will apply an RPHT-spline to define the geometry function.

Suppose the computation domain Ω can be exactly modeled by a bi-cubic NURBS function $G : \Omega_0 \to \Omega$, which maps the interior of Ω_0 to the interior of Ω, and the boundaries of Ω_0 to the boundaries of Ω, denoted by

$$G(s,t) := \sum_{i=1}^{m_0} P_i R_i(s,t),$$

where $R_i(s,t)$ is the bi-cubic NURBS basis function with C^1 continuity. From the basis construction of RPHT-spline [18], at level 0, the basis functions of the RPHT-spline are as same as the NURBS basis functions. So at level 0, the NURBS-based geometry can be viewed as an RPHT-spline based geometry.

When this global geometry function from the parametric domain Ω_0 is built, it is unchanged when we divide a cell into four subcells by connecting the mid-points of the opposite edges. The geometry function $G(s,t)$ can be exactly represented with RPHT-spline basis functions at any level k.

Suppose at level k, the geometry can be represented by an RPHT-spline surface

$$G(s,t) := \sum_{i=1}^{m_k} P_i^k R_i^k(s,t), \tag{2}$$

where $R_i^k(s,t)$, $i = 1, 2, \ldots, m_k$, which are the same as given in (1) except for the superscript k, are the basis functions of the RPHT-spline at level k, $\{b_i^k(s,t)\}_{i=1}^{m_k}$ are the corresponding basis functions of the PHT-splines over the same T-mesh $\mathscr{T}_k$, and $\{P_i^k\}_{i=1}^{m_k} \in \mathbb{R}^2$ are control points at level k. The RPHT-spline at level k can be viewed as a PHT-spline in its projective space $\mathbb{R}^3$ by

$$G^w(s,t) := \sum_{i=1}^{m_k} (w_i^k P_i^k, w_i^k) b_i^k(s,t). \tag{3}$$

At level $k+1$, the basis functions of the PHT-spline $\{b_i^{k+1}(s,t)\}_{i=1}^{m_{k+1}}$ can be obtained according to the construction of the basis functions of the PHT-spline [18]. The control points $\{(w_i^{k+1}P_i^{k+1}, w_i^{k+1})\}_{i=1}^{m_{k+1}}$ at level $k+1$ can be determined in this way: the control points at the old basis vertices remain unchanged, while the control points at the new basis vertices are determined according to the initial geometry function $G(s,t)$. The process is as follows [18].

Define a linear operator

$$\mathcal{L}b(s,t) = (b(s,t), b_s(s,t), b_t(s,t), b_{st}(s,t)),$$

for any fixed basis vertex (s_0, t_0) with which four basis functions with indices j_1, j_2, j_3, j_4 are associated, one has

$$\begin{aligned}\mathcal{L}G(s_0,t_0) &= \sum_{s=1}^{N} C_s \mathcal{L}b_s(s_0,t_0) \\ &= \sum_{j=1}^{4} C_j \mathcal{L}b_j(s_0,t_0) = C \cdot B,\end{aligned} \tag{4}$$

where

$C = (C_{j_1}, C_{j_2}, C_{j_3}, C_{j_4})$

is a 3×4 matrix,

$B = (\mathcal{L}b_{j_1}(s_0,t_0), \mathcal{L}b_{j_2}(s_0,t_0), \mathcal{L}b_{j_3}(s_0,t_0), \mathcal{L}b_{j_4}(s_0,t_0))$

is a 4×4 matrix. From Eq. (4), we have

$$C = \mathcal{L}G(s_0,t_0) \cdot B^{-1}. \tag{5}$$

All the control points $\{(w_i^{k+1}P_i^{k+1}, w_i^{k+1})\}_{i=1}^{m_{k+1}}$ at level $k+1$ can be obtained. So (3) can be expressed by

$$G^w(s,t) = \sum_{i=1}^{m_k+1} (w_i^{k+1}P_i^{k+1}, w_i^{k+1}) b_i^{k+1}(s,t).$$

Now the geometry at level k can be expressed at level $k+1$ by

$$G^{k+1}(s,t) = \sum_{i=1}^{m_{k+1}} P_i^{k+1} R_i^{k+1}(s,t),$$

where

$$R_i^{k+1}(s,t) = \frac{w_i^{k+1} b_i^{k+1}(s,t)}{\sum_{i=1}^{m_{k+1}} w_i^{k+1} b_i^{k+1}(s,t)}$$

are the basis functions of the RPHT-spline at level $k+1$.

4.2. Isogeometric approximation

The current IGA is quite similar in its structure to traditional FEA, the only difference is the basis being used. To keep the representation of the analysis simple, we consider a two dimensional elliptic boundary value problem (BVP) as the model problem. In Section 4.2.1, we review some notations in FEA framework and we will discuss the isogeometric approximation with RPHT-splines in Section 4.2.2.

4.2.1. Model problem

The strong form of the BVP is as follows. Find $u : \overline{\Omega} \to \mathbb{R}^1$, such that

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \Gamma_D \\ \frac{\partial u}{\partial \mathbf{n}} &= h \quad \text{on } \Gamma_N, \end{aligned}$$

where Ω is a connected, bounded domain $\Omega \subset \mathbb{R}^2$ with a Lipschitz-continuous boundary $\partial\Omega \equiv \Gamma = \overline{\Gamma_D \cup \Gamma_N}$, $\Gamma_D \cap \Gamma_N = \phi$. $\mathbf{n}$ is the outward unit normal to Γ_N. We assume that Γ_D is closed relative to Γ and has a positive length, while f and h are square-integrable on Ω and Γ_N, respectively.

The technique of FEA begins by defining a weak, or variational formulation of the strong form of model BVP. It can be stated as follows. Given f and h, find $u \in V$, such that for all $v \in V$,

$$a(u, v) = l(v), \tag{6}$$

where $V = \{u | u \in \mathbf{H}^1(\Omega), u|_{\Gamma_D} = 0\}$, $\mathbf{H}^1(\Omega)$ is the Sobolev space that consists of those functions in $\mathbf{L}^2(\Omega)$ that possess weak and square-integrable derivatives. $a(u, v)$ is the symmetric bilinear form defined as

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v \mathrm{d}\Omega.$$

$l(v)$ is a linear functional defined as

$$l(v) = \int_\Omega f v \mathrm{d}\Omega + \int_{\Gamma_N} h v \mathrm{d}\Gamma.$$

With Cauchy–Schwarz inequality and Poincare inequality, the Lax–Milgram lemma [21] led to the well-posed result, and Eq. (6) has a unique solution $u \in V$.

Moreover, the symmetric bilinear form defines the so-called *energy norm* of u as

$$\|u\|_E = \sqrt{a(u, u)}, \quad \forall u \in V.$$

4.2.2. Isogeometric discretization with RPHT-splines basis functions

The function space V of the weak form (6) is an infinite dimensional space. In traditional FEA, a certain finite dimensional space

$$V^h = \{u | u \in \mathbf{H}^1(\Omega), u|_{\Gamma_D} = 0\}$$

must be constructed first and then to find $u^h \in V^h$, such that for all $v^h \in V^h$,

$$a(u^h, v^h) = l(v^h).$$

In conformal FEM, the finite dimensional subspace V^h must satisfy $V^h \subset V$. The basic convergence requirements are that the basis functions must be smooth, at least C^1 on each element interior, continuous across each element boundary and complete [22]. In IGA, basis functions are used to represent the geometry as well as to discretize of the field variables. By invoking the concept of isoparametric elements, the same basis functions are used to discretize the primal field variable u.

The basis functions of RPHT-spline have global C^1 continuity, partition of unity and local compact support. With the geometric map from the parametric domain to a physical domain, partition of unity and isoparametric elements can guarantee the completeness condition, while the smooth and continuous conditions are also satisfied. So the space spanned by the basis functions of the RPHT-spline can be used as the finite-dimensional approximation space V^h.

Suppose the geometry function is

$$G(s, t) = \sum_{i=1}^{m} P_i R_i(s, t).$$

From all the basis functions, $\{R_i(s, t)\}_{i=1}^n (n < m)$ are RPHT-spline basis functions that satisfy the condition

$$R_i(s, t)|_{G^{-1}(\Gamma_D)} = 0.$$

Then the finite dimensional function space V^h on physical domain Ω can be defined as

$$V^h = \mathrm{span}\{R_i(x, y) | R_i(x, y) = R_i \circ G^{-1}|_{\Gamma_D} = 0, i = 1, \ldots, n\}.$$

The isogeometric approximation of the weak form in Eq. (6) is given below: Find $u^h \in V^h$, such that for all $v^h \in V^h$

$$a(u^h, v^h) = l(v^h). \tag{7}$$

Thus, the approximate solution u^h can be written as

$$u^h(x, y) = \sum_{i=1}^{n} q_i R_i(x, y) = \sum_{i=1}^{n} q_i R_i \circ G^{-1}(x, y) \tag{8}$$

with unknown real coefficients q_i, $i = 1, 2, \ldots, n$.

Substituting Eq. (8) into (7), for every basis function R_i, $i = 1, 2, \ldots, n$, we have

$$\sum_{j=1}^{n} q_j a(R_j, R_i) = l(R_i). \tag{9}$$

Solving Eq. (9), the approximate solution u^h can be obtained from Eq. (8).

5. Implementation of procedure for the adaptive process with RPHT-splines

Considering the flexibility of the T-meshes and simple local refinement algorithm for RPHT-splines, we will discuss how to implement the adaptive analysis process. In FEA, an adaptive procedure consists of successive loops of the form

Solve → Estimate → Mark → Refine.

The essential part of the loops is the Estimate step. Error estimate methods with a posteriori error control have so far been well-developed in adaptive FEA, but little work on this has been done in IGA literature. The posteriori error estimate started with the pioneering work in [23]. Except for the rather complete description of the 1D situation by Babuška and Vogelius [23], convergence of the loops in the multi-dimensional case is still an open issue. The fundamental paper by Dörfler [24] for the Poisson equation shows a linear error reduction rate for the energy norm toward a preassigned tolerance in a finite number of steps.

We follow up on the ideas presented by Verfürth [19] and Ainsworth [20] to derive a residual-based error estimator based on RPHT-splines, and then discuss the marking and refinement strategies in this section.

5.1. A residual-based posteriori error estimator based on RPHT-splines

Let $\mathscr{T}$ denote the current hierarchical T-mesh on the parametric domain, $\mathscr{F}$ denote the corresponding T-mesh on the physical domain, and Φ denote all the cells in $\mathscr{T}$. For each cell $Q_k \in \Phi$, we can define a corresponding cell $G(Q_k) \subseteq \Omega$ in the physical domain, which is often called a *patch* in IGA literature. Let Ψ denote all the patches in the physical domain.

For each parametric element $\widehat{K} \in \Phi$, and each edge $\widehat{\gamma}$, we can define a corresponding cell and an edge as $K = G(\widehat{K}) \in \Psi$, $\gamma = G(\widehat{\gamma})$ in the physical domain, respectively.

Denote

$$r = f + \Delta u_h \quad \text{in } K,$$

and

$$R = h - \frac{\partial u_h}{\partial n} \quad \text{on } \partial K \cap \Gamma_N,$$

where r is often called *interior residual*, and R is called the *boundary residual*.

We have the following two-sided bounds on the error.

Theorem 1. *Let*

$$\eta_K^2 := \|r\|_{\mathbf{L}^2(K)}^2 h_K^2 + \|R\|_{\mathbf{L}^2(\partial K)}^2 h_K. \tag{10}$$

Then there exists a constant C depending on the regularity of the patches, such that

$$C^{-1} \|e_h\|_E^2 \leq \sum_{K \in \Psi} \eta_K^2 \tag{11}$$

and

$$\sum_{K\in\Psi}\eta_K^2 \le C\left\{\|e_h\|_E^2+\sum_{K\in\Psi}h_K^2\|f-\bar{f}\|^2_{\mathbf{L}^2(K)} + \sum_{\gamma\subset\Gamma_N}h_\gamma\|h-\bar{h}\|^2_{L^2(\gamma)}\right\}. \tag{12}$$

Moreover,

$$\eta_K^2 \le C\left\{\|e_h\|^2_{w_K}+h_K^2\|f-\bar{f}\|^2_{\mathbf{L}^2(K)} + \sum_{\gamma\subset\partial K\bigcap\Gamma_N}h_\gamma\|h-\bar{h}\|^2_{\mathbf{L}^2(\gamma)}\right\} \tag{13}$$

also holds, where $e_h = u-u_h$ is the error, $\bar{f}$ and $\bar{h}$ are approximations of f and h, respectively.

Remark 1. The value of the generic constant C need not be the same in any two places, even in the same equation. Strictly speaking, one should perhaps label the various constants $C_1, C_2, \ldots$. However, at the end of the day, what remains is nonetheless a generic constant whose value is not explicitly known. Thus, in order to avoid a proliferation of subscripts, we shall follow the common practice of not distinguishing between different unknown constants and instead prefer to denote both by the letter C.

Proof. For all $v\in V^h$, we have

$$a(u,v)=l(v)$$
$$a(u_h,v)=l(v),$$

thus $a(e_h,v)=0$.

Then, $\forall v\in V$, we have

$$a(e_h,v)=l(v)-a(u_h,v) = \sum_{K\in\Psi}\left(\int_K fv\mathrm{d}\Omega+\int_{\partial K\cap\Gamma_N}hv\mathrm{d}\Gamma-\int_K\nabla u_h\cdot\nabla v\mathrm{d}\Omega\right).$$

Using Green's formula on each patch shows that

$$a(e_h,v)=\sum_{K\in\Psi}\int_K(f+\Delta u_h)v\mathrm{d}\Omega+\sum_{K\in\Psi}\int_{\partial K\cap\Gamma_N}\left(h-\frac{\partial u_h}{\partial n}\right)\mathrm{d}\Gamma - \sum_{K\in\Psi}\int_{\partial K\setminus\Gamma_N}\frac{\partial u_h}{\partial n}v\mathrm{d}\Gamma.$$

For RPHT-splines, the approximate solution has global C^1 continuity, so the term $\sum_{K\in\Psi}\int_{\partial K\cap\Gamma_N}\frac{\partial u_h}{\partial n}v\mathrm{d}\Gamma$ disappears, and then

$$a(e_h,v)=\sum_{K\in\Psi}\left(\int_K rv\mathrm{d}\Omega+\int_{\partial K\cap\Gamma_N}Rv\mathrm{d}\Gamma\right). \tag{14}$$

$\forall v\in V$, let $\mathcal{I}_h v$ be the interplant of v from the subspace V^h, so

$$a(e_h,\mathcal{I}_h v)=0.$$

Thus

$$a(e_h,v)=a(e_h,v-\mathcal{I}_h v) = \sum_{K\in\Psi}\left(\int_K r(v-\mathcal{I}_h v)\mathrm{d}\Omega+\int_{\partial K\cap\Gamma_N}R(v-\mathcal{I}_h v)\mathrm{d}\Gamma\right).$$

According to the approximation theory [20], it is possible to choose the interplant $\mathcal{I}_h v$ such that

$$\|v-\mathcal{I}_h v\|_{\mathbf{L}^2(K)}\le Ch_K\|\nabla v\|_{\mathbf{L}^2(\omega_K)}, \tag{15}$$

and

$$\|v-\mathcal{I}_h v\|_{\mathbf{L}^2(\gamma)}\le Ch_K\|\nabla v\|_{\mathbf{L}^2(\omega_\gamma)}, \tag{16}$$

where C is a generic constant, that depends on the regularity of the patches on the physical domain Ω. h_K is the diameter of the patch K, h_γ is the length of the edge γ, ω_K is the set of all the patches that have common vertices with patch K, and ω_γ represents all the patches that share a common edge γ.

Applying the Cauchy–Schwarz inequality and using (15), (16), we have

$$\begin{aligned}a(e_h,v) &\le \sum_{K\in\Psi}(\|r\|_{\mathbf{L}^2(K)}\|v-\mathcal{I}_h v\|_{\mathbf{L}^2(K)}) + \sum_{K\in\Psi}(\|R\|_{\mathbf{L}^2(\partial K\cap\Gamma_N)}\|v-\mathcal{I}_h v\|_{\mathbf{L}^2(\partial K\cap\Gamma_N)})\\ &\le C\left\{\sum_{K\in\Psi}(\|r\|_{\mathbf{L}^2(K)}\|\nabla v\|_{\mathbf{L}^2(\omega_K)}h_K) + \sum_{\gamma\in\partial K\cap\Gamma_N}(\|R\|_{\mathbf{L}^2(\gamma)}h_\gamma^{1/2}\|\nabla v\|_{\mathbf{L}^2(\omega_\gamma)})\right\}\\ &\le C\|\nabla v\|_{\mathbf{L}^2(\Omega)}\left\{\sum_{K\in\Psi}(\|r\|^2_{\mathbf{L}^2(K)}h_K^2) + \sum_{\gamma\in\partial K\bigcap\Gamma_N}(\|R\|^2_{\mathbf{L}^2(\gamma)}h_\gamma)\right\}^{\frac{1}{2}}.\end{aligned}$$

Let $\|e_h\|_{E(K)}$ be the energy norm of the error on patch K. Therefore,

$$\|e_h\|_E\le C\left\{\sum_{K\in\Psi}\|r\|^2_{\mathbf{L}^2(K)}h_K^2+\sum_{\gamma\in\partial K\cap\Gamma_N}\|R\|^2_{\mathbf{L}^2(\gamma)}h_\gamma\right\}^{\frac{1}{2}}. \tag{17}$$

If we redefine $R=h-\frac{\partial u_h}{\partial n}$ on $\partial K\cap\Gamma_N$ as

$$R=\begin{cases}h-\dfrac{\partial u_h}{\partial n} & \text{on } \partial K\cap\Gamma_N\\ 0 & \text{on } \partial K\setminus\Gamma_N\end{cases} \tag{18}$$

then (17) can be rewritten as

$$\|e\|_E\le C\left\{\sum_{K\in\Psi}(\|r\|^2_{\mathbf{L}^2(K)}h_K^2+\|R\|^2_{\mathbf{L}^2(\partial K)}h_K)\right\}^{\frac{1}{2}}. \tag{19}$$

This (11) is proved.

Let $\bar{r}$ be the approximation of r from the subspace V^h, ϕ_K denote the interior bubble function defined on K. Then the function $v=\phi_K\bar{r}$ vanishes on the boundary of patch K, and therefore v can be extended to the whole domain as a continuous function by defining its value outside the patch to be zero. The resulting function, again denoted by v, belongs to the space V^h. Substituting v into (14) yields

$$a(e_h,\bar{r}\phi_K)=\int_K\phi_K r\bar{r}\mathrm{d}x,$$

and thus

$$\int_K\phi_K\bar{r}^2\mathrm{d}x=\int_K\phi_K\bar{r}(\bar{r}-r)\mathrm{d}x+a(e_h,\bar{r}\phi_K).$$

With the Cauchy–Schwarz inequality,

$$\int_K\phi_K\bar{r}^2\mathrm{d}x\le\|\phi_K\bar{r}\|_{\mathbf{L}^2(K)}\|\bar{r}-r\|_{\mathbf{L}^2(K)}+\|\phi_K\bar{r}\|_{\mathbf{L}^2(K)}\|e_h\|_{E(K)}.$$

Applying Theorem 2.2 [20], we have

$$\|\phi_K \bar{r}\|_{\mathbf{L}^2(K)} \leq C\|\bar{r}\|_{\mathbf{L}^2(K)}$$

and

$$\|\phi_K \bar{r}\|_{\mathbf{H}^1(K)} \leq Ch_K^{-1}\|\bar{r}\|_{\mathbf{L}^2(K)},$$

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

Therefore,

$$\int_K \phi_K \bar{r}^2 \mathrm{d}x \leq C\|\bar{r}\|_{\mathbf{L}^2(K)}\|\bar{r}-r\|_{\mathbf{L}^2(K)} + Ch_K^{-1}\|\bar{r}\|_{\mathbf{L}^2(K)}\|e_h\|_{E(K)},$$

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

With triangle inequality, we have

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

In an analogous fashion to the interior residual, we can estimate the boundary residual $\|R\|_{\mathbf{L}^2(\gamma)}$ from space V^h. Let $\bar{R}$ be the approximation to the boundary residual of R from subspace V^h, χ_γ denote the edge bubble function defined on edge $\gamma \subset \partial K$. Applying Theorem 2.4 [20], we can obtain the estimates

$$\|R\|_{\mathbf{L}^2(\gamma)} \leq C\{h_\gamma^{-1/2}\|e_h\|_E(\widetilde{\gamma}) + h_\gamma^{1/2}\|\bar{r}-r\|_{\mathbf{L}}^2(\widetilde{\gamma}) + \|\bar{R}-R\|_{\mathbf{L}^2(\gamma)}\}.$$

From the definition of the residual $r = f + \Delta u_h$ and (18), since $u_h \in V^h$, $r - \bar{r}$ and $R - \bar{R}$ reduce to $f - \bar{f}$ and $h - \bar{h}$, respectively, where $\bar{f}$ and $\bar{h}$ are approximations of f and h from the subspace V^h.

Thus,

$$\eta_K^2 = \|r\|_{\mathbf{L}^2(K)}^2 h_K^2 + \|R\|_{\mathbf{L}^2(\partial K)}^2 h_K \leq C\left\{\|e_h\|_{\omega_K}^2 + h_K^2\|f-\bar{f}\|_{\mathbf{L}^2(\omega_K)}^2 + \sum_{\gamma\subset\partial K\cap\Gamma_N} h_\gamma\|h-\bar{h}\|_{\mathbf{L}^2(\gamma)}^2\right\}.$$

Summing the η_K^2, we have

$$\sum_{K\in\Psi}\eta_K^2 \leq C\left\{\|e_h\|_E^2 + \sum_{K\in\Psi} h_K^2\|f-\bar{f}\|_{\mathbf{L}^2(K)}^2 + \sum_{\gamma\subset\Gamma_N} h_\gamma\|h-\bar{h}\|_{\mathbf{L}^2(\gamma)}^2\right\},$$

which proves (12) and (13), and we finish the proof of the theorem. □

Remark 2. Apart from constant C, all the quantities on the right-hand side can be computed explicitly from the given data and the approximation solution. The estimate (13) shows that the error indicator η_K is local in a certain sense, since the terms on the bound only involves patch K and its nearest neighbors. The estimate (11) shows that the given posteriori error estimator is reliable and (12) shows that the given posteriori error estimator is efficient.

5.2. Marking and refinement strategy

Some of the ways in which NURBS can be refined were discussed in [1]. For the present work, we only recall h-refinement. In adaptive analysis loops, we must mark the patches where the true errors are large before refinement.

The local bound is important for designing the adaptive algorithms, since it shows that the estimator gives some indication of the distribution of the true error in the sense that if the indicator η_K is large on a particular patch, then the true error must also be large in the neighborhood of the cell. Therefore, refining those patches in which the indicator is large will target refinements toward the regions where the true errors are large.

5.2.1. Marking strategy

The goal of marking step is to find the patches with large error for refinement. In order to seek the patches with large error efficiently, we adopt the marking strategy give in [25] with a slight modification. The strategy avoids sorting the local error indicators. The modification is that if a patch marked for refinement is in level $k+1$, and if there are patches in level k that are adjacent to the marked patches, we first refine the adjacent patches which are in level k. For RPHT-splines, keeping the level structure make the refinement process simple [18]. It should be noted that all the operations take place in the adjacent cells, and hence the operation is local.

Marking algorithm:

Given parameters $0 < \theta, \nu < 1$,
$\eta_{\max} := \max_{K\in\Psi}\eta_K$, $\eta^2 := \sum_{K\in\Psi}\eta_K^2$,
sum := 0,
$\mu := 1$,
while sum $\leq \theta^2\eta^2$ do
  $\mu := \mu - \nu$,
  for all K in Ψ do
    if K is not marked
      if $\eta_K > \mu\eta_{\max}$
        if there are adjacent patches in level less
        than marked patch/cell
          mark the adjacent paths for refinement,
        end if
        mark the patch K for refinement,
        sum := sum + η_K^2
        end if
      end if
  end for
end while

With the convergence analysis in [25], the adaptive algorithm produces a sequence of approximate solutions that converge to the exact solution.

5.2.2. Refinement strategy

For RPHT-splines, Refining the marked patch is cross insertion, which means we refine the marked patch by connecting the middle points of the opposite edges of the corresponding cell in parametric domain. With the global geometry function, the marked patch in physical domain is subdivided into four subcells.

5.3. Adaptive procedure

Implementing the adaptive scheme in this paper is as follows.

Step 1: Build an RPHT-based geometry function to represent the physical domain and construct the approximate solution space V^h.

Step 2: Solve the system of equations of the isogeometric approximation to calculate field variables.

Step 3: Calculate the local error indicator η_K patch by patch. If the total error η is less than the prescribed tolerance, then end.

Step 4: Mark the patches that contribute most to the total error with the marking strategy.

Step 5: Refine the marked patches according to the refinement strategy, and go to Step 2.

6. Numerical examples

In this section, we present the numerical results of our adaptive scheme for three 2D examples. All the examples have been implemented on a PC with an Intel Core 2 Quad @2.4 GHz processor and 2.0 Gb of memory.

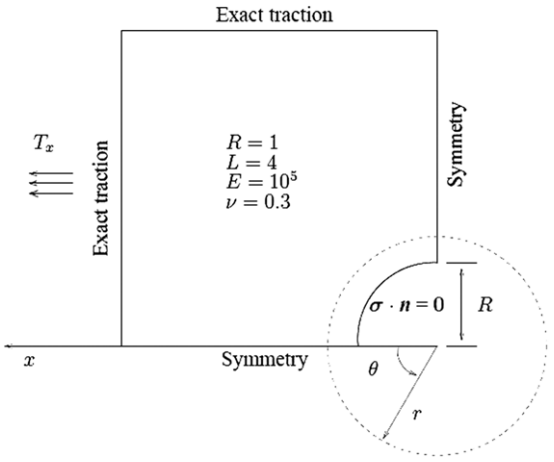


Fig. 3. Setup of an infinite elastic plate with a circular hole.

In every example, we compared the approximation solution, the computation time, the accuracy with our adaptive RPHT-splines and NURBS. Considering the implementation details, Adaptive process is much complicated. Generally, in order to get a given degree of freedom, the adaptive process with RPHT-splines takes up much more time, for the adaptive process may need much more iteration steps, additional time to estimate the error distribution, the time to mark and refine the patches. But for problems with prominent local features, adaptive method is advantageous compared to the uniform method. With good adaptive algorithm, fewer degree of freedom can reach a good accuracy. This is very desired properties in FEA. The numerical examples show the potential of RPHT-splines as basis for IGA, and the adaptive refinement process shows the reliability and efficiency of the given posteriori error estimator.

6.1. Linear elasticity: Infinite plate with circular hole under constant in-plane tension in the x-direction

This problem is a benchmark problem in IGA literature. The infinite plate is modeled by a finite quarter plate due to the symmetry. The setup is illustrated in Fig. 3. The exact solution evaluated at the boundary of the finite quarter plate, is applied as a Neumann boundary condition. Here, T_x is the magnitude of the applied stress for the infinity plate case, R is the radius of the hole, L is the length of the finite quarter plate, E is Young's modulus, and υ is Poisson's ratio.

At the coarsest level, the geometry is exactly represented with two patches by RPHT-splines. Fig. 4 shows the adaptive hierarchical T-meshes after 3, 5, 6, 13 refinements. The results show that the geometry can be exactly represented with the RPHT-splines at every refinement level. Fig. 5 gives the convergence results with NURBS and with RPHT-splines.

Table 1 shows the iteration number (n), degree of freedom (DOF), the CPU time and the x-direction stress error $\|\sigma_{\text{exact}} - \sigma^h\|_{\mathbf{L}^2(\Omega)}$ for NURBS and RPHT-splines. Fig. 5 and Table 1 both show that the adaptive refinement reaches a given tolerance with slightly fewer degree of freedom than the uniform refinement with NURBS. This is because the solution of this problem is regular with the most changes around the circular hole, which results in a more evenly spread refinement with a small biasing toward the hole.

This problem is largely governed by geometry and the concept of an exact encapsulation of the geometry at every level of mesh refinement is certainly attractive. In traditional FEA, this kind of domain is usually approximately represented by meshes. Some problems, such as thin shell analysis, and entropy layers problems in fluid mechanics [26–29] are sensitive to the geometry. If the geometry is approximate, a geometric error occurs, which can lead to accuracy problems. The geometry of the example given here can be exactly represented with RPHT-splines at any level, shows the potential of RPHT-splines for IGA.

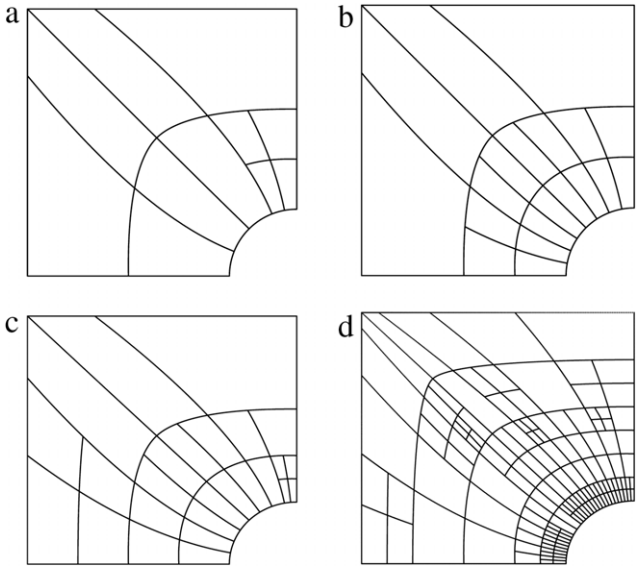


Fig. 4. Resulting meshes after 3, 5, 6 and 13 refinements.

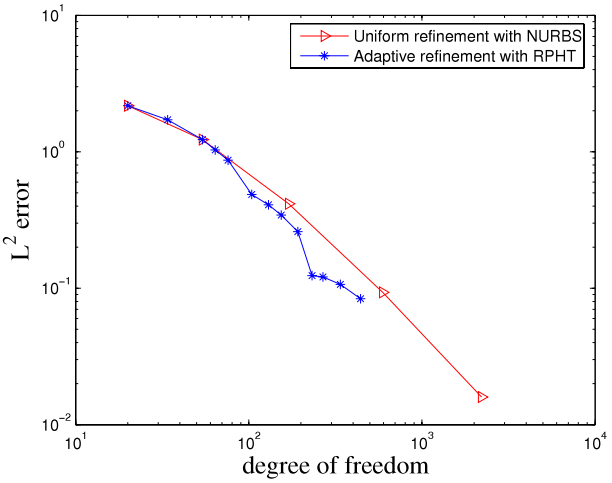


Fig. 5. Comparison of the exact L^2 error of x-direction stress.

Table 1
Experiment data. Here n is iteration number, DOF degree of freedom, t CPU time, and $\|\sigma_{\text{exact}} - \sigma^h\|_{\mathbf{L}^2(\Omega)}$ the x-direction stress error.

n	DOF	t(s)	Error
(a) RPHT-splines			
1	20	2.141	2.17809
3	54	36.22	1.22775
6	104	222.346	0.48715
13	442	2269.425	0.083725
(b) NURBS			
1	20	2.156	2.17809
2	54	25.221	1.22775
3	170	203.535	0.416571
4	594	1380.70	0.093252

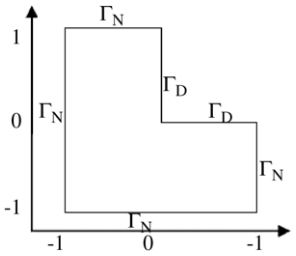


Fig. 6. Setup of the stationary heat conduction.

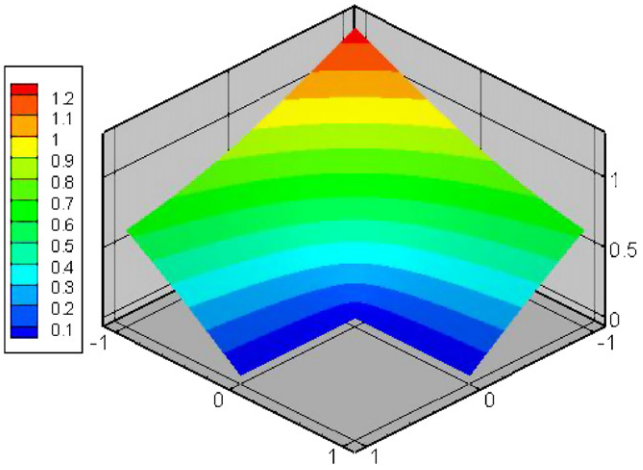


Fig. 7. The surface plot of the exact solution of the stationary heat conduction.

6.2. Stationary heat conduction: L-domain

The geometry of the domain $\Omega = [-1, 1] \times [-1, 1] \setminus [0, 1] \times [0, 1]$ is illustrated in Fig. 6. The control equation is

$$-\Delta u = 0 \quad \text{in } \Omega$$

with a homogeneous condition on Γ_D, and Neumann boundary conditions

$$\frac{\partial u}{\partial \mathbf{n}} = \frac{\partial f}{\partial \mathbf{n}} \quad \text{on } \Gamma_N,$$

where f is an exact solution of the problem.

Since the domain is concave, the solution is singular at the origin coordinate. The solution $u \in \mathbf{H}^s$ for $0 < s < \frac{2}{3}$. The surface plot of the exact solution is depicted in Fig. 7. At the coarsest level, the surface plot of the relative error $(u_{\text{exact}} - u^h)/\|u_{\text{exact}}\|_{\mathbf{L}^2(\Omega)}$ is depicted in Fig. 9. The relative error is larger around the singularity point (0, 0). In view of the approximation, the convergence rate in $\mathbf{H}^1-$ *norm* or *energy norm* is not better than h^s if we insist on uniform refinement of the mesh.

At the coarsest level, the geometry is exactly represented with two patches by RPHT-splines, see Fig. 8. The resulted mesh after

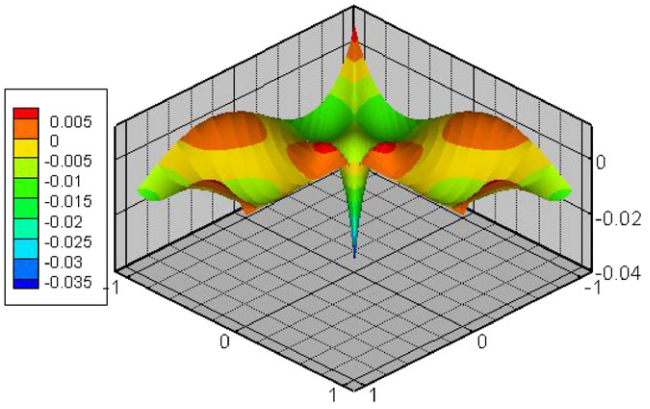


Fig. 9. The relative error at the coarsest level.

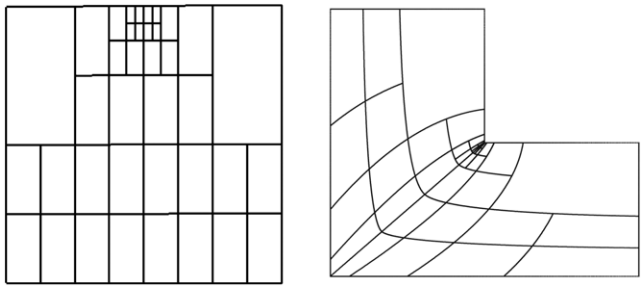


Fig. 10. Resulting T-meshes after 5 refinements, parametric domain (left) and the corresponding physical domain (right).

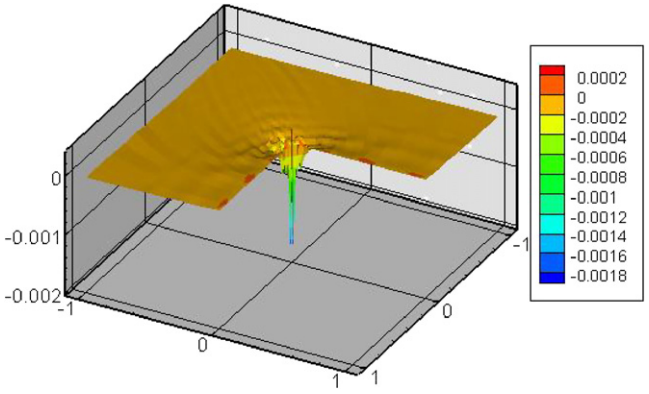
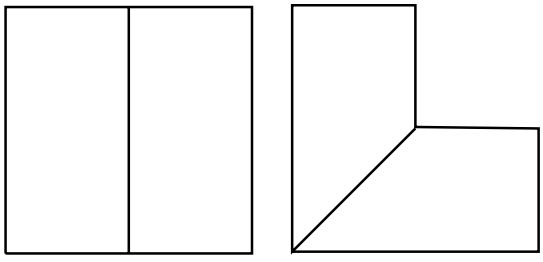


Fig. 11. The surface plot of relative error after 5 refinements with RPHT-splines.

5 refinements over the parametric domain and the corresponding physical meshes are given in Fig. 10, the surface plot of the corresponding relative error is given in 11. Fig. 12 gives the convergence results comparing with the adaptive refinements



(a) Parametric domain. (b) The corresponding physical domain.

Fig. 8. The initial meshes.

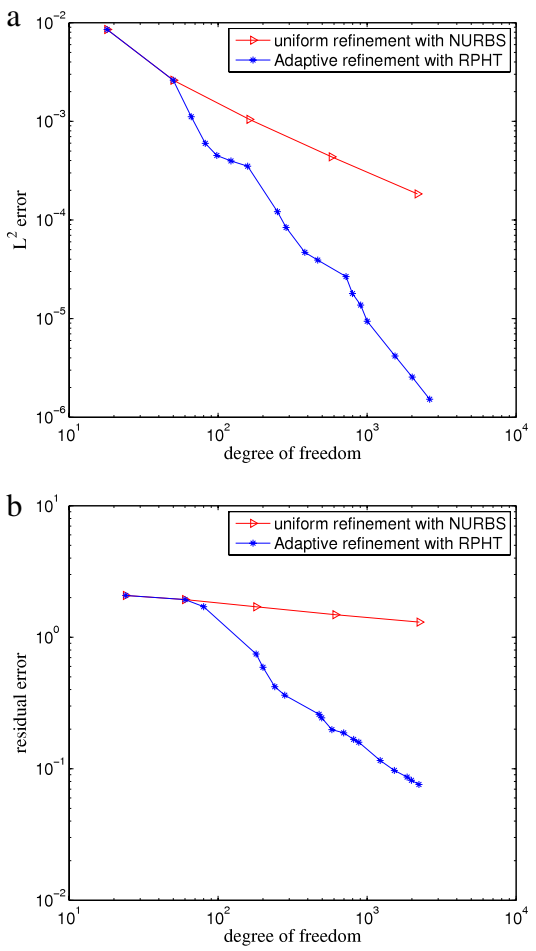


Fig. 12. Comparison of the convergence results with the adaptive and uniform refinement.

Table 2
Experiment data. Here n iteration number, DOF degree of freedom, t CPU time, and $\|u_{\text{exact}} - u^h\|_{\mathbf{L}^2(\Omega)}$ error.

n	DOF	t(s)	Error
(a) RPHT-splines			
1	18	0.157	0.0118779
2	50	0.734	0.0028047
3	66	1.843	0.00120436
6	142	8.188	0.000559688
9	370	25.235	0.000179648
(b) NURBS			
1	18	0.157	0.0118779
2	50	0.734	0.0028047
3	162	3.828	0.00102064
4	578	16.938	0.000414188
5	2178	135.734	0.000172815

with RPHT-splines and the uniform refinements with NURBS. Table 2 presents the iteration number, the degree of freedom, the CPU time and the error $\|u_{\text{exact}} - u^h\|_{\mathbf{L}^2(\Omega)}$ for RPHT-splines and NURBS. With our adaptive method, we can see that our adaptive refinement guided by the given error estimator detects the singularity, and reaches a given tolerance with much fewer degrees of freedom.

6.3. Biharmonic equation

PHT-splines are globally C^1 continuity. If the geometry function is smooth, then the RPHT-splines are still globally C^1 continuity.

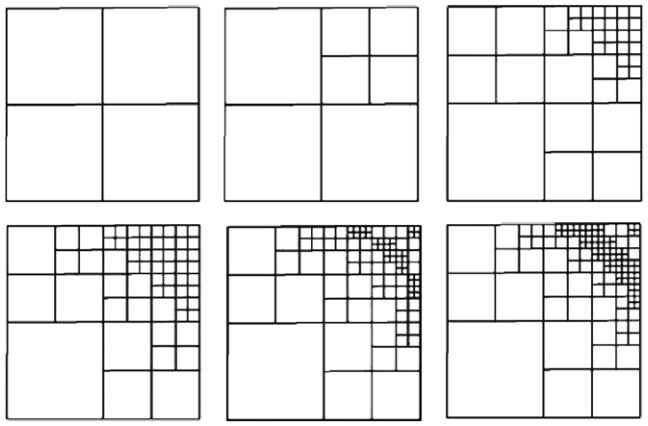


Fig. 13. The resulted adaptive meshes.

While solving higher order equations with the conformal finite element method, it is very reasonable to use the RPHT-splines space as the projection space. In traditional FEA, if we refine a quadrilateral element by subdividing a cell into four subcells, then hanging nodes (T-vertices) will appear. To keep the global smoothness, we need a way to constrain the degree of freedom of the hanging nodes, which is not a simple matter especially in a higher order continuity space. Using (rational) PHT-splines, we can avoid these problems, for there are no degree of freedoms at T-vertices. Here, we test a biharmonic problem defined on a unit square $\Omega = [0, 1]\times[0, 1]$ with zero Dirichlet boundary conditions,

$$\Delta^2 u = f \quad \text{in } \Omega$$
$$u = 0 \quad \text{on } \partial\Omega$$
$$\frac{\partial u}{\partial n} = 0 \quad \text{on } \partial\Omega$$

where the exact solution is

$$u(x, y) = 10^4 x^6 y^6 (1 - x)^2 (1 - y)^2.$$

In this example, the physical domain is $[0, 1] \times [0, 1]$, and the geometry function is identity. We may rewrite it as an RPHT-spline by selecting an appropriate knot vector and control points with all weights set to 1. In this case, the RPHT-splines reduce to PHT-splines. Fig. 13 gives a sequence of a adaptive refinement meshes.

Beginning with one patch, the surface plot of relative error after 1-refinement and 11-refinement are given in Fig. 14. Fig. 15 gives the convergence results of the adaptive and the uniform refinements. Table 3 presents the iteration number, the degree of freedom, the CPU time and the error $\|u_{\text{exact}} - u^h\|_{\mathbf{L}^2(\Omega)}$ with RPHT-splines and NURBS. We can see that our adaptive refinement detects the area with large error, and reaches a given tolerance with much fewer degree of freedoms with RPHT-splines than the uniform refinement with NURBS. The results show great potential for solving higher order PDE equations with (rational) PHT-splines in IGA.

7. Conclusions and future work

In this paper, we extended the PHT-splines to RPHT-splines, and used the RPHT-splines as the basis for IGA. We derived a residual-based posteriori error estimator and presented a corresponding marking and refinement strategy. The refinement process is simple, local and adaptive. The numerical results of some two dimensional analysis problems show the potential for RPHT as a basis for IGA. In the future, we will focus on the following work.

We will consider using a more general spline space $\mathbb{S}(m, n, \alpha, \beta, \mathscr{T})$ and 3D spline space for the analysis.

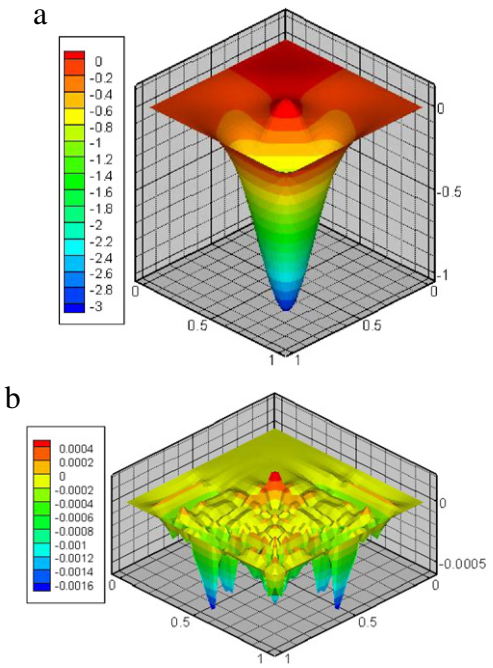


Fig. 14. The surface plot of the relative error after (a) 1 refinement (b) 11 refinements.

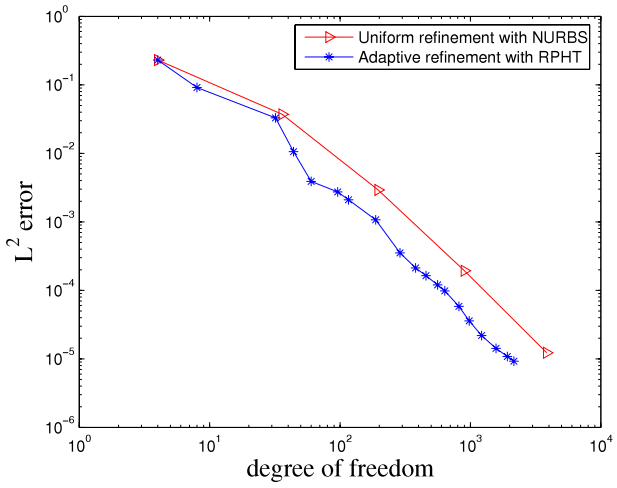


Fig. 15. Comparison of solution of the adaptive and uniform refinement.

Since the basis functions of the spline are not interpolation bases, how to deal with the non-zero Dirichlet boundary condition conveniently, efficiently and accurately needs to be investigated.

IGA is a direct design-to-analysis process that uses CAD geometry and basis functions for analysis without mesh generation. However, the construction of initial geometry function is time consuming, and the geometry function is not always unique. This raises the question of which function is more suitable for analysis. How to design the geometry automatically according to the analysis results is another issue that needs further investigation.

IGA for trimmed CAD surfaces [30,31] has been well studied. Xin Li provided a method [32] to create a watertight trimmed-NURBS models using untrimmed T-splines. [18] shows two or more PHT-splines can be stitched together. All these facts indicate that (rational) PHT-splines based IGA for trimmed NURBS geometries is hopeful. We will do some work on this issue in the future.

In adaptive analysis, a reliable and efficient posteriori error estimator is expected. Although there are many mature theories on posteriori error estimation in FEA, little work has been done on this issue in IGA. We thus need to ascertain what the difference is when the different posteriori error estimators given in the FEA literature perform with spline basis functions?

Table 3
Experiment data. Here n iteration number, DOF degree of freedom, t CPU time, and $\|u_{\text{exact}} - u^h\|_{L^2(\Omega)}$ error.

n	DOF	t(s)	Error
(a) RPHT-splines			
1	4	0.406	0.230135
2	8	0.75	0.0916619
5	68	3.39	0.00385517
10	548	20.499	0.000172061
12	940	37.625	5.01843×10^{-5}
(b) NURBS			
1	4	0.422	0.230135
2	36	1.375	0.037109
3	196	5.657	0.00292462
4	900	31.031	0.000193407
5	3844	4395.746	1.2262×10^{-5}

Acknowledgments

The authors are supported by a NKBRPC (2011CB302400), the NSF of China (11031007, 61073108, and 60873109), Program for New Century Excellent Talents in University (No. NCET-08-0514), and the 111 Project (No. b07033).

References

[1] Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD,finite elements, NURBS, exact geometry, and mesh refinement. Computer Methods in Applied Mechanics and Engineering 2005;194:4135–95.
[2] Cottrell JA, Hughes TJR, Bazilevs Y. Isogeometric analysis: toward integration of CAD and FEA. Chichester: Wiley; 2009.
[3] Bazilevs Y, Beirao de Veiga L, Cottrell JA, Hughes TJR, Sangalli G. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. Mathematical Models and Methods in Applied Sciences 2006;16:1031–90.
[4] Bazilevs Y, Calo VM, Cottrell JA, Hughes TJR, Reali A, Scovazzi G. Variational multiscale residual-based turbulencemodeling for large eddy simulation of incompressible flows. Computer Methods in Applied Mechanics and Engineering 2007;197:173–201.
[5] Bazilevs Y, Calo VM, Hughes TJR, Zhang Y. Isogeometric fluid–structure interaction: theory, algorithms, and computations. Computational Mechanics 2008;43:3–37.
[6] Cottrell JA, Hughes TJA, Reali A. Studies of refinement and continuity in isogeometric analysis. Computer Methods in Applied Mechanics and Engineering 2007;196:4160–83.
[7] Cottrell JA, Reali A, Bazilevs Y, Hughes TJR. Isogeometric analysis of structural vibrations. Computer Methods in Applied Mechanics and Engineering 2006; 195:5257–96.
[8] Sederberg TW, Zheng J, Bakenov A, Nasri A. T-splines and T-NURCCS. ACM Transactions on Graphics 2003;22(3):477–84.
[9] Sederberg TW, Cardon DL, Finnigan GT, North NS, Zheng J, Lyche T. T-spline simplification and local refinement. ACM Transactions on Graphics 2004; 23(3):276–2834.
[10] Döfel MR, Jütter B, Simeon B. Adaptive isogeometric analysis by local h-refinement with T-splines. Computer Methods in Applied Mechanics and Engineering 2010;199(58):264–75.
[11] Bazilevs Y, Calo VM, Cottrell JA, Evans JA, Hughes TJR, Lipton S, Scott MA, Sederberg TW. Isogeometric analysis using T-splines. Computer Methods in Applied Mechanics and Engineering 2010;199:229–63.
[12] Li X, Zheng J, Sederberg TW, Hughes TJR, Scott MA. On the linear independence of T-splines. Computer-Aided Geometric Design [submitted for publication].
[13] Scott MA, Li X, Sederberg TW, Hughes TJR. Local refinement of analysis-suitable T-splines. Computer Methods in Applied Mechanics and Engineering [submitted for publication].
[14] Piegl L, Tiller W. The NURBS book. 2nd ed. Monographs in visual communication, New York: Springer-Verlag; 1997.
[15] Buffa A, Cho D, Sangalli G. Linear independence of the T-spline blending functions associated with some particular T-meshes. Computer Methods in Applied Mechanics and Engineering 2010;199(23–24):1437–45.

[16] Li X, Scott MA. On the nesting behavior of T-splines. Computer-Aided Geometric Design [submitted for publication].
[17] Deng J, Chen F, Feng Y. Dimension of spline spaces over T-meshes. Journal of Computational and Applied Mathematics 2006;194:267–83.
[18] Deng J, Chen F, Li X, Hu C, Tong W, Yang Z, et al. Polynomial splines over hierarchical T-meshes. Graphical Models 2008;74(4):76–86.
[19] Verfürth R. A review of a posteriori error estimation and adaptive mesh-refinement techniques. Chichester (UK): Wiley-Teubner; 1996.
[20] Ainsworth M, Oden JT. A posteriori error estimation in finite element analysis. New York: Wiley-Interscience; 2002.
[21] Ciarlet P. The finite element method for elliptic problems. SIAM; 1978.
[22] Hughes TJR. The finite element method: linear static and dynamic finite element analysis. Englewood Cliffs (New Jersey): Prentice-Hall, Inc.; 1987.
[23] Babuška I, Vogelius M. Feedback and adaptive finite element solution of one-dimensional boundary value problems. Numerical Mathematics 1984;44: 75–102.
[24] Dörfler W. A convergent adaptive algorithm for Poisson's equation. SIAM Journal on Numerical Analysis 1996;33:1106–24.
[25] Morin P, Nochetto RH, Siebert KG. Data oscillation and convergence of adaptive FEM. SIAM Journal on Numerical Analysis 2000;38(2):466–88.
[26] Belytschko T, Stolarski H, Liu WK, Carpenter N, Ong JSJ. Stress projection for membrane and shear locking in shell finite elements. Computer Methods in Applied Mechanics and Engineering 1985;51:221–58.
[27] Barth TJ. Simplified numerical methods for gas dynamics systems on triangulated domains. Ph.D. thesis. Department of Aeronautics and Astronautics. Stanford University. 1998.
[28] Moin P. Fundamentals of engineering numerical analysis. Cambridge University Press; 2001.
[29] McMullen M, Jameson A, Alonso JJ. Application of a non-linear frequency domain solver to the Euler and Navier–Stokes equations. In: 40th AIAA aerospace sciences meeting and exhibit. 2002.
[30] Kim H-J, Seo Y-D, Youn S-K. Isogeometric analysis for trimmed CAD surfaces. Computer Methods in Applied Mechanics and Engineering 2009;198: 2982–95.
[31] Kim H-J, Seo Y-D, Youn S-K. Isogeometric analysis with trimming technique for problems of arbitrary complex topology. Computer Methods in Applied Mechanics and Engineering 2010;199:2796–812.
[32] Sederberg TW, Li X, et al. Watertight trimmed NURBS. In: Proceedings of ACM SIGGRAPH. ACM Transactions on Graphics 2008;27(3).