Geometry-Independent Field approximaTion: CAD-Analysis Integration, geometrical exactness and adaptivity

Gang Xu[a,*], Elena Atroshchenko[b], Weiyin Ma[c], Stéphane P.A. Bordas[d,e]

[a] *Hangzhou Dianzi University, Hangzhou 310018, P.R.China*
[b] *Department of Mechanical Engineering, University of Chile, Santiago, 8370448, Chile*
[c] *Department of Mechanical and Biomedical Engineering, City University of Hong Kong, Hong Kong*
[d] *Research Unit in Engineering, Faculté des Sciences, de la Technologie et de la Communication 6, University of Luxembourg, Luxembourg*
[e] *School of Engineering, Institute of Mechanics and Advanced Materials, Cardiff University, Cardiff CF24 3AA*

Abstract

In isogeometric analysis (IGA), the same spline representation is employed for both the geometry of the domain and approximation of the unknown fields over this domain. This identity of the geometry and field approximation spaces was put forward in the now classic 2005 paper [20] as a key advantage on the way to the integration of Computer Aided Design (CAD) and subsequent analysis in Computer Aided Engineering (CAE). [20] claims indeed that any change to the geometry of the domain is automatically inherited by the approximation of the field variables, without requiring the regeneration of the mesh at each change of the domain geometry. Yet, in Finite Element versions of IGA, a parameterization of the interior of the domain must still be constructed, since CAD only provides information about the boundary. The identity of the boundary and field representation decreases the flexibility in which this parameterization can be generated and somewhat constrains the modeling and simulation process, because an approximation able to represent the domain geometry accurately need not be adequate to also approximate the field variables accurately, in particular when the solution is not smooth. We propose here a new paradigm called *Geometry-Independent Field approximaTion* (GIFT) where the spline spaces used for the geometry and the field variables can be chosen and adapted independently while preserving geometric exactness and tight CAD integration. GIFT has the following features: (1) It is possible to flexibly

*Corresponding author

Email addresses: xugangzju@gmail.com; gxu@hdu.edu.cn (Gang Xu), eatroshchenko@ing.uchile.cl (Elena Atroshchenko), mewma@cityu.edu.hk (Weiyin Ma), stephane.bordas@uni.lu; bordasS@cardiff.ac.uk; stephane.bordas@alum.northwestern.edu (Stéphane P.A. Bordas)

choose between different spline spaces with different properties to better represent the solution of the problem, e.g. the continuity of the solution field, boundary layers, singularities, whilst retaining geometrical exactness of the domain boundary. (2) For multi-patch analysis, where the domain is composed of several spline patches, the continuity condition between neighboring patches on the solution field can be automatically guaranteed without additional constraints in the variational form. (3) Refinement operations by knot insertion and degree elevation are performed directly on the spline space of the solution field, independently of the spline space of the geometry of the domain, which makes the method versatile. GIFT with PHT-spline solution spaces and NURBS geometries is used to show the effectiveness of the proposed approach.
Keywords: Super-parametric methods, Isogeometric analysis (IGA), Geometry-independent Spline Space, PHT-splines, local refinement, adaptivity

1. Introduction

In Computer Aided Engineering (CAE), Computational Biomechanics, discretization of complex geometries through mesh generation for subsequent numerical analysis is, still today and despite recent fulgurant progress, costly in terms of human intervention. It was argued, e.g. in [20] that the geometrical information available from the Computer Aided Design (CAD) data is typically discarded when the mesh is generated.

A number of methods have been proposed over the past decade to use geometry information to build field approximations when solving partial differential equations (PDEs): subdivision surfaces [15], implicit surfaces and non-fitted meshes [28], [29] and isogeometric analysis (IGA) [20][1].

In all these cases, geometrical information has to be passed to the numerical approximation used to discretize the field variables. Isogeometric analysis was proposed by Hughes et al. in 2005 and focuses on enabling the seamless integration from Computer-Aided Design (CAD) to analysis. Current work on isogeometric analysis can be classified into six main categories:

1. mathematical foundations of IGA [6, 7, 8, 46];
2. isogeometric applications in multi-field problems [2, 14, 19];

[1] [25] proposes a relatively recent review on attempts to integrate CAD and Analysis

3. application of different spline models in isogeometric analysis [3, 9, 30, 26, 22, 21];
4. improving the accuracy and efficiency of IGA framework by refinement operations and parallel computing [1, 11, 13, 52];
5. constructing analysis-suitable parameterization of computational domain from given boundary data [47, 52, 53, 54].
6. isogeometric boundary element methods [34, 35, 36, 10].

For a detailed review of isogeometric analysis, readers can refer to [12][25].

In IGA, the same spline representation is employed for both the geometry and the field approximation. This has the potential to provide closer integration between geometric design and numerical analysis since any change in the CAD model is directly inherited by the field approximation. On the other hand, using the same spline spaces for both geometry and field approximation creates a constraint which may be unwanted, for example when the geometry spline space is not well-suited to approximate the solution of the Partial Differential Equation (PDE), in particular when local mesh refinement is required to capture the solution with limited computational resources. In isogeometric analysis with NURBS geometry, in order to use spline models with properties allowing local refinement, such as T-splines [3], PHT-splines [17, 30], LR B-splines[22] and Powell-Sabin splines [38], an effective method to convert the given NURBS geometry into the specified spline model is required as presented in [39]. However, in most cases, the exact conversion between different spline models is impossible.

In this paper, as a generalization of the isogeometric analysis method, a new discretization scheme is proposed where the spline spaces used for the geometry and the field solution can be chosen independently. We call the resulting method *Geometry-Independent Field approximaTion* (GIFT). This work is built upon the principle of "super-parameteric" or "generalized" IGA incepted in [27, 5] and presented in [50].

In the proposed method, for a given computational domain with spline form, the solution field can have a different spline representation, such as B-splines of different degree, PHT-splines, T-splines and generalized B-splines. Importantly, the geometry of the computational domain has the same spline representation as that of the given CAD boundary, which allows the method to preserve the potential to realize seamless integration of CAD and analysis. Our method therefore addresses three shortcomings of IGA, by enabling:

1. flexibility of solution representation;
2. straightforward treatment of multi-patch configurations;
3. independent field refinement.

The rest of the paper is organized as follows. The main idea and features of the GIFT method are described in Section 2. In Section 3, we present a general framework for geometry-independent field approximation method to general elliptical PDEs. The applications of GIFT method for two-dimensional heat conduction problems using different spline spaces is also presented in this section to show the effectiveness of GIFT by two numerical examples. With an emphasize on local refinement, we study the adaptive GIFT approach by using PHT-splines in Section 4. That is, the computational domain is represented by a NURBS model, while the solution field is formed in a PHT-spline space. Several examples are also presented to illustrate the effectiveness of the proposed GIFT method using PHT-splines. Finally, we conclude the paper and outline some future work in Section 5.

2. Main principles behind GIFT

In this section, we describe the main idea of the GIFT method.

Suppose that the computational domain $\Omega \in \mathbb{R}^d, d \geq 2$, defined on parametric domain $\mathcal{P}$ has the following spline representation:

$$\Omega = \{\boldsymbol{x} : \boldsymbol{x} = \boldsymbol{F}(\boldsymbol{\xi}), \boldsymbol{\xi} \in \mathcal{P}\}, \qquad \boldsymbol{F}(\boldsymbol{\xi}) = \sum_{\alpha=1}^{N} \boldsymbol{C}_\alpha N_\alpha(\boldsymbol{\xi})$$

where $\boldsymbol{C}_\alpha \in \mathbb{R}^d$ are control points, $N_\alpha(\boldsymbol{\xi})$ are tensor product spline basis functions defined over d given knot vectors.

The key idea of IGA is to represent the solution field $U^I(\boldsymbol{\xi})$ in the parametric domain using the same spline representation as that of the computational domain (see Fig. 1), i.e.

$$U^I(\boldsymbol{\xi}) = \sum_{\alpha=1}^{N} U_\alpha^I N_\alpha(\boldsymbol{\xi}),$$

where U_α^I, $\alpha = 0, 1, ..., N$ are the control variables to be solved for. Note that when h-refinement and p-refinement are performed, the solution field will have the same parameterization as that of the refined computational domain.

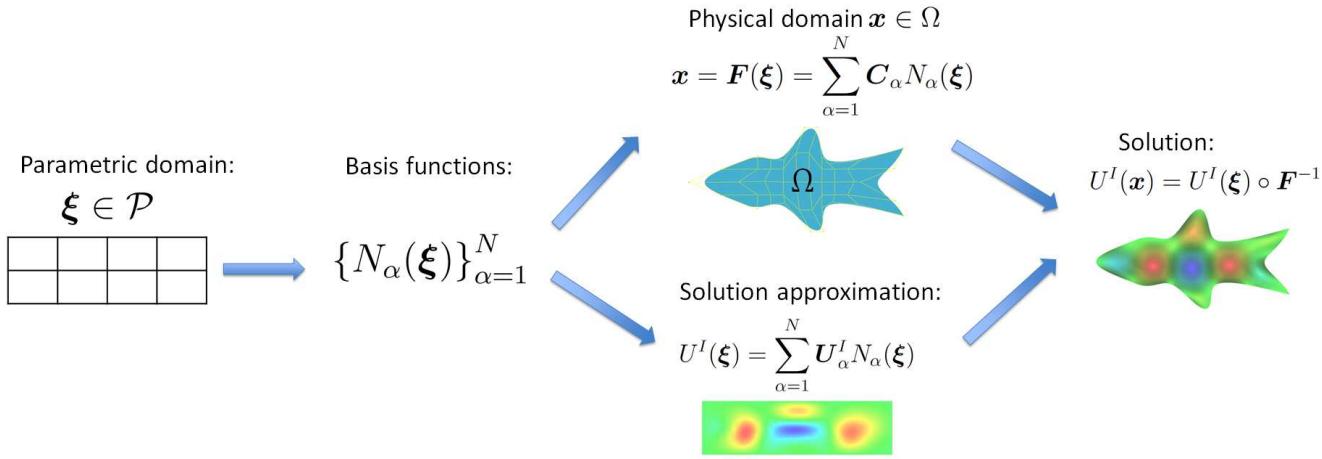

Fig. 1. The main idea of isogeometric analysis.

In the proposed geometry-independent field approximation method, the solution field $U^G(\boldsymbol{\xi})$ can have different spline representation as shown in Fig. 2:

$$U^G(\boldsymbol{\xi}) = \sum_{\beta=1}^{M} U_\beta^G M_\beta(\boldsymbol{\xi}),$$

where U_β^G, $\beta = 0, 1, ..., M$ are the control variables to be solved for and $M_\beta(\boldsymbol{\xi})$ are the basis functions of the specified spline space defined on the parametric domain $\mathcal{P}$, such as B-splines of different degree, PHT-splines, T-splines and generalized B-splines.

In GIFT, the computational domain Ω has the same spline representation as that of the given CAD boundary, hence it preserves the potential to realize a seamless integration of CAD and CAE. If $M_\alpha(\boldsymbol{\xi}) = N_\alpha(\boldsymbol{\xi})$, GIFT becomes IGA or, in other words, IGA is a subset of GIFT.

IGA and GIFT are compared in Table 1. The advantages of GIFT are summarized below:

- *Flexibility for solution representation*: It is possible to flexibly choose between different spline spaces with different properties to better represent the solution of the PDE, e.g. the continuity of the solution field, local strong gradient, etc.

- *Independent control of continuity conditions*: In case of multi-patches, the continuity condition between neighboring patches in the solution field can be automatically guaranteed without setting additional continuity constraints in the linear system.

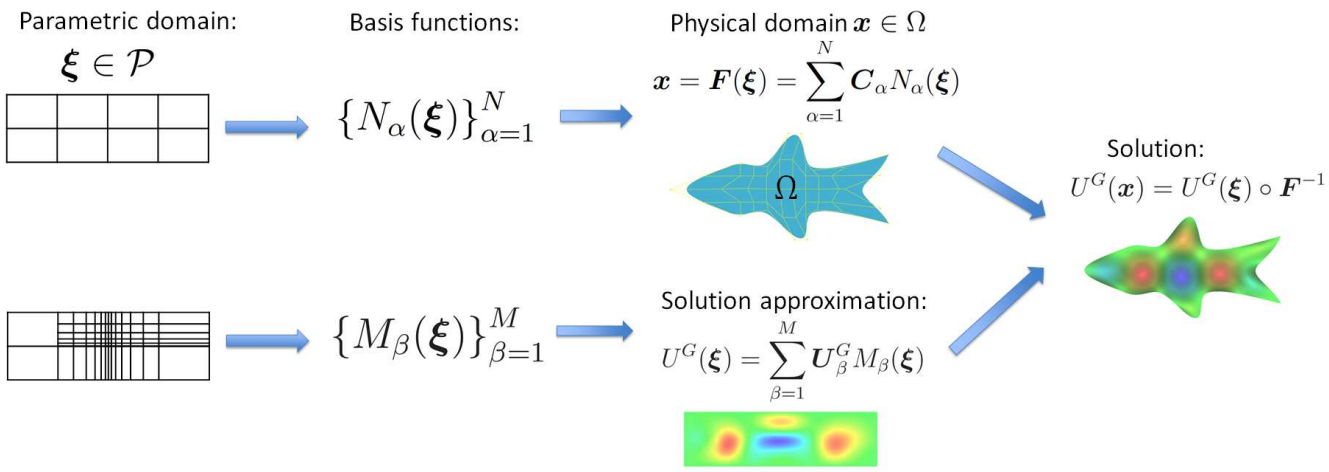

Fig. 2. The main idea of geometry-independent field approximation (GIFT).

	IGA	**GIFT**
Solution field	Same spline space as the geometry	Geometry-independent spline space
Computational domain	Varies during h/p-refinements	Fixed during h/p-refinement
Integration element	Knot span of the parametric domain of the computational domain	Knot span of the parametric domain of the solution field
Degree of freedom	Fixed by the computational domain	Flexible
Continuity of solution field	Fixed by the computational domain	Flexible

Table 1: Comparison of IGA and GIFT.

- *Independent field refinement*: Refinement operations by knot insertion and degree elevation are performed directly on the spline space of the solution field, independently of the spline space of the domain geometry, i.e. the parameterization of the given geometry is not altered during the refinement process. Hence, the initial design can be optimized in the subsequent shape optimization iterations without constraining the geometry discretization space to conform to the field approximation space, which is limiting [25].

- *Efficient processing with pre-computing*: The Jacobian of the transformation from the computational domain to the parametric domain, which is needed in setting up the stiffness matrix, can be pre-computed to save computational expense in GIFT.

3. A general framework for GIFT

3.1. The mathematical foundations of GIFT

In what follows we consider an open domain $\Omega \in \mathbb{R}^d$, $d \geq 2$ with boundary $\Gamma = \partial\Omega$, consisting of two parts Γ_D and Γ_N, such that: $\Gamma = \overline{\Gamma_D \bigcup \Gamma_N}$, $\Gamma_D \bigcap \Gamma_N = \emptyset$. The boundary value problem (strong form) for an elliptical partial differential operator of the second order $\mathcal{A}$ consists in finding the function $u : \overline{\Omega} \to \mathbb{R}$ such that:

$$\begin{aligned} \mathcal{A}\, u &= f, \qquad && \boldsymbol{x} \in \Omega \\ u &= 0, && \boldsymbol{x} \in \Gamma_D \\ \frac{\partial u}{\partial \boldsymbol{n}_A} &= g, && \boldsymbol{x} \in \Gamma_N, \end{aligned} \tag{1}$$

where $\mathcal{A}$ is given by

$$\mathcal{A}u = -\partial_j \left(a_{ij}(\boldsymbol{x})\partial_i u\right) + b_j(\boldsymbol{x})\partial_j u + c(\boldsymbol{x})u \tag{2}$$

and Neumann boundary condition $\dfrac{\partial u}{\partial \boldsymbol{n}_A}$ is defined by

$$\frac{\partial u}{\partial \boldsymbol{n}_A} = \boldsymbol{n}_j a_{ij}(\boldsymbol{x})\partial_i u \tag{3}$$

and $\boldsymbol{n}$ is a unit outward normal to Γ_N.

Next, the following functional space is defined:

$$H_0^1(\Omega) = \{u \in H^1(\Omega), u|_{\Gamma_D} = 0\} \tag{4}$$

The weak form of the problem (1) consists in finding $u \in H_0^1(\Omega)$, such that for any $v \in H_0^1(\Omega)$

$$a(u, v) = l(v), \tag{5}$$

where

$$\begin{aligned} a(u, v) &= \int\limits_{\Omega} \left\{(a_{ij}(\boldsymbol{x})\partial_i u \partial_j v + b_j(\boldsymbol{x})\partial_j uv + c(\boldsymbol{x})uv)\right\} d\Omega, \\ l(v) &= \int\limits_{\Omega} f(\boldsymbol{x})v(\boldsymbol{x})d\Omega + \int\limits_{\Gamma_N} g(\boldsymbol{x})v(\boldsymbol{x})d\Gamma \end{aligned} \tag{6}$$

Next we introduce the parameterization of the physical domain Ω on a parametric domain $\mathcal{P}$:

$$\boldsymbol{F} : \mathcal{P} \to \Omega, \quad \boldsymbol{x} = \boldsymbol{F}(\boldsymbol{\xi}) \tag{7}$$

Typically, geometrical map $\boldsymbol{F}$ is given by a set of spline basis functions $N_{i_1,i_2,...i_d}(\boldsymbol{\xi})$ and a set of control points $\boldsymbol{C}_{i_1,i_2,...i_d}$ as

$$\boldsymbol{F}(\boldsymbol{\xi}) = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} ... \sum_{i_d=1}^{n_d} \boldsymbol{C}_{i_1,i_2,...i_d} N_{i_1,i_2,...i_d}(\boldsymbol{\xi}), \tag{8}$$

where $N_{i_1,i_2,...i_d}(\boldsymbol{\xi})$ can be a tensor product of NURBS, B-splines, T-splines, PHT-splines, etc. For brevity eq.(8) is written as

$$\boldsymbol{F}(\boldsymbol{\xi}) = \sum_{\alpha=1}^{N} \boldsymbol{C}_\alpha N_\alpha(\boldsymbol{\xi}), \quad N = n_1 n_2 \cdots n_d. \tag{9}$$

In what follows we will need the Jacobian matrix $\boldsymbol{J}$ of the transform $\boldsymbol{F}$ given by

$$\boldsymbol{J}(\boldsymbol{\xi}) = \sum_{\alpha=1}^{N} \boldsymbol{C}_\alpha \frac{\partial N_\alpha(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \tag{10}$$

The main idea of the GIFT method is to seek for a solution in an independent on geometry spline space

$$V = \{u \in \text{span}\{M_{i_1,i_2,...i_d}(\boldsymbol{\xi})\} \circ \boldsymbol{F}^{-1}, u|_{\Gamma_D} = 0\}, \tag{11}$$

where $i_1 = 1..m_1, i_2 = 1..m_2, ...i_d = 1..m_d$ and $M_{i_1,i_2,...i_d}(\boldsymbol{\xi})$ can be a tensor product of NURBS, B-splines, T-splines, PHT-splines, etc.

The problem then consists in finding $u^G \in V$ such that for any $v^G \in V$

$$a(u^G, v^G) = l(v^G) \tag{12}$$

The solution is sought in the form

$$u^G(\boldsymbol{x}) = \sum_{\beta=1}^{M} U_\beta^G M_\beta(\boldsymbol{x}), \tag{13}$$

where $M_\beta(\boldsymbol{x}) = M_{i_1,i_2,...i_d}(\boldsymbol{\xi}) \circ \boldsymbol{F}^{-1}$, $M = m_1 m_2 \cdots m_d$ and U_β^G are unknown control variables. Then eq.(12) is transformed into a linear system

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}, \tag{14}$$

where the stiffness matrix $\boldsymbol{K}$ and the force vector $\boldsymbol{f}$ are given by:

$$K_{\gamma\beta} = a(M_\gamma(\boldsymbol{x}), M_\beta(\boldsymbol{x})) \tag{15}$$

$$f_\gamma = l(M_\gamma(\boldsymbol{x})) \tag{16}$$

and vector $\boldsymbol{u}$ consists of all unknown control variables U_β^G.

3.2. GIFT for Poisson's equation.

In this section, we demonstrate applications of GIFT method for two-dimensional heat conduction problems, in which the computational domain is parameterized by a planar NURBS surface, and the solution is sought in a different spline space. Although the presentation is done in the bivariate case, the trivariate case follows identical principles.

We consider a two-dimensional Poisson's equation with the homogeneous Dirichlet boundary condition assigned on the entire boundary, i.e.

$$\begin{aligned} -\boldsymbol{\Delta} u &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \partial\Omega \end{aligned} \tag{17}$$

where $\Omega \in \mathbb{R}^2$, $f(\boldsymbol{x}) \in L^2(\Omega) : \Omega \to \mathbb{R}$ is a given source term, and $u(\boldsymbol{x}) : \overline{\Omega} \to \mathbb{R}$ represents the unknown solution. The boundary value problem (17) corresponds to the boundary value problem described by eq.(1) if in eq.(2) $a_{ij}(\boldsymbol{x}) = \delta_{ij}$, $b_j(\boldsymbol{x}) = 0$, $c(\boldsymbol{x}) = 0$ i.e. $\mathcal{A} = -\boldsymbol{\Delta}$, and the corresponding weak form is given by eq.(5),(6) as:

$$\begin{aligned} a(u,v) &= \int_\Omega \nabla u \cdot \nabla v d\Omega, \\ l(v) &= \int_\Omega f(\boldsymbol{x}) v(\boldsymbol{x}) d\Omega \end{aligned} \tag{18}$$

The solution is sought in the form (13). Substituting $u = M_\gamma(\boldsymbol{x})$, $v = M_\beta(\boldsymbol{x})$ into (18) we arrive to the matrix equation:

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}, \tag{19}$$

where the stiffness matrix $K_{\gamma\beta}$ and force vector f_γ are given by:

$$\begin{aligned} K_{\gamma\beta} &= \int_\Omega \boldsymbol{\nabla} M_\gamma(\boldsymbol{x}) \, \boldsymbol{\nabla} M_\beta(\boldsymbol{x}) \, d\Omega \\ &= \int_{\mathcal{P}} \boldsymbol{B}(\boldsymbol{\xi})^T \nabla_{\boldsymbol{\xi}} M_\gamma(\boldsymbol{\xi}) \nabla_{\boldsymbol{\xi}} M_\beta(\boldsymbol{\xi}) \, \boldsymbol{B}(\boldsymbol{\xi}) \det|\boldsymbol{J}(\boldsymbol{\xi})| d\boldsymbol{\xi} \end{aligned} \tag{20}$$

$$f_\gamma = \int_\Omega f(\boldsymbol{x}) \, M_\gamma(\boldsymbol{x}) \, d\Omega = \int_{\mathcal{P}} f(\boldsymbol{\xi}) \, M_\gamma(\boldsymbol{\xi}) \, \det|\boldsymbol{J}(\boldsymbol{\xi})| d\boldsymbol{\xi}, \qquad \gamma, \beta = 1, ..., M$$

The Jacobian $\boldsymbol{J}(\boldsymbol{\xi})$ corresponds to the parameterization of domain Ω, as given by eq.(10) and $\boldsymbol{B}(\boldsymbol{\xi}) = \boldsymbol{J}^{-1}(\boldsymbol{\xi})$. Note, that these two matrices fully depend on the parameterization of the domain and h-refinement, p-refinement and k-refinement are performed only on the solution space.

In the following numerical examples we consider NURBS geometries, i.e. computational domain Ω, consisting of points $\boldsymbol{x} = (x, y)$, is represented by a planar NURBS surface defined by a set of control points $\boldsymbol{C}_\alpha = \mathbf{c}_{ij}$ and a set of $N = nm$ basis functions:

$$\boldsymbol{N}_\alpha(\boldsymbol{\xi}) = \frac{w_{ij} N_i(\xi)\, N_j(\eta)}{\sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij}\, N_i(\xi)\, N_j(\eta)}, \;\; i = 1, 2, ...n, \;\; j = 1, 2, ...m, \tag{21}$$

where $\boldsymbol{\xi} = (\xi, \eta) \in \mathcal{P}$ and

$$\boldsymbol{F}(\boldsymbol{\xi}) = (x(\xi, \eta), y(\xi, \eta)) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} \mathbf{c}_{ij} N_i(\xi)\, N_j(\eta)}{\sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij}\, N_i(\xi)\, N_j(\eta)},$$

where $N_i(\xi)$, $N_j(\eta)$ are classical B-spline basis functions associated with two knot-vectors:

$$\Xi = \{\xi_1, \xi_2, ...\xi_{n+p+1}\}, \;\; \mathcal{H} = \{\eta_1, \eta_2, ...\eta_{m+p+1}\}$$

and w_{ij} are the weights of the corresponding control points $\mathbf{c}_{ij}$.

In the following subsections, we demonstrate various numerical examples of (17), where representation of the temperature field in various geometry-independent spline space leads to the better accuracy of the solution.

3.3. Example of GIFT with polynomial B-splines: annulus.

The first example is an annulus region bounded by two concentric circles as shown in Fig. 3(a). Here the source function is

$$f(\boldsymbol{x}) = \frac{8 - 9\sqrt{x^2 + y^2}}{x^2 + y^2} \sin(2 \arctan(y/x)),$$

and the corresponding exact solution for problem (17) is

$$u(\boldsymbol{x}) = (x^2 + y^2 - 3\sqrt{x^2 + y^2} + 2) \sin(2 \arctan(y/x))$$

as shown in Fig. 3(c).

3.3.1. Pre-processing: construction of the solution spline space.

We consider an annulus region bounded by two concentric circles, described by a quadratic C^1 NURBS surface as shown in Fig.3(a) with 6×6 control points and knot vectors $\Xi = [0, 0, 0, 0.04, 0.2, 0.36, 1, 1, 1]$ and $\mathcal{H} = [0, 0, 0, 0.04, 0.2, 0.36, 1, 1, 1]$ on the parametric domain presented in Fig. 3(d).

The corresponding knot lines on the NURBS patch are shown in Fig. 3(b), which leads to a non-uniform sub-patch structure. In order to improve the accuracy and efficiency, we adopt the GIFT

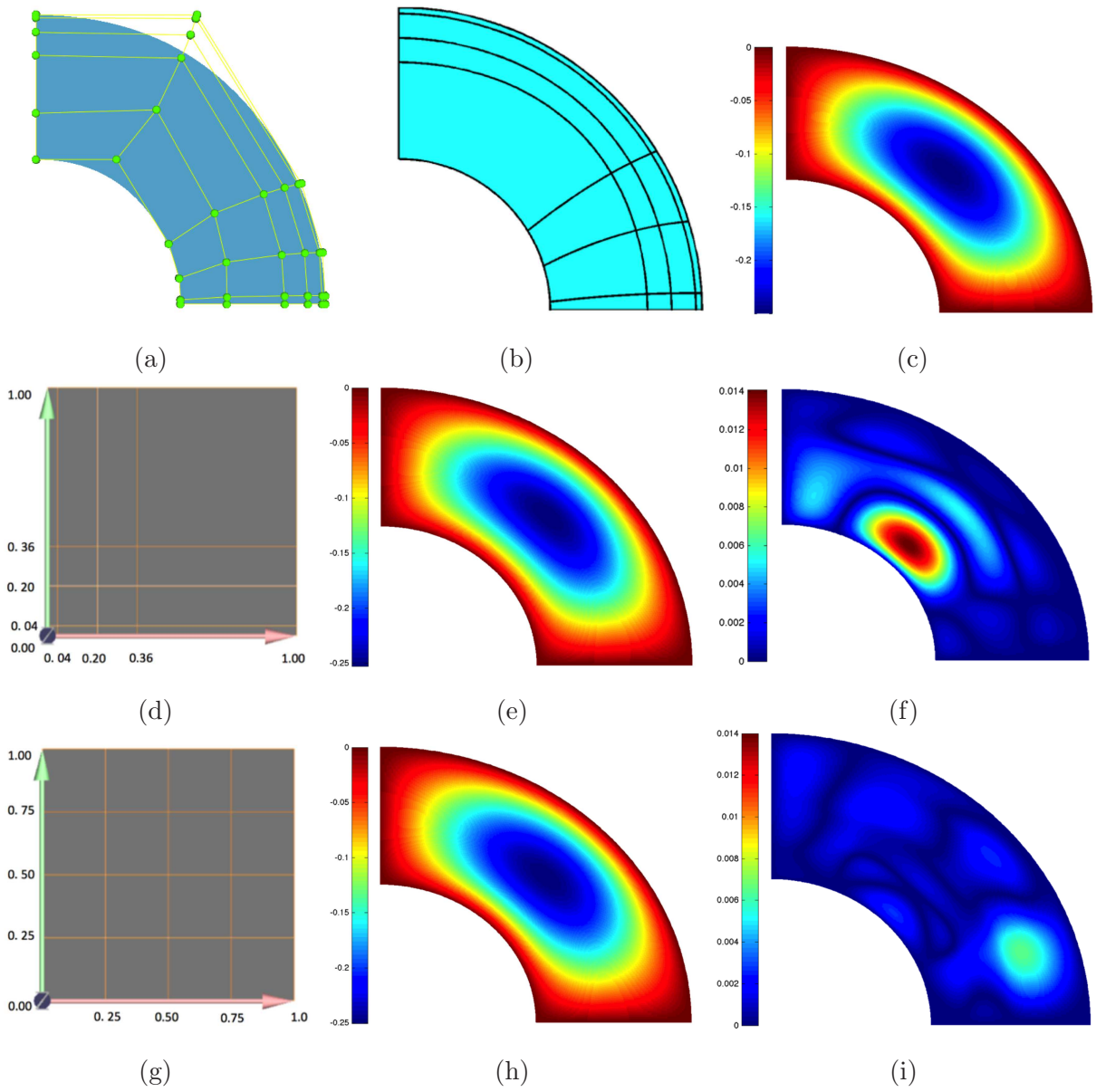

Fig. 3. Annulus example with different knot line structures: (a) NURBS computational domain with control mesh; (b) knot elements; (c) exact solution; (d) IGA parametric domain; (e) IGA solution with NURBS; (f) error color-map of IGA solution; (g) parametric domain of the solution field in GIFT; (h) GIFT solution with polynomial B-spline; (i) error color-map of the GIFT solution with the same scale as that in (f).

method, in which the solution is sought in the quadratic polynomial tensor-product B-spline space, formed by uniform knot vectors $\Xi = [0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1]$ and $\mathcal{H} = [0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1]$ as shown in Fig. 3(g). That is, the solution and the computational domain are parameterized on $\mathcal{P} = [0, 1] \times [0.1]$ but the knot structures are different.

3.3.2. Post-processing: visualization and error comparison

In the proposed GIFT method, the visualization of the simulation results is performed by combining the geometrical map $\boldsymbol{F}(\boldsymbol{\xi})$ with the GIFT solution $U^G(\boldsymbol{\xi})$. That is, for a sampling point $\boldsymbol{\xi}$ in the parametric domain, we compute the 3D coordinates $(\boldsymbol{F}(\boldsymbol{\xi}), U^G(\boldsymbol{\xi}))$, and the color-map over the computational domain $\boldsymbol{F}(\boldsymbol{\xi})$ is rendered according to the physical attribute $U^G(\boldsymbol{\xi})$.

In what follows we consider numerical results obtained by GIFT with those, obtained by quadratic NURBS-based IGA. From the error color-map shown in Fig. 3(f) and Fig. 3(i) , we can find that by using uniform partition of the parametric domain, GIFT method leads to improved accuracy than IGA for the same parameterization of the computational domain.

Quantitative data of the solution field for the annulus example in Fig. 3 is summerized in Table. 2. The quantitative data for the computational domain is the same as the solution information in the IGA method. Fig. 4 shows the comparison of the convergence results between the IGA method using quadratic NURBS and GIFT method using quadratic B-spline field with different knot structure during $h-$refinement.

3.4. Example of GIFT with NUAT B-splines: L-shaped plate.

The second example is the classical L-shape problem, in which the source function is defined as

$$f(\boldsymbol{x}) = 2\pi^2 \sin(\pi x) \sin(\pi y),$$

and the exact solution for problem (17) is

$$u(\boldsymbol{x}) = \sin(\pi x) \sin(\pi y)$$

as shown in Fig. 5(c).

3.4.1. Pre-processing: construction of the solution spline space.

Here the L-shape computational domain in Fig. 5(a) is represented as a planar C^1 quadratic B-spline surface with 6×10 control points, which is a particular case of NURBS parameterization

Table 2: Quantitative data of solution field for annulus example shown in Fig. 3. SSSF: spline space of solution field ; # DegF.: degree of the solution field; # ConF.: number of control variables in the solution field; Ξ: knot vector in ξ-direction of the solution space; $\mathcal{H}$: knot vector in η-direction of the solution space; MeanErr: mean value of error at sampling points; MaxErr: maximum value of error at sampling points.

	IGA	**GIFT**
SSSF	NURBS	polynomial B-spline
# DegF.	2×2	2×2
# ConF.	6×6	6×6
Ξ	$[0, 0, 0, 0.04, 0.2, 0.36, 1, 1, 1]$	$[0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1]$
$\mathcal{H}$	$[0, 0, 0, 0.04, 0.2, 0.36, 1, 1, 1]$	$[0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1]$
MeanErr	0.0022	0.001
MaxErr	0.0141	0.0065

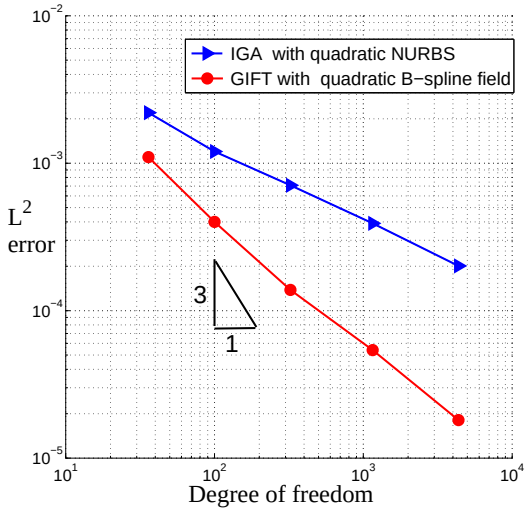

Fig. 4. Comparison of the convergence results with the IGA method using quadratic NURBS and GIFT method using cubic B-spline spaces during $h-$refinement in Fig. 3.

Table 3: Quantitative data of solution field in the L-shape example illustrated in Fig. 5. SSSF: spline space of solution field; # DegF.: degree of the solution field; # ConF.: number of control variables in the solution field ; Ξ: knot vector in ξ-direction of the solution space; $\mathcal{H}$: knot vector in η-direction of the solution space; MeanErr: mean value of error at sampling points; MaxErr: maximum value of error at sampling points.

	IGA	**GIFT**
SSSF	polynomial B-spline	NUAT B-spline
# DegF.	2×2	2×2
# ConF.	10×6	10×6
Ξ	$[0, 0, 0, 0.02, 0.1, 0.12, 0.2, 0.22, 0.3, 0.44, 1, 1, 1]$	$[0, 0, 0, 0.01, 0.07, 0.1, 0.2, 0.25, 0.38, 0.64, 1, 1, 1]$
$\mathcal{H}$	$[0, 0, 0, 0.04, 0.2, 0.36, 1, 1, 1]$	$[0, 0, 0, 0.02, 0.2, 0.46, 1, 1, 1]$
MeanErr	0.0321	0.0209
MaxErr	0.1475	0.1107

(3.2) with all weights $w_{ij} = 1$. As the expression for the source function involves the trigonometric functions, we propose to seek the GIFT-solution in the spline space of NUAT B-splines [49] spanned by $\{1, t, ..., t^{k-3}, \cos t, \sin t\}$ in which k is an arbitrary integer larger than or equal to 3. For the definition of NUAT B-splines with degree n, please refer to Appendix I. In this example we have chosen NUAT B-splines of degree 2 in the space spanned by $\{1, \cos t, \sin t\}$ with the knot structure shown in Fig. 5(g), which is different from the knot structure of B-spline parameterization of the computational domain illustrated in Fig. 5(d).

3.4.2. Post-processing: visualization and error comparison.

The solution obtained by IGA method using polynomial B-splines is presented in Fig. 5(e), and the solution obtained by GIFT method using NUAT B-splines is shown in Fig. 5(g). Improved accuracy is achieved by GIFT method from the error colormap shown in Fig. 5(f) and Fig. 5(i).

Table. 3 shows the quantitative data of the solution field for the L-shape example in Fig. 5. The comparison of the convergence results between the IGA using B-spline solution space and GIFT with NUAT B-spline solution space during $h-$refinement is presented in Fig. 6.

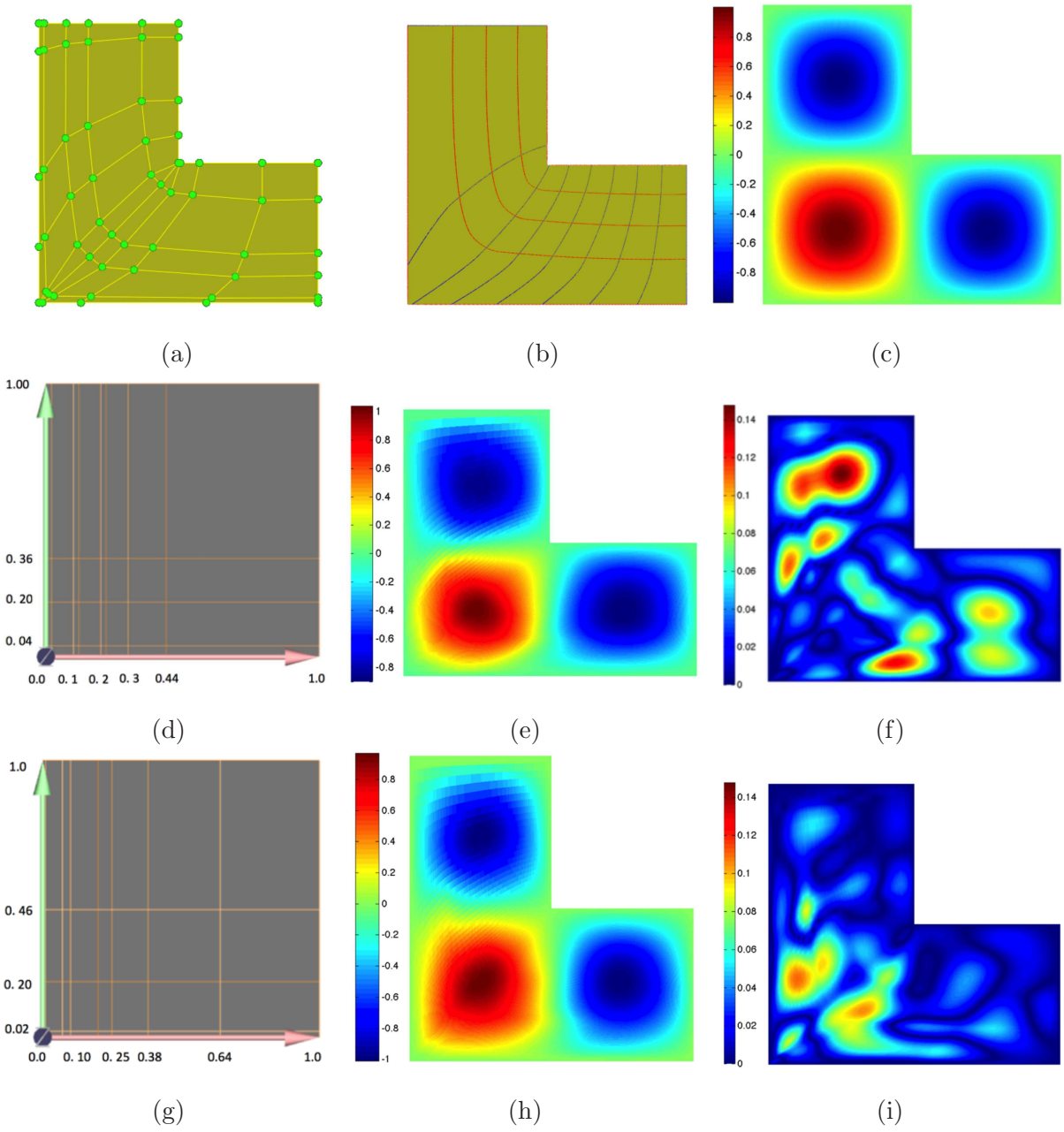

Fig. 5. L-shape example: (a) B-spline computational domain with control mesh; (b) knot elements; (c) exact solution; (d) IGA parametric domain; (e) IGA solution with B-spline form; (f) error color-map of IGA solution; (g) parametric domain of the solution field in GIFT; (h) GIFT solution with NUAT B-spline; (i) error color-map of the GIFT solution with the same scale as that in (f).

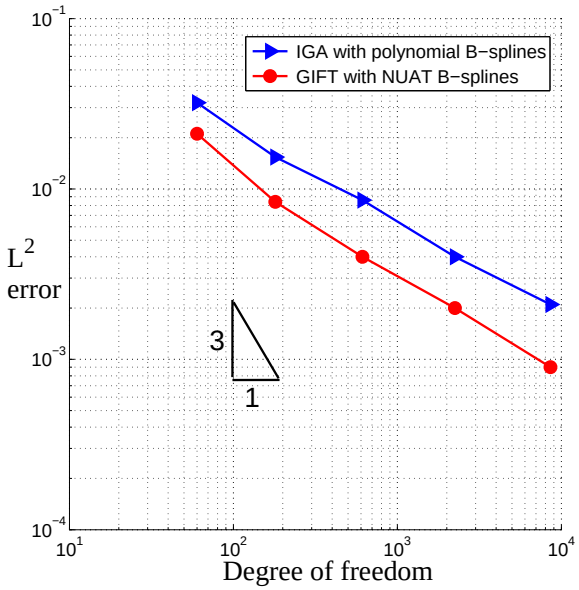

Fig. 6. Comparison of the convergence results with the IGA method using quadratic B-spline and GIFT method using quadratic NUAT B-spline field during $h-$refinement in Fig. 5.

4. Adaptive GIFT with PHT-splines

In this chapter we study adaptive GIFT approach with NURBS geometry and PHT-splines solution space.

4.1. PHT-splines

PHT-splines proposed by Deng et al. [17] are piecewise bicubic polynomials over a hierarchical T-mesh, which inherit the advantageous properties of T-splines. Unlike T-splines, PHT-splines are non-rational polynomial splines, and the refinement algorithm of PHT-splines is local and simple. The blending functions of PHT splines are linearly independent, an important property needed for finite element approximations.

A T-mesh is a rectangular partition of a planar domain with grid lines parallel to the boundary of the domain which allows T-junctions. In T-meshes, the end points of each grid line must lie on two other grid lines, and each cell or facet in the grid must be a rectangle. If a vertex is inside of the domain, it is called an *interior vertex*, otherwise, it is called a *boundary vertex*. There are two types of interior vertices, namely *crossing vertices*(i.e., it possesses valency 4) and *T-vertices* with valency 3, respectively.

Let $\Omega \in \mathbb{R}^m$ be a rectangular domain with boundary $\partial\Omega$. Denote by $\mathbb{T} = \cup\mathcal{K}$ a hierarchical T-mesh over domain Ω, where $\mathcal{K}$ is a cell of the mesh. We further define a spline space

$$S(p, q, \alpha, \beta, \mathbb{T}) := \{s(\xi, \eta) \in \mathcal{C}^{\alpha,\beta}(\Omega) | s(\xi, \eta) \in \mathbb{P}_{pq} \; for \quad any \quad element \; \mathcal{K} \in \mathbb{T}\},$$

where $\mathbb{P}_{pq}$ is the space of all the bivariate polynomials with degree (p, q), and the space $\mathcal{C}^{\alpha,\beta}(\Omega)$ consists of all continuous bivariate spline functions up to order α in the ξ-direction and order β in the η-direction. The dimension formula for the spline space $S(p, q, \alpha, \beta, \mathbb{T})$, with $p \geq 2\alpha + 1$ and $q \geq 2\beta + 1$, has already been provided in [17]. For the cubic PHT-spline space, the dimension formula can be written as

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

where V^b stands for boundary vertices and V^+ stands for interior crossing vertices. From the dimension formula, four basis functions are associated with each basis vertex (boundary vertex or crossing vertex), and they can be built with a hierarchical approach.

For the initial level, i.e., level 0 denoted as $\mathbb{T}_0$, the standard bi-cubic tensor-product B-splines are used as basis functions. For simplicity, we set the initial mesh to be a uniform rectangular grid. Suppose that the grid is $[\xi_1, \xi_2, \xi_3, ..., \xi_s] \times [\eta_1, \eta_2, \eta_3, ..., \eta_t]$. Since all the vertices are either crossing vertices or boundary vertices, there are four basis functions to be defined on any vertex (ξ_i, η_j). Each basis function at (ξ_i, η_j) has support $[\xi_{i-1}, \xi_{i+1}] \times [\eta_{j-1}, \eta_{j+1}]$. These four basis functions are defined to be B-spline basis functions with knots $[\xi_{i-1}, \xi_{i-1}, \xi_i, \xi_i, \xi_{i+1}]$ $\times$ $[\eta_{j-1}, \eta_{j-1}, \eta_j, \eta_j, \eta_{j+1}]$, $[\xi_{i-1}, \xi_i, \xi_i, \xi_{i+1}, \xi_{i+1}] \times [\eta_{j-1}, \eta_{j-1}, \eta_j, \eta_j, \eta_{j+1}]$, $[\xi_{i-1}, \xi_i, \xi_i, \xi_{i+1}, \xi_{i+1}] \times$ $\eta_{j-1}, \eta_j, \eta_j, \eta_{j+1}, \eta_{j+1}]$, $[\xi_{i-1}, \xi_{i-1}, \xi_i, \xi_i, \xi_{i+1}] \times [\eta_{j-1}, \eta_j, \eta_j, \eta_{j+1}, \eta_{j+1}]$, respectively, such that their function values and derivatives vanish outside $[\xi_{i-1}, \xi_{i+1}] \times [\eta_{j-1}, \eta_{j+1}]$.

In the GIFT framework with PHT-splines, the computational domain is in NURBS space whereas the solution field is in PHT-spline form. Firstly, we construct the parametric domain of the PHT-spline model to represent the solution field. We can then get the analytic expression of the initial solution field, in which the unknown control variables can be solved by the method presented in 3.2. By using an a-posteriori error estimation technique, the supporting cell with large errors in the parametric domain of the solution field can be marked, and local h-refinement is performed only on the solution field. Several local refinement steps can be performed until the desired error level is achieved.

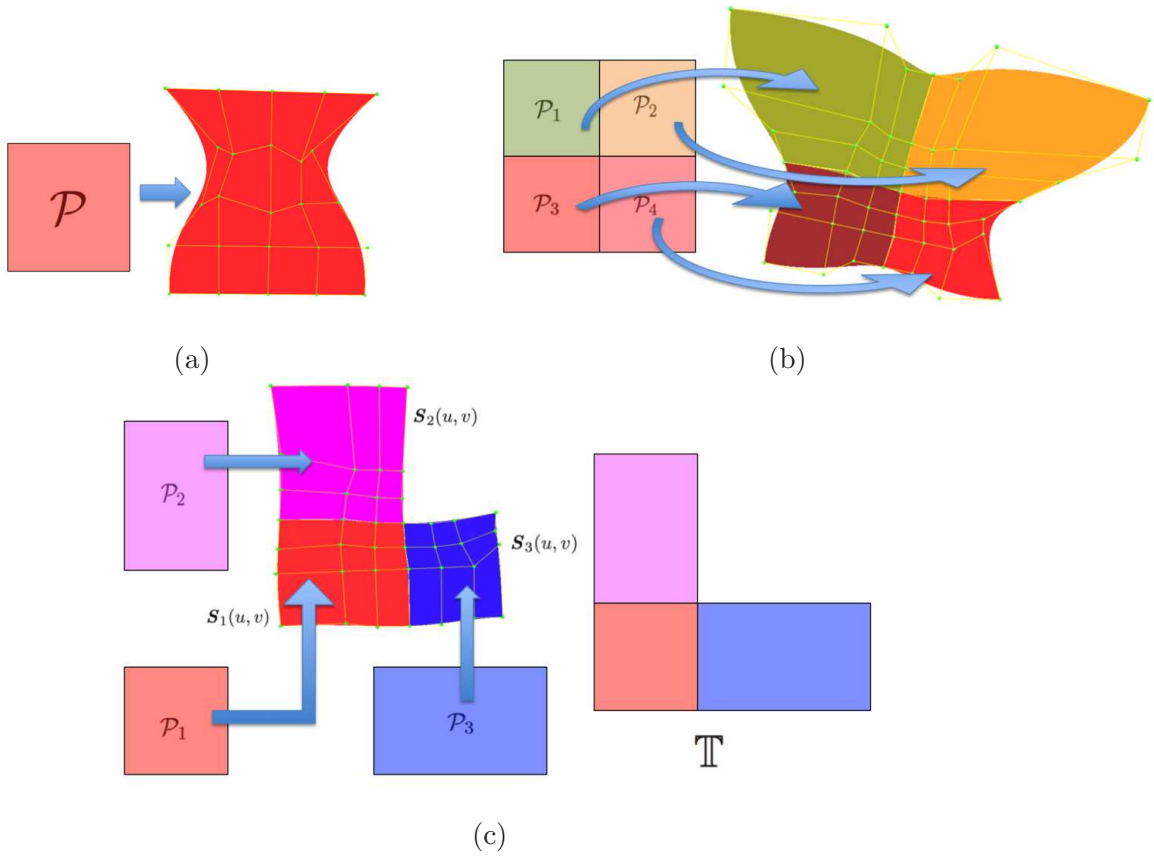

Fig. 7. Parametric T-mesh $\mathbb{T}$ construction for GIFT with PHT-splines: (a) case of a single NURBS patch; (b) case of multi-patch in which the parametric domain of each patch forms a quad-mesh; (c) other cases in which reparameterization is required.

4.2. Pre-processing

As a first step, we should construct a parametric domain of the solution field. This can be seen a preprocessing stage. Depending on the parametric domain of the NURBS patches under consideration, two kinds of operations are proposed in this step:

- If the computational domain is made of a single NURBS patch (Fig. 7(a)) or multi-patches in which the parametric domain of each patch forms a quad-mesh as presented in Fig. 7(b), then the parametric domain of the solution field is constructed as the partition-mesh formed by the knot lines on the parametric domain of the NURBS patches.

- For other cases, such as the examples shown in Fig. 7(c), we should construct the initial parametric mesh $\mathbb{T}$ according to the topological connection information of the NURBS patches. A reparameterization operation should be performed in this case as described below.

Here we present an example to show the reparameterization method. Suppose that the patch $\boldsymbol{S}_1(u,v)$ in Fig. 7(c) has parametric domain $\mathcal{P}_1 = [a_1, b_1]\times[c_1, d_1]$, the patch $\boldsymbol{S}_2(u,v)$ has parametric domain $\mathcal{P}_2 = [a_1, b_1]\times[c_2, d_2]$, and the patch $\boldsymbol{S}_3(u,v)$ has parametric domain $\mathcal{P}_3 = [a_3, b_3]\times[c_1, d_1]$. Then according to the topological information of all three patches, the parametric domain for the PHT-spline solution field should be constructed as shown in Fig. 7(c). In other words, the parametric domain of $\boldsymbol{S}_1(u,v)$ does not change, while the parametric domain of $\boldsymbol{S}_2(u,v)$ changes to $[a_1, b_1] \times [d_1, d_1 + (d_2 - c_2)]$, and the parametric domain of $\boldsymbol{S}_3(u,v)$ changes to $[b_1, b_1 + (b_3 - a_3)] \times [c_1, d_1]$.

In order to maintain the geometry of the patches during the transformation of parametric domain, a reparameterization technique should be adopted to obtain the new parametric representation of each patch. Suppose that the initial parametric domain of $\boldsymbol{S}(u,v)$ is $[a, b] \times [c, d]$, we can then use the following parameter transformation to achieve a new parameterization $\boldsymbol{F}(\xi,\eta)$ with parametric domain $[e, f] \times [g, h]$

$$u(\xi,\eta) = \frac{1}{f-e}[(f-\xi)a + (\xi-e)b] \tag{22}$$

$$v(\xi,\eta) = \frac{1}{h-g}[(h-\eta)c + (\eta-g)d] \tag{23}$$

Remark. There is actually no need to derive the explicit parametric representation of the reparameterized surface $\boldsymbol{F}(\xi,\eta)$. The derivative information required in the solving stage, as shown in (20), can be evaluated from the original parameterization $\boldsymbol{S}(u,v)$ through the Jacobian transformation matrix.

After the planar T-mesh $\mathbb{T}$ is constructed as the parametric domain of PHT-splines, we can write the initial formula of the solution field in PHT-spline form as follows,

$$\mathcal{T}(\boldsymbol{\xi}) = \sum_{i=1}^{4n} T_i\, M_i(\boldsymbol{\xi}), \tag{24}$$

where n is the number of basis vertices on $\mathbb{T}$, T_i, $i = 1, 2, \cdots, 4n$, are the control variables to be solved.

4.3. Solving and local refinement

After constructing PHT-spline representation of the solution field, the control variables in (24) is obtained by solving the heat-conduction problem as introduced in Section 3.2 .

Algorithm 1 Local refinement scheme in GIFT using PHT-splines

Input: Planar NURBS parameterization $\boldsymbol{F}(\xi,\eta)$ of computational domain Ω

Output: PHT-spline solution field

1: Compute the PHT-spline solution $\mathcal{T}(\xi,\eta)$ of model problem (17) over the given NURBS parameterization $\boldsymbol{F}(\xi,\eta)$ by GIFT .

2: Calculate the local error indicator $e_{\mathcal{K}}$ patch by patch on the solution field (See Section 4.3.1)

3: Mark the parametric cells to be refined by a mean-value marking algorithm (See Section 4.3.2)

4: Subdivide the marked cells into four sub-celles on the parametric domain of the PHT-spline field.

5: Construct the PHT-spline basis functions over the refined T-mesh of the parametric domain.

6: Compute the new solution field $\mathcal{T}(\xi,\eta)$ in the refined PHT-spline space.

7: Repeat the above refinement steps until the estimated error is less than a given threshold.

In order to obtain a solution with required accuracy, refinement operations are often needed to increase the approximation power of the solution space locally and thereby optimize the computational expense required for a given accuracy level. Contrary to the classical isogeometric analysis, the refinement operation in GIFT is only performed on the solution field, while the parameterization of the computational domain remains the same during the refinement process. Choosing a PHT-spline approximation offers a natural local refinement. Local refinement is performed on the sub-patches of the solution field on which the local error indicator is larger than a given marking threshold. The corresponding working flow is summerized in Algorithm. 1.

4.3.1. Residual-based error indicator

Suppose that T_h is the PHT-spline solution of the problem (17) by using the GIFT method, and T is the exact solution. Let $e_h = T - T_h$ be the error of the GIFT approximation T_h. As in GIFT the refinement operation is only performed on the solution field, in order to determine the parametric cell to be refined, we should give an error indicator on each parametric cell of the T-mesh in the parametric domain of the solution field rather than an error indicator on the sub-patch with respect to the knot span on the NURBS parameterization $\boldsymbol{F}(\xi,\eta)$.

Algorithm 2 Marking Algorithm. (Mean-value strategy)

Input: Parametric T-mesh $\mathbb{T}$ of solution field, error estimates $e_{\mathcal{K}}$ for all parametric cells $\mathcal{K} \in \mathbb{T}$.

Output: A subset $\tilde{\mathbb{T}}$ of marked parametric cells to be refined

1: Compute

$$e_{\mathbb{T},mean} = \frac{\sum_{\mathcal{K}\in\mathbb{T}} e_{\mathcal{K}}}{N},$$

where N is the number of parametric cells in $\mathbb{T}$.

2: If

$$e_{\mathcal{K}} \geq e_{\mathbb{T},mean}$$

mark $\mathcal{K}$ for refinement and put it into the set $\tilde{\mathbb{T}}$.

Suppose that $\mathcal{K}$ is the parametric cell on the T-mesh $\mathbb{T}$ of the parametric domain $\mathcal{P}$ for the PHT-spline solution field T_h. The residual-based a posteriori error estimate $\|e_h\|^2_{\mathcal{P}}$ over the parametric domain $\mathcal{P}$ proposed in [52] can be rewritten as follows:

$$\|e_h\|^2_{\mathcal{P}} \leq C \sum_{\mathcal{K}\in\mathbb{T}} h^2_{\mathcal{K}} \|f(\boldsymbol{x}) + \Delta T_h(\boldsymbol{x})\|^2_{L^2(\mathcal{K})} \tag{25}$$

where $\boldsymbol{x} = \boldsymbol{F}(\xi,\eta) = (x(\xi,\eta), y(\xi,\eta))$, C is a positive constant, and $h_{\mathcal{K}}$ is the diameter of the sub-patch in the NURBS parameterization $\boldsymbol{F}(\xi,\eta)$ of Ω with respect to the parametric cell $\mathcal{K}$ in the parametric domain $\mathcal{P}$.

From (25), we can obtain the local error indicator $e_{\mathcal{K}}$ on each parametric cell $\mathcal{K}$ as follows

$$e_{\mathcal{K}} = h^2_{\mathcal{K}} \|f(\boldsymbol{x}) + \Delta T_h(\boldsymbol{x})\|^2_{L^2(\mathcal{K})}. \tag{26}$$

If the parametric cell $\mathcal{K}$ is written as $[\xi_0,\xi_1]\times[\eta_0,\eta_1]$, we have then

$$h_{\mathcal{K}} = \int_{\xi_0}^{\xi_1} \boldsymbol{F}(\xi,\eta_0)d\xi + \int_{\xi_0}^{\xi_1} \boldsymbol{F}(\xi,\eta_1)d\xi + \int_{\eta_0}^{\eta_1} \boldsymbol{F}(\xi_0,\eta)d\eta + \int_{\eta_0}^{\eta_1} \boldsymbol{F}(\xi_1,\eta)d\eta. \tag{27}$$

4.3.2. Marking strategies

The local refinement in GIFT requires a marking strategy to decide which elements should be refined. That is, we should determine a subset $\tilde{\mathbb{T}}$ of the parametric T-mesh $\mathbb{T}$ consisting of all those parametric cells $\mathcal{K}$ that must be refined due to too large an $e_{\mathcal{K}}$ value. We use the mean-value strategy shown in Algorithm. 2 to determine the set $\tilde{\mathbb{T}}$.

For some physical problems, wild distribution of estimated errors may be observed with (I) very few cells having extremely large estimated errors, (II) some cells having extremely small

estimated errors, and (III) other cells having estimated errors that are much smaller than that of (I) while much larger than that of (II). We group the cells into three groups for (I), (II) and (III), respectively. In this case, if only the parametric cells in the first group will be refined by using the above marking algorithm, it would deteriorate the performance of the local h-refinement method. In this paper, a simple modification is proposed. For a given small percentage ϵ, the $\epsilon\%$ parametric cells are firstly marked with largest estimated error for refinement and then the mean-value marking approach is further applied to the remaining parametric cells.

4.3.3. Local refinement

In order to achieve accurate and efficient simulation results, local refinement should be performed on the marked parametric cells. In this part, the refinement rules of PHT-splines will be introduced.

Suppose that the T-mesh on the parametric domain at level k is denoted by $\mathbb{T}_k$, and the PHT-spline basis functions on $\mathbb{T}_k$ are constructed as M_j^k, $j = 1, ..., d_k$, then the basis functions on $\mathbb{T}_{k+1}$ can be constructed as follows: some basis functions are from the modifications of the former basis functions on $\mathbb{T}_k$, and others are from the new basis functions associated with the new basis vertices of $\mathbb{T}_{k+1}$. We represent a PHT-spline basis function by specifying 16 Bézier coefficients in each cell within the compact support of the basis function. When a cross vertex is added in a cell, the cell can be refined into four subcells. Each subcell supports the original basis function, and also has 16 Bézier coefficients. Five new vertices are obtained by adding a cross vertex, and some new basis vertices are also introduced. Hence, for the old basis functions, all the Bézier coefficients associated with the new basis vertices should be reset to zero. The new basis vertices are introduced from two sources, i.e., some are crossing vertices while others are T-vertices from the previous level. The later become basis vertices as the addition of cross points to the neighboring cells. The new basis functions can be further constructed over their supporting cells as in the initial level. For further details, the readers can refer to [17].

Note that in GIFT local refinement is only performed on the solution field, hence we only need to update the PHT-spline basis functions on the local-refined T-mesh for the new solution field. It is unnecessary to derive the updated control variables from the old control variables after local refinement, and the new control variables for the refined solution field can be obtained by re-solving the PDE in the new PHT-spline space over the refined T-mesh.

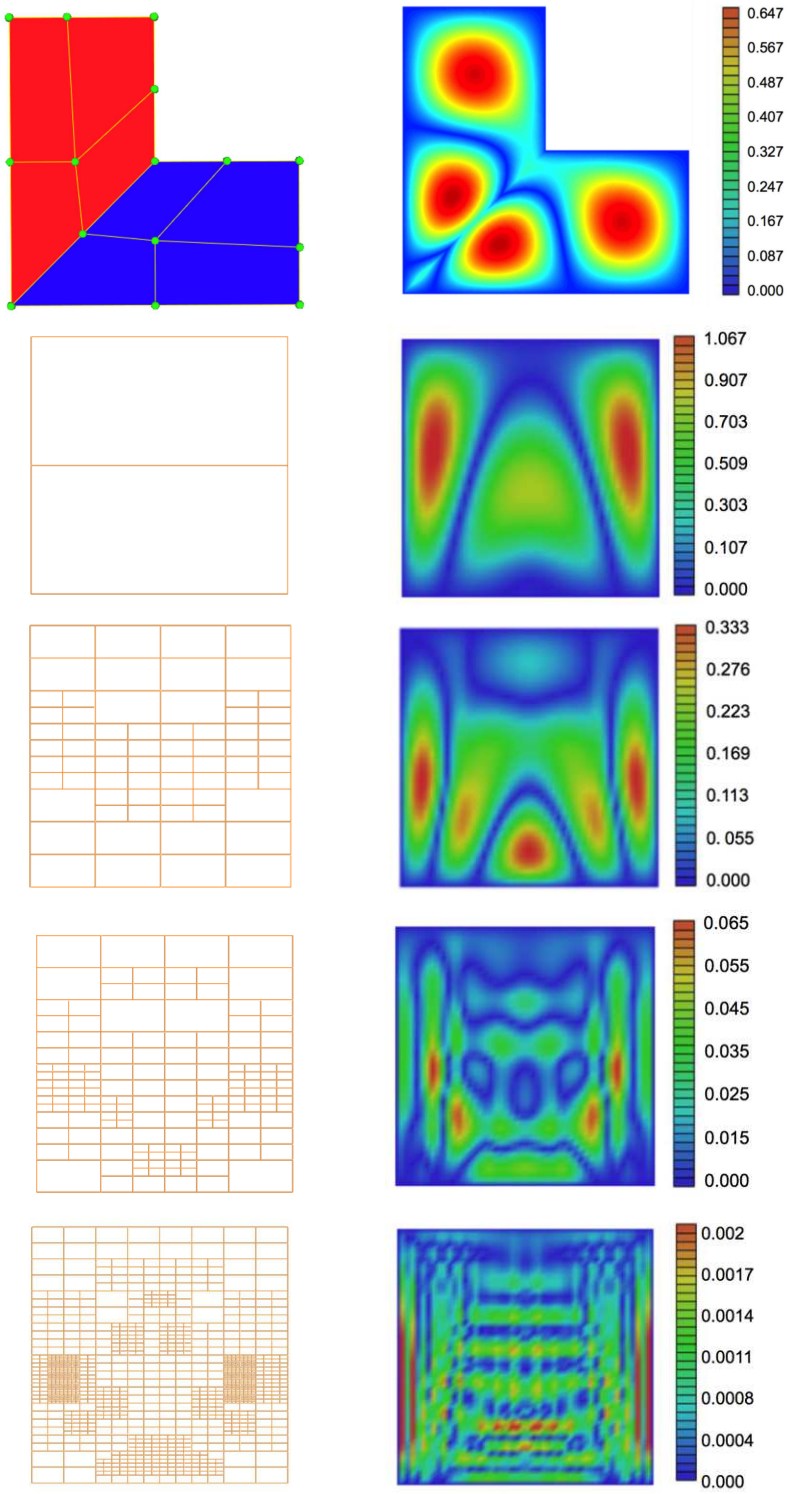

Fig. 8. L-shape example with local refinement. The first row shows the C^0 two-patch computational domain (left) and IGA error color-map (right) on the computational domain. From the second row to the fifth row, we show the T-mesh on parametric domain (left) of GIFT method using PHT-spline and the corresponding error color-map (right) on parametric domain during local refinement.

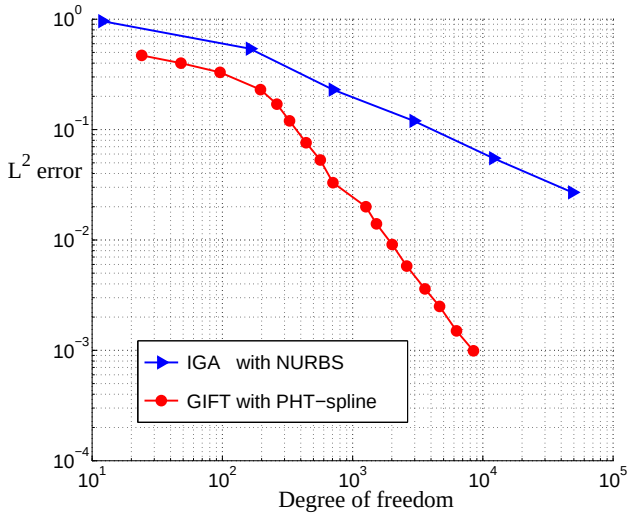

Fig. 9. Comparison of the convergence results with IGA method using NURBS and GIFT method using PHT-splines in Fig. 8.

4.4. Examples

In this subsection, we present two examples on heat conduction (17) with source function

$$f(\boldsymbol{x}) = 2\pi^2 \sin(\pi x) \sin(\pi y)$$

to illustrate the effectiveness of the proposed GIFT method.

The multi-patch L-shape example with exact solution

$$u(\boldsymbol{x}) = \sin(\pi x) \sin(\pi y)$$

over the computational domain is presented in Fig. 8. The first row shows the C^0 computational domain with two quadratic NURBS patches (left) and the IGA error color-map (right) on the computational domain. From the second row to the fifth row, we show the T-mesh on the parametric domain (left) of GIFT method using C^1 PHT-spline solution field and the corresponding exact error color-map (right) on the parametric domain with different scales during local refinement operations. Different from the IGA method, for this multi-patch example, GIFT method with cubic PHT-spline can achieve C^1 continuity automatically without special treatment on the C^1 constraint conditions. As shown in Fig. 9, it is seen that the convergence rate of GIFT with PHT-spline is higher than that of isogeometric analysis using NURBS.

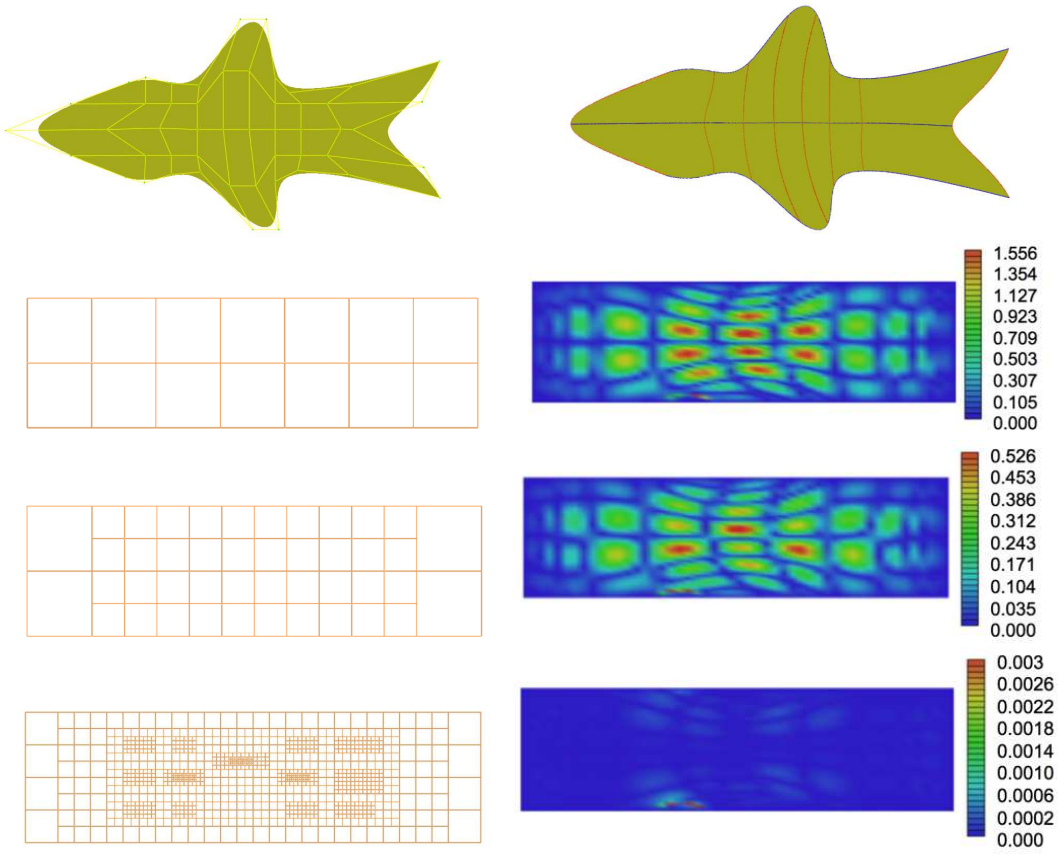

Fig. 10. GIFT for an example where no exact solution is available using local refinement with PHT-splines. The first row shows the computational domain with the control mesh (left) and knot patches (right). From the second row to the fourth row, we show the T-mesh in the parametric domain (left) and the corresponding color-map of error measurement (right) with different scales during local refinement operations.

The second example for which an exact solution is unavailable is shown in Fig. 10. The computational domain with control mesh (left) and knot patches (right) are shown in the first row. From the second row to the fourth row, the T-mesh in the parametric domain (left) and the corresponding color-map of error measurement (right) with different scales during local refinement operations are shown. As the exact solution is unavailable, we adapt the method proposed in [52] to compute the error measurement.

5. Conclusion

In this paper, a new field approximation method called GIFT in which the spline spaces of the solution field and the geometry can be chosen independently is proposed. In the proposed method,

for a given computational domain with NURBS form, the solution field can have a different spline representation, such as PHT-splines, T-splines and generalized B-splines. The main principle and features of the proposed method is described, and adaptive GIFT with PHT-spline solution is also proposed. Several examples are shown to illustrate the effectiveness of the proposed method.

This paper could be the basis for the following research directions:

- Detailed studies of the stability and convergence properties of the method for various types of boundary and field approximation.
- Coupling of spline approximations for the geometry with non-spline approximation for the field variables, including meshless methods. It would be particularly attractive to work with Maximum Entropy Shape functions [43, 44, 45].
- In this paper, we assumed the field approximation spline space to be fixed, albeit independently of the geometry approximation. It would be desirable for sensitivity analysis, e.g. through adjoint methods to drive the automatic selection of the most suitable field spline space based on goal-oriented error estimators, e.g. [41, 42].
- GIFT for isogeometric shape and topology optimisation [48], where the constraint of using the same space for the geometry and the approximation is particularly undesired as described in [23] and [24].
- Extension to other partial differential equations where the ability to locally enrich the field approximation independently of the geometry is particularly important (crack growth or other free boundary problems).
- Parallelization on GPUs is particularly attractive if the field and geometry approximations can be built on a hierarchical oct-tree structure. Then Jacobians can be computed at selected integration points, and pre-computing can be performed to accelerate the simulation.
- As implemented here, the proposed method based on a PHT-spline solution field is only suitable for "regular" domains, in which the control structure formed by the control meshes of each patch has no extraordinary inner vertices. The cases with complex computational domains, will be discussed in the forthcoming paper.
- GIFT within an isogeometric boundary element framework.

Acknowledgements

Stéphane Bordas thanks Prof. Gernot Beer, Mr. B. Marussig, Dr. J. Zechner and Dr. C. Duenser for constructive discussions in Graz, Cardiff, Udine and Vienna. We would like to thank also useful input from Dr. Pierre Kerfriden on computational efficiency of the Jacobian computations and Dr. Robert Simpson for sharing his expertise on IGA.

The work presented in this paper is partially supported (Gang Xu) by the National Nature Science Foundation of China (Nos. 61004117, 61272390, 61211130103), the Defense Industrial Technology Development Program (A3920110002), the Scientific Research Foundation for Returned Overseas Chinese Scholars from State Education Ministry, the Open Project Program of the State Key Lab of CAD&CG (Grant No. A1406), Zhejiang University.

Weiyin Ma thanks the partial support from Research Grants Council of Hong Kong SAR (Grant No. CityU 118512).

Stéphane P.A. Bordas thanks the financial support for his time from the European Research Council Starting Independent Research Grant (ERC Stg grant agreement No. 279578) entitled "Towards real time multiscale simulation of cutting in non-linear materials with applications to surgical simulation and computer guided surgery".

Bordas also thanks partial support from EPSRC Projects EP/I006494/1 "Sustainable domain-specific software generation tools for extremely parallel particle-based simulations" and EP/G042705/1 "Increased Reliability for Industrially Relevant Automatic Crack Growth Simulation with the eXtended Finite Element Method."

The research group in Cardiff and visits of Dr. Atroshchenko have been supported by the European Union Initial Training Network (ITN) INSIST from the Framework Programme 7 under grant number 289361 "Integrating Numerical Simulation and Geometric Design Technology."

The project has been supported by High Performance Computing (HPC) Wales, a company formed between the Universities and the private sector in Wales which provides the UKs largest distributed supercomputing network.

Appendix.I NUAT B-splines

In this part, we will review the definition of NUAT B-splines [49].

Let T be a given knot sequence $\{t_i\}_{i=-\infty}^{+\infty}$ with $\Delta t_i = t_{i+1} - t_i \in [0, \pi)$. Let $\Omega_k[T]$ denote the collection of all algebraic-trigonometric spline of order k in the span of $\{1, t, \cdots, t^{k-3}, \cos t, \sin t\}$, $k \geq 3$. The basis of $\Omega_k[T]$ is called a *non-uniform algebraic-trigonometric B-splines (NUAT B-spline) basis of order* k if the basis functions satisfy the properties of nonnegative, partition of unity and linear independence.

To construct a NUAT B-spline basis of $\Omega_k[T]$ when $k \geq 3$, we first define a set of functions over $\Omega_2[T]$

$$N_{i,2}(t) = \begin{cases} \sin(t - t_i)/\sin(t_{i+1} - t_i), & t_i \leq t \leq t_{i+1} \\ \sin(t_{i+2} - t)/\sin(t_{i+2} - t_{i+1}), & t_{i+1} \leq t \leq t_{i+2} \\ 0, & otherwise \end{cases} \tag{28}$$

For $k \geq 3$, $N_{i,k}(t)$ are defined recursively by

$$N_{i,k}(t) = \int_{-\infty}^{t} \left(\delta_{i,k-1} N_{i,k-1}(s) - \delta_{i+1,k-1} N_{i+1,k-1}(s)\right) ds, \qquad k \geq 3, i = 0, \pm 1, \ldots, \tag{29}$$

where $\delta_{i,k} := \left(\int_{-\infty}^{+\infty} N_{i,k}(t) dt \right)^{-1}$.

After constructing NUAT B-spline basis function, the NUAT B-spline surfaces can be defined as

$$\boldsymbol{S}(u, v) = \sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,k}(u) N_{j,h}(v) \boldsymbol{P}_{i,j}, \ u \in [t_k, t_{n+1}], v \in [t_h, t_{m+1}]; n \geq k, m \geq h.$$

where $N_{i,k}(u)$ and $N_{j,h}(v)$ are NUAT B-spline basis functions, $\boldsymbol{P}_{i,j}$ are control points.

References

[1] Y. Bazilevs, L. Beirao de Veiga, J.A. Cottrell, T.J.R. Hughes, and G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for refined meshes. Mathematical Models and Methods in Applied Sciences, 6(2006) 1031-1090.

[2] Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and Y. Zhang. Isogeometric fluid structure interaction: Theory, algorithms, and computations. Computational Mechanics, 43(2008) 3-37.

[3] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-Splines. Computer Methods in Applied Mechanics and Engineering, 199(2010) 229-263.

[4] G. Beer, B.Marussig, Ch. Duenser. Isogeometric boundary element method for the simulation of underground excavations. Géotechnique letters, 2013(3) 108-111.

[5] G. Beer, B. Marussig, J. Zechner, C. Duenser, TP Fries. Boundary Element Analysis with trimmed NURBS and a generalized IGA approach. 11th World Congress on Computational Mechanics (WCCM XI), 2014.

[6] L. Beirão da Veiga, A. Buffa. D. Cho, G. Sangalli. Analysis-suitable T-splines are dual-compatible. Computer methods in applied mechanics and engineering, 249(2012): 42-51.

[7] L. Beirão da Veiga, A. Buffa. D. Cho, G. Sangalli. IsoGeometric analysis using T-splines on two-patch geometries. Computer methods in applied mechanics and engineering, 200(2011): 1787-1803.

[8] L. Beirão da Veiga, A. Buffa, C. Lovadina, M. Martinelli, G. Sangalli. An isogeometric method for the Reissner–Mindlin plate bending problem. Computer Methods in Applied Mechanics and Engineering, 209(2012): 45-53

[9] D. Burkhart, B. Hamann and G. Umlauf. Iso-geometric analysis based on Catmull-Clark subdivision solids. Computer Graphics Forum, 29(2010) 1575-1584.

[10] L. Chen, N. Nguyen-Thanh, H. Nguyen-Xuan, T. Rabczuk, S. Bordas, G. Limbert. Explicit finite deformation analysis of isogeometric membranes. Computer Methods in Applied Mechanics and Engineering, 277(2014): 104-130.

[11] E. Cohen, T. Martin, R.M. Kirby, T. Lyche and R.F. Riesenfeld, Analysis-aware modeling: understanding quality considerations in modeling for isogeometric analysis. Computer Methods in Applied Mechanics and Engineering, 199(2010) 334-356.

[12] J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs. Isogeometric Analysis: Toward Integration of CAD and FEA. Wiley, 2009

[13] J.A. Cottrell, T.J.R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric analysis. Computer Methods in Applied Mechanics and Engineering, 196(2007) 4160-4183.

[14] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. Computer Methods in Applied Mechanics and Engineering, 195(2006) 5257-5296.

[15] F. Cirak, M. Ortiz, P. Schröder. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. International Journal for Numerical Methods in Engineering, 47 (12), 2039-2072

[16] N. Crouseilles, A. Ratnani, E. Sonnendrücker. An isogeometric analysis approach for the study of the gyrokinetic quasi-neutrality equation. Journal of Computational Physics, 231(2012) 373-393.

[17] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang et al. Polynomial splines over hierarchical T-meshes. Graphical Models, 74(2008) 76-86

[18] J.M. Escobara, J.M. Cascónb, E. Rodrígueza, R. Montenegro. A new approach to solid modeling with trivariate T-spline based on mesh optimization. Computer Methods in Applied Mechanics and Engineering, 200(2011) 3210-3222.

[19] H. Gomez, V.M. Calo, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of the Cahn-Hilliard phase-field model. Computer Methods in Applied Mechanics and Engineering, 197(2008) 4333-4352.

[20] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. Computer Methods in Applied Mechanics and Engineering, 194(2005) 4135-4195.

[21] Y. Jia, Y. Zhang, G. Xu, X. Zhuang, T. Rabczuk. Reproducing kernel triangular B-spline-based FEM for solving

PDEs. Computer Methods in Applied Mechanics and Engineering, 267(2013) 342-358.

[22] K. Johannessen, T. Kvamsdal, T. Dokken. Isogeometric analysis using LR B-splines. Computer Methods in Applied Mechanics and Engineering. 269(2014) 471-514.

[23] H. Lian, P. Kerfriden and S.P.A. Bordas. Shape sensitivity analysis and optimization using isogeomgetric boundary element methods in two-dimensional linear elasticity. http://publications.uni.lu/handle/10993/16043 (2014).

[24] H. Lian, P. Kerfriden, and S.P.A. Bordas. Sensitivity Analysis and Shape Optimisation through a T-spline Isogeometric Boundary Element Method. http://publications.uni.lu/handle/10993/13847 (2013).

[25] H. Lian, S. P. A. Bordas, R. Sevilla, R. N. Simpson. Recent developments in CAD/analysis integration. arXiv:1210.8216, 2012

[26] C. Manni, F. Pelosi, M. Sampoli. Generalized B-splines as a tool in isogeometric analysis. Computer Methods in Applied Mechanics and Engineering, 200(2011), 867–811

[27] B. Marussig, J. Zechner, G. Beer, C. Duenser and T. P. Fries. Fast isogeometric boundary element method based on super-parametric representation. Proceeding of IGA 2014, Austin.

[28] N. Moës, M. Cloirec, P.Cartraud, J.F. Remacle. A computational approach to handle complex microstructure geometries. Computer Methods in Applied Mechanics and Engineering, 192(2003), 3163–3177

[29] M. Moumnassi, S. Belouettar, E. Béchet, S.P.A. Bordas, D. Quoirin, M.Potier-Ferry. Finite element analysis on implicitly defined domains: An accurate representation based on arbitrary parametric surfaces. Computer Methods in Applied Mechanics and Engineering, 200(2011), 774–796

[30] N. Nguyen-Thanh, H. Nguyen-Xuan, S.P.A. Bordasd and T. Rabczuk. Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids. Computer Methods in Applied Mechanics and Engineering, 200(2011) 1892-1908.

[31] N. Nguyen-Thanh, J. Kiendl, H. Nguyen-Xuan, R. Wüchner, K.U. Bletzinger, Y. Bazilevs, T. Rabczuk. Rotation free isogeometric thin shell analysis using PHT-splines. Computer Methods in Applied Mechanics and Engineering, 200(2011) 3410-3424.

[32] K.F. Pettersen, V. Skytt. Spline volume fairing. 7th International Conference on Curves and Surfaces. Avignon, France, June 24-30, 2010, Lecture Notes in Computer Science, 6920(2012) 553-561.

[33] A. Ratnani, E. Sonnendrücker. An arbitrary high-order spline finite element solver for the time domain maxwell equations. Journal of Scientific Computing, 51(2012) 87-106.

[34] MA. Scott, RN. Simpson, JA. Evans, S. Lipton, SPA. Bordas, TJR Hughes, TW Sederberg. Isogeometric boundary element analysis using unstructured T-splines. Computer Methods in Applied Mechanics and Engineering, 254(2013): 197-221.

[35] RN. Simpson, SPA. Bordas, H. Lian, J. Trevelyan. An isogeometric boundary element method for elastostatic analysis: 2D implementation aspects. Computers & Structures, 118(2013), 2-12.

[36] RN. Simpson, MA. Scott, M. Taus, DC. Thomas, H. Lian. Acoustic isogeometric boundary element analysis. Computer Methods in Applied Mechanics and Engineering, 269(2014), 265–290.

[37] RN. Simpson, SPA. Bordas, J. Trevelyan, T. Rabczuk. A two-dimensional isogeometric boundary element method for elastostatic analysis. Computer Methods in Applied Mechanics and Engineering, 209(2012): 87-100.

[38] H. Speleers, C. Manni, F.Pelosi, M.L. Sampoli. Isogeometric analysis with Powell-Sabin splines for advection-diffusion-reaction problems. Computer Methods in Applied Mechanics and Engineering 221-222(2012) 132-148

[39] H. Speleers ,C. Manni , F. Pelosi. From NURBS to NURPS geometries. Computer Methods in Applied Mechanics and Engineering, 255(2013) 238-254

[40] N. Valizadeh, S. Natarajan, O.A. Gonzalez-Estrada, T. Rabczuk, T.Q. Bui, S.P. Bordas. NURBS-based finite element analysis of functionally graded plates: Static bending, vibration, buckling and flutter, Composite Structures , 99 (2013) 309-326

[41] Stein, E., Rüter, M. Finite Element Methods for Elasticity with Errorcontrolled Discretization and Model Adaptivity. Encyclopedia of Computational Mechanics. (2007)

[42] González-Estrada, O. A., E. Nadal, J. J. Ródenas, P. Kerfriden, S. P. A. Bordas, and F. J. Fuenmayor. Mesh adaptivity driven by goal-oriented locally equilibrated superconvergent patch recovery. Computational Mechanics (2013): 1-20. http://publications.uni.lu/bitstream/10993/12025/1/GOAFEM.pdf http://orbilu.uni.lu/handle/10993/12025

[43] Sukumar, N. Construction of polygonal interpolants: a maximum entropy approach. International Journal for Numerical Methods in Engineering 61.12 (2004): 2159-2181.

[44] Arroyo, M, and Ortiz, M. Local maximumentropy approximation schemes: a seamless bridge between finite elements and meshfree methods. International Journal for Numerical Methods in Engineering 65.13 (2006): 2167-2202.

[45] Amiri, F., C. Anitescu, M. Arroyo, S. P. A. Bordas, and T. Rabczuk. XLME interpolants, a seamless bridge between XFEM and enriched meshless methods. Computational Mechanics 53, no. 1 (2014): 45-57.

[46] R. Vázquez, A. Buffa. Isogeometric analysis for electromagnetic problems. IEEE Transactions on Magnetics, 46(2010), 3305-3308.

[47] W. Wang, Y. Zhang, L. Liu, T. J.R. Hughes. Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. Computer-Aided Design, 45(2013) 351-360.

[48] W. A. Wall, M. A. Frenzel, and C. Cyron. Isogeometric structural shape optimization. Computer Methods in Applied Mechanics and Engineering 197, no. 33 (2008): 2976-2988.

[49] G. Wang, Q. Chen, M. Zhou. NUAT B-spline curves. Computer Aided Geometric Design, 21(2004) 193-205.

[50] G. Xu, E. Atroshchenko, S. P. A. Bordas. Geometry-independent field approximation for spline-based finite element methods. 11th World Congress on Computational Mechanics (WCCM XI), 2014.

[51] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo. Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis. Computer-Aided Design, 45(2013) 812-821.

[52] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo. Parameterization of computational domain in isogeometric analysis: methods and comparison. Computer Methods in Applied Mechanics and Engineering, 200(2011) 2021-2031.

[53] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo. Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications. Computer-Aided Design, 45(2013) 395-404.

[54] Y. Zhang, W. Wang, T. J.R. Hughes. Conformal solid T-spline construction from boundary T-spline representations. Computational Mechanics, 51(2013) 1051-1059.