Adaptive *w*-refinement: a new paradigm in isogeometric analysis

Alireza H. Taheri[1], Krishnan Suresh[1*]

[1] Department of Mechanical Engineering, UW-Madison, Madison, Wisconsin 53706, USA.

[*] Corresponding author, email: ksuresh@wisc.edu; 608-262-3594

Abstract:

Motivated by the concept of generalized NURBS (GNURBS), recently introduced by the authors, we devise a novel adaptivity technique in isogeometric analysis (IGA), referred to as adaptive *w*-refinement. GNURBS-based IGA is a natural extension of IGA where the weights of the basis functions in geometry and solution space are decoupled. Considering the additional unknown *control weights* in the solution function space as design variables, we develop an adaptive algorithm to find these unknowns by solving an unconstrained optimization problem. Due to the decoupling of the weights, the analytical sensitivities can be derived cost effectively; consequently, the optimization problem can be solved efficiently by a gradient-based algorithm. This procedure leads to the optimal rational function space associated with the problem under study, while preserving the underlying geometry as well as its parameterization.

We study the performance of this algorithm on elliptic problems with both smooth and rough solutions. Numerical results demonstrate significant improvement of accuracy as well as the convergence rate compared to classic NURBS-based IGA. Moreover, the proposed method enables the isogeometric method to solve problems, whose closed-form solutions lie in rational space, exactly, revealing a new crucial aspect of employing rational splines for analysis. The proposed adaptive *w*-refinement technique serves as a new powerful adaptive technique in IGA, and perhaps a competitive tool with hierarchical splines for alleviating the deficiencies of NURBS for analysis.

Keywords:

Adaptivity, GNURBS, control weights, optimal rational space.

1. Introduction:

Isogeometric analysis (IGA) was introduced by Hughes et al. [1] as an innovative numerical methodology for the solution of boundary value problems. In contrast to classic Finite Element Method (FEM), IGA is more tightly integrated with the geometry, and circumvents the requirement for a conventional mesh generation process, via direct communication with CAD models. Moreover, it has been shown that, when deployed for analysis, higher-order smooth spline bases commonly used in CAGD yield superior results in terms of accuracy and robustness compared to standard C^0 discretizations. This has been demonstrated in a variety of application areas such as structural, fluid, etc. [2].

While offering many well-known advantages over classic FEM, the ultimate success of the method in integrating design and analysis is mainly contingent upon the development of modern spline technologies which sufficiently meet the demands of both analysis and design. Extensive research has been conducted towards this, and immense progress has been made in various aspects. A major difficulty which has attracted significant attention is the inability of piecewise smooth tensor product splines in solving problems with irregularities such as sharp layers or singularities. As will be discussed further below, the same concept of providing the possibility of local *h*-refinement in splines has been commonly pursued for alleviating this issue by the community so far.

In this paper, after providing an overview of these studies, we will explore an alternative powerful technology that isogeometric analysis exclusively provides for addressing these problems. The proposed method provides the possibility of enrichment of function space without introducing additional basis functions. This novel adaptivity technique, which will be referred to as *adaptive w-refinement*, is established based on a new variation of NURBS, named Generalized NURBS (GNURBS), recently developed by the authors [3] for parametric curves. The extension of this variation for surfaces which will form the basis for *w*-refinement will be discussed in Section 2.

1.1. Refinement techniques in isogeometric analysis

One of the most interesting aspects of using splines as the basis for analysis is the possibility of exploiting multiple elegant and efficient techniques that they provide for the enrichment of function space. Having an initial parameterization of computational model, a variety of refinement techniques can be used to improve the accuracy of approximation in IGA. One may classify these techniques into two categories, namely, *function space refinement*, and *control net refinement* techniques described next.

Function space refinement techniques attempt to enrich the function space while preserving the underlying geometry and its parameterization unchanged. The existing techniques in this class are *h*, *p*, and *k* refinements. The details of these algorithms can be found in [2]. While *h* and *p* refinements are common to both classic FEM and IGA, *k*-refinement is exclusive to IGA. This additional possibility of refinement is one of the key advantages of IGA over classic FEM as it provides higher order continuity by performing degree elevation followed by knot insertion in a special manner. Finally, we note here that these algorithms may be employed in combination for improved performance, usually in an adaptive manner, as in *hp*-adaptive refinement [4], etc.

Despite providing the above-mentioned effective refinement techniques, in contrast to the standard nodal basis commonly used in FEA, a multivariate tensor-product spline basis, such as NURBS, does not provide a natural possibility for local mesh refinement. In fact, this was soon known as a fundamental limitation of IGA since the possibility of adaptive local mesh refinement is critical in FEA, and is commonly used to resolve local features such as internal and boundary layers in advection dominated flows and stress concentrations in structures [5].

Eliminating this limitation by modifying the existing spline technologies or developing new variations of splines has perhaps been the most active area of research in IGA community over the last decade; see, for example, [5–12]. We do not intend to review these studies here; instead we refer to [13,14] for a comprehensive review. We suffice to mention here that the primary purpose of these studies is to bring the possibility of local *h*-refinement in splines by allowing for some unstructuredness. Various forms of T-splines [8,15–27], subdivision basis functions coupled with the truncation mechanism [28–30], LR-splines [31], (truncated) hierarchical B-splines [6,32,33], (R)PHT-splines [11,34], and most recently U-splines [35] are some of the popular technologies in this category. This concept has shown promising results for resolving local features and achieving an improved rate of convergence in problems with poor regularities that contain steep layers or singularities [5]. These techniques have also been incorporated with commercial FEM software such as Abaqus and LS-DYNA to support real engineering IGA applications [36].

In one of these studies, which is of particular interest to the current research, Atroshchenko et al. [37] suggest a generalization of IGA by weakening the tight coupling between geometry and solution space. This concept, which is referred to as Geometry-Independent Field approximaTion (GIFT) by its authors, allows for different spaces for the parameterization of the computational domain and approximation of the solution field. They argue that this method inherits the main advantage of IGA by preserving the original exact CAD representation of the geometry, such as NURBS, but allows for pairing it with an approximation space, such as T-splines, LR-splines, (truncated) hierarchical B-splines, or PHT-splines, which is more suitable/flexible for analysis [37]. In particular, it offers the advantage of adaptive local refinement without the need to reparametrize the domain, and therefore without losing the link with the CAD model. They study the performance of this method with different choices of geometry and field spaces and demonstrate that, despite the failure of the standard patch test, the optimum convergence rate is achieved for non-nested spaces [37].

In contrast to function space refinement, the second class of *control net refinement* adaptively modifies an initial parameterization to improve the accuracy of approximation *without* enrichment of the function space while preserving the boundaries of the geometric domain. Strategies in this category are usually referred to as *r-refinement* [38,39]. This refinement is usually posed as an optimization problem which aims at minimization of an estimation of the error, obtained using a posteriori error estimation technique, by adaptively repositioning the *interior* control points; see e.g. [40]. For a review of these studies please see [38]. There are however many deficiencies in this method discussed below which makes it impractical, especially for large scale problems.

For instance, the above-mentioned procedure leads to solving a heavily constrained non-linear optimization problem since the parameterization must remain valid (bijective) throughout the optimization process. This involves imposing a large number of constraints to ensure the positivity of Jacobian over the whole domain [40]. Another major shortcoming of these methods is that they are only applicable to problems which have an *interior* region and do not apply for problems with an arbitrary geometry such as free-form shells or curved beams. Further, derivation of analytical sensitivities does not seem possible; hence, the existing studies, e.g. [40], rely on finite difference method which makes the algorithm prohibitively expensive. Finally, to the best of our knowledge, none of the existing studies in this class report a substantial improvement in the accuracy or rate of convergence.

1.2. Applications of rational splines in CAGD and IGA

Since the proposed adaptivity technique in IGA is essentially based on rational splines and exploring interesting opportunities that they provide, we first review the current applications of these rational forms in the context of IGA as well as their broader scope in CAGD.

To quote [3]: "*Historically, NURBS were primarily introduced to represent conical shapes precisely. This is the critical advantage of NURBS over other polynomial-based classes of splines, and the main reason for its prevalence*". However, other applications of this rational form can be found in CAGD. A thorough study of these applications has been recently reported in [3].

One of these applications which is of particular interest for the current study, is employing the weights as additional design variables in data-fitting for obtaining better accuracy [41,42]. For instance, Carlson [42] develops a non-linear least square fitting algorithm based on NURBS, and discusses multiple methods for solving this problem. His numerical results demonstrate significant improvement in the accuracy of approximation compared to B-splines, especially in the case of rapidly varying data. This, in fact, seems to be one of the other main advantages of NURBS over B-splines. While smooth piecewise polynomials such as B-splines are inherently poor for the approximation of rapidly varying data and discontinuities, employing rational functions seems to be an effective tool for addressing this shortcoming [42]. To avoid solving a non-linear least-square problem, Ma [41,43] develops a two-step linear algorithm for data approximation using NURBS.

Despite being an effective technique for improving the performance of NURBS in a variety of applications in CAGD, considering the literature of IGA, the application of NURBS as well as other rational splines surprisingly seems to be merely limited to precise representation of conical shapes in this area. In this paper, by taking inspiration from the concept of GIFT [37], we study a generalization of NURBS which allows for treating the weights as additional degrees of freedom for improved flexibility in a wider range of applications in IGA. This generalization significantly improves the performance of NURBS and provides an alternative powerful tool for removing its deficiencies for analysis. It will be shown that, unlike T-splines, GNURBS are only disguised form of classic NURBS, and do not constitute a new superset of NURBS, making it easy to integrate and deploy them in existing IGA packages.

2. Generalized Non-Uniform Rational B-Spline (GNURBS) surfaces

The concept of Generalized Non-Uniform Rational B-Splines (GNURBS) has recently been introduced and investigated for parametric curves in [3] by the authors. We provide an extension of this concept for parametric surfaces here which will establish the foundation for the proposed adaptivity technique in IGA.

2.1. Definition

We recall that the equation of a NURBS surface is defined in the following parametric form

$$\mathbf{S}(\xi,\eta)=\sum_{i=0}^{n_1}\sum_{j=0}^{n_2}R_{ij}^{p,q}(\xi,\eta)\mathbf{P}_{ij}\qquad \begin{array}{l}0\le\xi\le1\\0\le\eta\le1\end{array}\tag{1}$$

where $\mathbf{P}_{ij}=[x_{ij},y_{ij},z_{ij}]^T$ is a set of $(n_1+1)\times(n_2+1)$ control points and $R_{ij}^{p,q}(\xi,\eta)$ are the corresponding rational basis functions associated with $(i,j)^{\text{th}}$ control point defined as

$$R_{ij}^{p,q}(\xi,\eta)=\frac{N_{ij}^{p,q}(\xi,\eta)w_{ij}}{\sum_{k=0}^{n_1}\sum_{l=0}^{n_2}N_{kl}^{p,q}(\xi,\eta)w_{kl}}\tag{2}$$

where w_{ij} are the weights associated with control points, and $N_{ij}^{p,q}(\xi,\eta)=N_{i,p}(\xi)N_{j,q}(\eta)$ are bivariate B-spline basis functions. $N_{i,p}(\xi)$ and $N_{j,q}(\eta)$ are the univariate B-spline basis functions of degree p and q defined on sets of non-decreasing real numbers $\boldsymbol{\Xi}=\{\xi_0,\xi_1,...,\xi_{n+p}\}$ and $\mathbf{H}=\{\eta_0,\eta_1,...,\eta_{n+p}\}$, respectively, called knot vectors.

According to Eq. (1), NURBS surfaces are *isoparametric* representations where all the physical coordinates are constructed by linear combination of the same set of scalar basis functions in parametric space. This is the case for all the other popular CAGD representations such as different types of splines; and ensures critical properties such as affine invariance and convex hull which are of interest in geometric modelling [3].

We extend here the concept of Generalized Non-Uniform Rational B-Splines (GNURBS) [3] to surfaces by modifying Eq. (1) as follows

$$\mathbf{S}(\xi,\eta)=\sum_{i=0}^{n_1}\sum_{j=0}^{n_2}\mathbf{R}_{ij}^{p,q}(\xi,\eta)\odot\mathbf{P}_{ij}\qquad \begin{array}{l}0\le\xi\le1\\0\le\eta\le1\end{array}\tag{3}$$

where $\odot$ denotes Hadamard (entry-wise) product of two vector variables and $\mathbf{R}_{ij}(\xi,\eta)=[R_{ij}^{x}(\xi,\eta),\, R_{ij}^{y}(\xi,\eta),\, R_{ij}^{z}(\xi,\eta)]^{T}$ is now a vector set of basis functions. Note that superscripts p,q have been omitted for brevity. Denoting an arbitrary coordinate in physical space by $d\in\{x,y,z\}$, the corresponding basis function in direction *d* can be written as

$$R_{ij}^{d}(\xi,\eta)=\frac{N_{ij}^{p,q}(\xi,\eta)\,w_{ij}^{d}}{\sum_{k=0}^{n_1}\sum_{l=0}^{n_2}N_{kl}^{p,q}(\xi,\eta)\,w_{kl}^{d}}. \tag{4}$$

In above equations, $\left(w_{ij}^{x},w_{ij}^{y},w_{ij}^{z}\right)$ represents the set of coordinate-dependent weights associated with $(i,j)^{\text{th}}$ control point.

Comparison of the above equation with that of classic NURBS in Eq. (1) shows that the main difference of the proposed generalized form is assigning independent weights to different physical coordinates of control points. As can be seen, the above leads to a *non-isoparametric* representation. This representation demonstrates different geometric properties compared to NURBS. For instance, it can be shown that a GNURBS surface need not satisfy properties such as strong convex hull and affine invariance. In fact, most of the theoretical properties, such as Axis-Aligned Bounding Box (AABB), which have been rigorously discussed for GNURBS curves in [3], can be easily extended for GNURBS surfaces. While these properties can play an important role in certain applications such as isogeometric structural optimization, many of them are not of significance in analysis, which is the focus of this paper. Therefore, we do not discuss all these properties here. Instead, we only elaborate two key properties of GNURBS surfaces, namely their local modification effect and their equivalence with higher order classic NURBS, which inspire their necessity for *w*-adaptivity. Before discussing these properties, we first construct a simplified variation of GNURBS surfaces which will be the emphasis for the rest of this paper.

2.2. Partial decoupling of the weights for 3D surfaces

A more practical variation of GNURBS which will later form the foundation for *w*-adaptivity in IGA, is obtained by partial decoupling of the weights. In particular, for 3D surfaces, one can use the same set of in-plane weights along *x* and *y* directions, denoted by w^{xy}, and a different set of out-of-plane weights in *z* direction w^{z}. Accordingly, Eq. (3) could be re-written in the following expanded form

$$\begin{Bmatrix} x(\xi,\eta) \\ y(\xi,\eta) \\ z(\xi,\eta) \end{Bmatrix} = \sum_{i=0}^{n_1}\sum_{j=0}^{n_2} \begin{Bmatrix} R_{ij}^{xy}(\xi,\eta)\, x_{ij} \\ R_{ij}^{xy}(\xi,\eta)\, y_{ij} \\ R_{ij}^{z}(\xi,\eta)\; z_{ij} \end{Bmatrix} \quad \begin{matrix} 0 \le \xi \le 1 \\ 0 \le \eta \le 1 \end{matrix} \tag{5}$$

where

$$R_{ij}^{xy}(\xi,\eta) = \frac{N_{ij}^{p,q}(\xi,\eta)\, w_{ij}^{xy}}{\sum_{k=0}^{n_1}\sum_{l=0}^{n_2} N_{kl}^{p,q}(\xi,\eta)\, w_{kl}^{xy}} \tag{6}$$

Observe that owing to this decoupling of the in-plane and out-of-plane weights, unlike in classic NURBS, one can now freely manipulate the weights along *z* direction, for instance, without perturbing the geometry or parameterization of the underlying surface in *x-y* plane. We discuss this property in more detail in the following section.

2.3. Local modification effect

Similar to NURBS, one can show that, in GNURBS, if a control point $\mathbf{P}_{ij}$ is moved, or if any of the weights $w_{ij}^{d}(d = xy, z)$ is changed, it affects the surface shape only over the rectangle $[\xi_i, \xi_{i+p+1}) \times [\eta_j, \eta_{j+q+1})$. However, unlike NURBS, changing the weights will only affect the parameterization of the surface along the corresponding physical coordinate(s) d, while the surface parameterization in the other direction(s) will be preserved. This is, in fact, the key difference between GNURBS and NURBS which inspires their application in adaptive *w*-refinement. In particular, assuming $(\xi,\eta) \in [\xi_i, \xi_{i+p+1}) \times [\eta_j, \eta_{j+q+1})$, if w_i^d is increased (decreased), the surface will move closer to (farther from) $\mathbf{P}_{ij}$. Further, for a fixed (ξ,η), a point on $\mathbf{S}(\xi,\eta)$ moves along a straight line along *d* towards $\mathbf{P}_{ij}$ as a weight w_{ij}^{d} is modified. This can be directly concluded from Eq. (5) and the properties of classic NURBS.

For better insight, we provide here a graphical representation of how this property differs in GNURBS compared to NURBS. For this purpose, we first generate a B-spline surface with linear in-plane parameterization using a net of 7×7 control points and quadratic basis functions in both parametric directions constructed over the knot vectors $\mathbf{\Xi} = \mathbf{H} = \{0,0,0,0.2,0.4,0.6,0.8,1,1,1\}$. The employed net of control points is illustrated in Fig. 1. As the figure shows, the heights of all control points are set to zero except for z_{44} which is raised to 1.

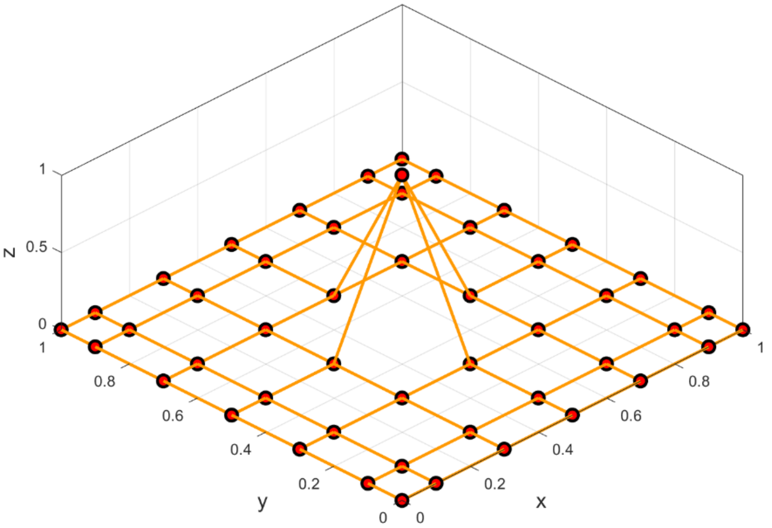


Fig. 1. Employed control net for construction of different NURBS surfaces.

The B-spline surface obtained by using this control net is depicted in Fig. 2.

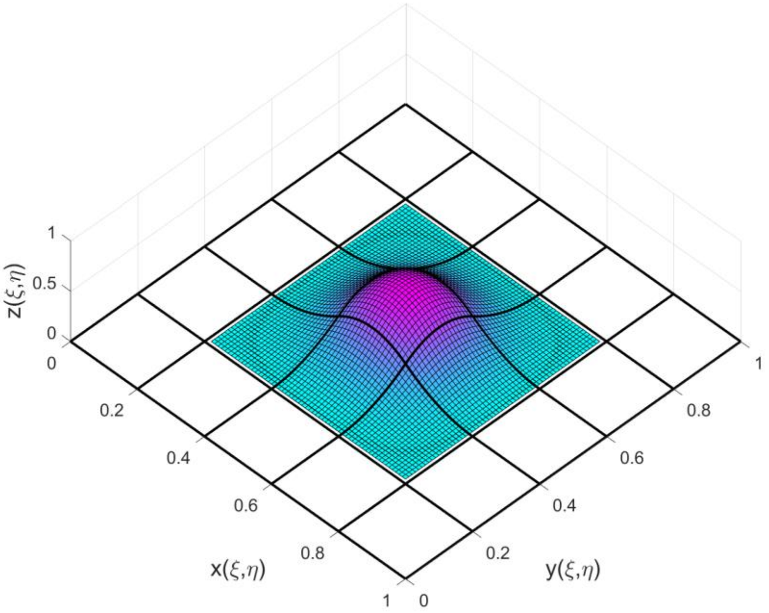


Fig. 2. The B-spline surface in physical space.

Next, we increase w_{44} to 4 and plot the resulting NURBS surface in the physical space in Fig. 3.

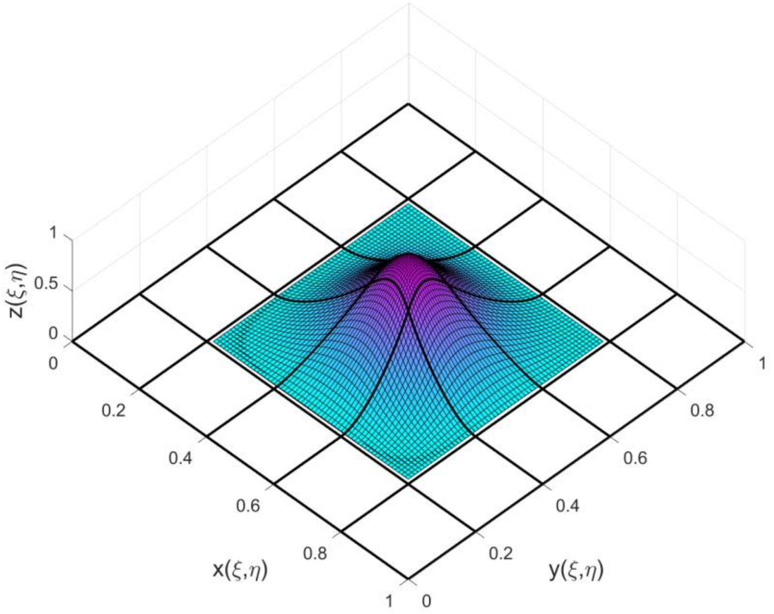


Fig. 3 The NURBS surface with $w_{44} = 4$ in physical space.

Finally, using Eq. 3, we construct a GNURBS surface by only setting w_{44}^{z} to 4, and maintaining all other weights at 1. The resulting surface is shown in Fig. 4.

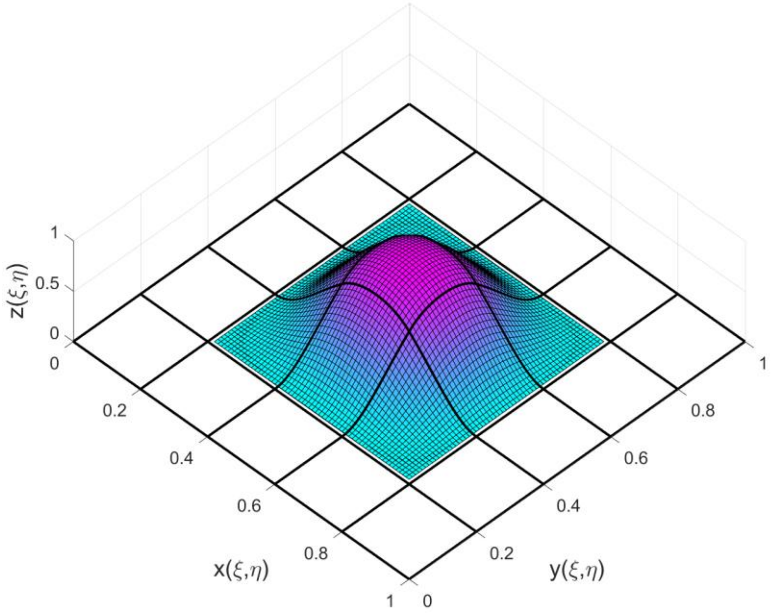


Fig. 4. The GNURBS surface with $w_{44}^{z} = 4$ in physical space.

Note that the depicted GNURBS surface in Fig. 4 is obtained by using two different sets of basis functions. The in-plane coordinates are obtained using the B-spline basis functions, while the out of plane coordinate is constructed using rational basis functions.

Comparing Figs. 3 and 4, one can clearly observe that modifying a weight in classic NURBS alters the parameterization of the surface in all physical directions, while in the case of GNURBS, the parameterization of the surface only changes in the direction of the varied directional weight (*z*-direction in Fig. 4). It will be seen later that this property is critical for the development of *w*-adaptivity in IGA and brings numerous geometrical and computational advantages.

2.4. Equivalence with NURBS

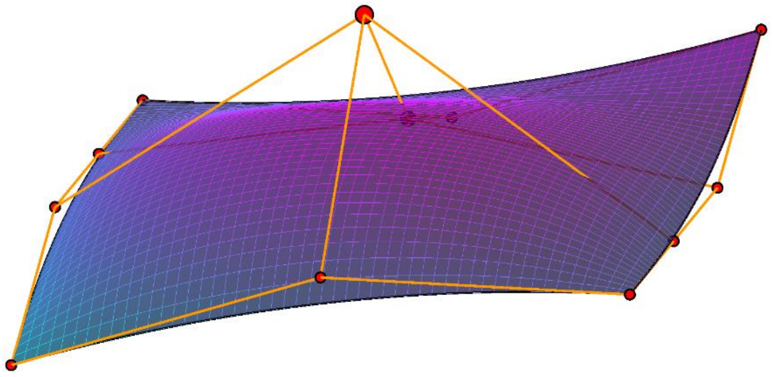
Despite losing some properties of NURBS which might be of interest in certain applications, we state here a theorem which establishes that GNURBS are nothing but disguised forms of higher-order classic NURBS. Therefore, all the properties of NURBS can be recovered through a suitable transformation and a strong theoretical foundation will be ensured.

Theorem 1. A 3D GNURBS surface of degree (p,q) with partially decoupled set of weights (w^{xy}, w^{z}), can be exactly transformed into a higher order NURBS surface of degree $(2p,2q)$.

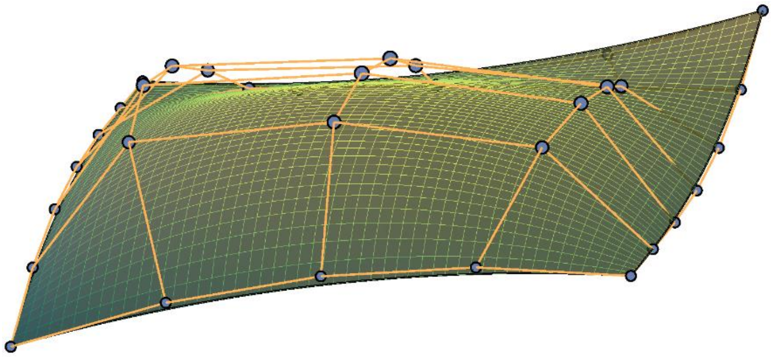
The proof of this theorem is provided in Appendix A. Fig. 5(a) shows an example of a degree (2,3)th generalized R-Bézier surface with the directional weights given by:

$$\mathbf{w}^{xy} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \ \mathbf{w}^{z} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{7}$$

Its equivalent higher order R-Bézier surface obtained using the above theorem is depicted in Fig. 5(b). Note that the size of control points in Fig. 5 are plotted proportional to their weights for better insight.



(a)



(b)

Fig. 5 (a) A degree (2,3)th GR-Bézier surface with directional weights in Eq. (7), and (b) its equivalent (isoparametric) R-Bézier surface of degree (4,6).

3. Adaptive *w*-refinement in isogeometric analysis

3.1. GNURBS-based IGA

We now consider the application of GNURBS in a proposed adaptive method for improved solution of boundary value problems. While the proposed adaptivity technique is potentially applicable to arbitrary boundary value problems, in this paper, we limit our study to single variable elliptic problems. In particular, we focus on steady reactive-diffusive transport problem over a 2D domain $\Omega \in \mathbb{R}^2$ with the boundary $\Gamma = \Gamma_D \bigcup \Gamma_N$, where Γ_D and $\Gamma_N = \Gamma \setminus \Gamma_D$ are the partitions of boundary where Dirichlet and Neumann boundary conditions are specified, respectively. We recall that the strong form of this PDE can be expressed as

$$\begin{aligned} &\mathcal{L}(u) \equiv -\nabla.(D\nabla u) + \sigma u = f \quad \text{in } \Omega, \\ &u = u_D \qquad \text{on } \Gamma_D, \\ &\mathbf{n}.D\nabla u = h \quad \text{on } \Gamma_N \end{aligned} \tag{8}$$

where $\mathcal{L}$ denotes the reaction-diffusion operator, D is the diffusion coefficient, σ is the reaction coefficient and f denotes the source term. Also, u_D and h are the specified Dirichlet and Neumann (normal diffusive flux) boundary conditions, and $\mathbf{n}$ denotes the unit outward normal along Γ.

We recall that in 2D NURBS-based IGA, the geometry is constructed by a planar NURBS surface

$$\mathbf{x}(\xi,\eta) = \sum_{i=0}^{n_1}\sum_{j=0}^{n_2} R_{ij}(\xi,\eta)\mathbf{x}_{ij} = \mathbf{RP}, \qquad \begin{array}{l} 0 \le \xi \le 1 \\ 0 \le \eta \le 1 \end{array} \tag{9}$$

where $\mathbf{x}=\left[x,y\right]^T$. Following the conventional *isoparametric* concept, the unknown field variable of the PDE is also approximated by using the same set of NURBS basis functions:

$$u\left(\xi,\eta\right)\approx u^h\left(\xi,\eta\right)=\sum_{i=0}^{n_1}\sum_{j=0}^{n_2}R_{ij}\left(\xi,\eta\right)u_{ij}=\mathbf{Ru} \tag{10}$$

where $\mathbf{u}$ is the unknown vector of degrees of freedom, whose components are referred to as *control variables*, and $\mathbf{R}$ denotes the vector of basis functions. This *isoparametric* concept is commonly used in Finite Element Analysis and provides well-known benefits in certain applications which are discussed in [2]. For instance, it ensures the ability to represent all affine motions (i.e., rigid translations and rotations, uniform stretchings and shearings) exactly [2].

However, by invoking the proposed generalization of NURBS in Section 2 as well as the concept of GIFT [37], we introduce a natural extension of isogeometric analysis where the field variable is approximated using a set of NURBS basis functions with independent weights of the underlying geometry. Accordingly, Eqs. (9) and (10) can be written as (11) and (12)

$$\mathbf{x}\left(\xi,\eta\right)=\sum_{i=0}^{n_1}\sum_{j=0}^{n_2}R_{ij}^G\left(\xi,\eta\right)\mathbf{x}_{ij}=\mathbf{R}^G\mathbf{P} \tag{11}$$

$$u^h\left(\xi,\eta\right)=\sum_{i=0}^{n_1}\sum_{j=0}^{n_2}R_{ij}^u\left(\xi,\eta\right)u_{ij}=\mathbf{R}^u\mathbf{u} \tag{12}$$

where

$$R_{ij}^G\left(\xi,\eta\right)=\frac{N_{ij}^{p,q}(\xi,\eta)w_{ij}^G}{\sum_{k=0}^{n_1}\sum_{l=0}^{n_2}N_{kl}^{p,q}(\xi,\eta)w_{kl}^G} \tag{13}$$

$$R_{ij}^u\left(\xi,\eta\right)=\frac{N_{ij}^{p,q}(\xi,\eta)w_{ij}^u}{\sum_{k=0}^{n_1}\sum_{l=0}^{n_2}N_{kl}^{p,q}(\xi,\eta)w_{kl}^u} \tag{14}$$

Observe that this decoupling of the weights in geometry and field variable space brings the critical possibility of treating the *control weights* w_{ij}^u as additional degrees of freedom without perturbing the exact underlying geometry or its parameterization. One can simply imagine that Eq. (11) together with Eq. (12) represent a GNURBS surface similar to Eq. (5), where the in-plane coordinates represent the exact planar geometry, while the out of plane coordinate can be viewed as the field primary variable, i.e. $w^G := w^{xy}, w^z := w^u$.

It is noted here that the above non-isoparametric approximation with additional unknown control weights leads to an unknown function space in which the optimal solution of the PDE is sought. This makes formulating the problem by directly applying the standard variational formulation, or the method of weighted residuals commonly used in FEM/IGA difficult. To circumvent this difficulty, in the next section, we will devise an elegant adaptivity technique which yields the optimal values of control weights via an iterative process.

Nevertheless, assuming that an initial guess of control weights is available, we shall proceed with the formulation of the problem to find the corresponding unknown control variables. Following a standard Galerkin formulation as described in [44], the equivalent discrete set of governing equation (8) can be obtained in the following form

$$\left(\mathbf{K}_d + \mathbf{K}_r\right)\mathbf{u} = \mathbf{f} \tag{15}$$

where $\mathbf{K}_d$ and $\mathbf{K}_r$ are the diffusion and reaction components of the global stiffness matrix, respectively, and $\mathbf{f}$ is the force vector. These expressions can be obtained as

$$\begin{aligned} \mathbf{K}_d &= \sum_P \left(\int_{\Omega_P} (\nabla \mathbf{R}^u)^T \left(D \nabla \mathbf{R}^u \right) \mathrm{d}\Omega \right) \\ \mathbf{K}_r &= \sum_P \left(\int_{\Omega_P} (\mathbf{R}^u)^T \mathbf{R}^u \, \mathrm{d}\Omega \right) \end{aligned} \tag{16}$$

and

$$\mathbf{f} = \sum_P \left(\int_{\Omega_P} \left(\mathbf{R}^u\right)^T f \, \mathrm{d}\Omega + \int_{\Gamma_{N,P}} \left(\mathbf{R}^u\right)^T h \, \mathrm{d}\Gamma \right) \tag{17}$$

where P denotes the index of a typical NURBS patch. Solving (15) will yield the unknown control variables such that the obtained solution is optimal in the energy sense.

3.2. *Residual-based a posteriori error estimation*

The proposed adaptivity algorithm mainly relies on a posteriori error estimator. A variety of a posteriori error estimation techniques have been proposed in the literature; we refer to [45] for a comprehensive review. One can hypothetically use any of these techniques within the framework of adaptive *w*-refinement. In this research, we employ one of the most common techniques which is the residual-based a posteriori error estimation. Defining the interior and boundary residual terms of the reaction-diffusion PDE in Eq. (8) as Eqs. (18) and (19), respectively

$$r = f + \nabla.\left(D\nabla u_h\right) - \sigma u_h \quad in\ K, \tag{18}$$

$$R = h - \mathbf{n}.D\nabla u_h \quad on\ \partial K \cap \Gamma_N, \tag{19}$$

this estimator establishes that the energy norm of error $\|e_h\|_{E(\Omega)}$ which is defined as

$$\|e_h\|_{E(\Omega)} = \|u - u^h\|_{E(\Omega)} = \left(\int_\Omega \left(\left[\nabla(u-u^h)\right]^T \nabla(u-u^h)\right) \mathrm{d}\Omega\right)^{\frac{1}{2}} \tag{20}$$

is restricted by the following upper bound [11]

$$\|e_h\|^2_{E(\Omega)} \leq \eta_E^2 = C \sum_{K \in \psi} \left(\|r\|^2_{L^2(K)} h_K^2 + \|R\|^2_{L^2(\partial K)} h_K\right) \tag{21}$$

where C is an unknown constant, K is the index of a knot-element with diameter of a maximal inscribed circle h_K and ψ denotes the set of all knot-elements. The derivation of higher order derivatives in Eq. (18) are provided in Appendix B.

3.3. Formulation of adaptive w-refinement

The problem could be simply posed as an optimization problem, where the objective function is the estimated error in a desired norm and the design variables are all or a subset of control weights. The optimization problem can be expressed in the following mathematical form

$$\begin{aligned}
&Find\ \boldsymbol{\chi} = \left\{w_1^u, w_2^u, ..., w_N^u\right\} \\
&\underset{\chi}{Minimize}\ E(\boldsymbol{\chi}) = \eta_E^2(\boldsymbol{\chi}) \\
&Subject\ to: \mathbf{K}(\boldsymbol{\chi})\mathbf{u} = \mathbf{f}(\boldsymbol{\chi}) \\
&\qquad\qquad w_{\min} \leq w_L^u \leq w_{\max},\ L = 1, ..., N
\end{aligned} \tag{22}$$

where $\boldsymbol{\chi}$ represents the vector of design variables with N elements and $\mathbf{K} = \mathbf{K}_r + \mathbf{K}_d$. For simplicity, the global index L is used for numbering the design variables which is defined as $L = j(n_1 + 1) + i + 1$ for the basis (i, j). Further, $w_{\min}$ and $w_{\max}$ denote the selected bounds on the design variables. We will later discuss how these bounds be chosen and how they may affect the performance of the algorithm.

It should also be mentioned here that the equilibrium constraint in Eq. (22) is intrinsically satisfied and need not be externally imposed. In addition, most unconstrained optimization algorithms allow for the imposition of the simple bounds on design variables as required in Eq. (22). Hence, this problem can be regarded as an unconstrained optimization problem in practice.

Moreover, we point out here that a variety of options exists for selecting the vector of design variables in Eq. (22). For instance, based on the type of differential operator $\mathcal{L}(u)$ in Eq. (8) and expected behavior of the solution, one can decide to include all or a selective subset of control weights as design variables. Obviously, including a larger number of control weights in Eq. (22) is expected to result in better accuracy as well as an increased computational cost. In this study, we will only perform global adaptivity. Further, numerous subtleties need to be undertaken for proper and efficient setup of this optimization problem; we will cover these details in the remainder of this paper.

3.4. *Sensitivity analysis*

In order to efficiently solve the optimization problem in Eq. (22) using a gradient based algorithm, cost effective and accurate computation of sensitivities is critical. Fortunately, decoupling of control weights from the underlying geometric space provides the possibility of the derivation of these sensitivities analytically and in a very cost-effective fashion.

Differentiating the right-hand side of Eq. (21) with respect to an arbitrary design variable w_L^u yields the sensitivities of the objective function as

$$\frac{\partial\left(\eta_E^2\right)}{\partial w_L^u} = \sum_{K\in\psi}\left(\frac{\partial\left(\left\|r\right\|_{L^2(K)}^2\right)}{\partial w_L^u}h_K^2 + \frac{\partial\left(\left\|R\right\|_{L^2(\partial K)}^2\right)}{\partial w_L^u}h_K\right) \tag{23}$$

where constant C is assumed to be unity. As observed, evaluation of the above expression requires finding the derivatives of the interior as well as boundary residual terms with respect to control weights. Using Eq. (18), the sensitivities of the interior residual term for a typical knot-element K becomes

$$\frac{\partial\left(\left\|r\right\|_{L^2(K)}^2\right)}{\partial w_L^u} = 2\int_{\Omega_K} r\left(D\left(\frac{\partial^3 u^h}{\partial x^2 \partial w_L^u} + \frac{\partial^3 u^h}{\partial y^2 \partial w_L^u}\right) - \sigma\left(\frac{\partial u^h}{\partial w_L^u}\right)\right)\mathrm{d}\Omega \tag{24}$$

which includes the gradients of u^h as well as its first and second order spatial derivatives with respect to control weights. The derivation of these sensitivities is non-trivial and tedious. Hence, we have provided these derivations in Appendix C.

On the other hand, the sensitivities of the boundary residual term are obtained as

$$\frac{\partial(\|R\|^2_{L^2(\partial K)})}{\partial w^u_L} = -2\int_{\Gamma_N} R\left(\frac{\partial^2 u^h}{\partial x \partial w^u_L}, \frac{\partial^2 u^h}{\partial y \partial w^u_L}\right).\mathbf{n}\, \mathrm{d}\Gamma \tag{25}$$

Similarly, the detailed derivation of the required derivatives in Eq. (25) is provided in Appendix C. It is important to note that due to decoupling of control weights from the underling geometry, the derivatives of h_K as well as $\mathrm{d}\Omega$ and $\mathrm{d}\Gamma$ with respect to design variables vanish. Moreover, this substantially simplifies sensitivity derivation of the spatial derivatives of the field variable in above equations. These simplifications result in substantial reduction of the sensitivity analysis cost.

3.5. Driving the adaptivity process with exact error

To verify the performance and effectiveness of the proposed *w*-adaptivity process, we will also run experiments on problems with existing closed-form solutions in which case the exact error can be calculated using Eq. (20) and be used instead of the estimated error $\eta_E(\chi)$ in Eq. (22) for driving the adaptivity process. The derivation of analytical sensitivities in this case is more straightforward as the sensitivities of the exact solution in Eq. (20) with respect to design variables vanish.

3.6. Treatment of boundary conditions

One of the subtleties for proper implementation of adaptive *w*-refinement is appropriate treatment of boundary conditions. We remind that since the control weights do not lie in the function space of the geometry, we can freely manipulate these *boundary control weights* for improved accuracy. In fact, this is critical for obtaining monotone distribution of error throughout the computational domain and subsequently achieving optimal convergence rates. We discuss the treatment of Neumann (natural) and Dirichlet (essential) conditions separately here.

3.6.1. Natural conditions

Treatment of natural boundary conditions is more straightforward as they naturally arise in the variational formulation. Therefore, one can freely include the boundary control weights associated with Neumann boundaries in the vector of design variables when setting up the optimization problem (22). The only important point here is that, at each iteration of the optimization process followed by updating the design variables, the force vector associated with these (non-zero) boundary conditions need to be reevaluated with the new set of boundary control weights using Eq. (17).

3.6.2. Essential conditions

On the other hand, the imposition of essential conditions during *w*-adaptivity is more intricate and needs precise attention. The strategy depends on whether these conditions are homogeneous or non-homogeneous. The homogeneous (or constant) Dirichlet conditions can be satisfied exactly by simple setting the respective boundary control variables to zero (or the given constant), irrespective of the values of their control weights. Note that in this case, variation of the corresponding boundary control weights does not affect the exact satisfaction of these conditions. Therefore, similar to natural boundary conditions, these boundary control weights can be freely included as design variables in Eq. (22), and their optimal values will be determined by the adaptivity algorithm.

In contrast, finding the optimal boundary control weights of non-homogeneous essential conditions needs special treatment. We remind here that due to the non-interpolatory behavior of spline basis functions, the imposition of non-homogeneous essential boundary conditions in IGA is, in general, non-trivial.

It has been found that direct imposition of these boundary conditions to control variables may lead to significant error and non-optimal rate of convergence [46–48]. Several strategies have been proposed for improved treatment of these conditions. These methods can be classified into two main types: ‘strong’ imposition by approximating the boundary profile in the NURBS space, and ‘weak’ imposition via variational methods.

Most common techniques for weakly imposition of these boundary conditions are Lagrange multiplier methods [49], Nitsche method [46,48,50,51], and penalty method [52]. On the other hand, strong approximation of boundary conditions include least square fitting [53], collocation and transformation methods [47], quasi interpolation techniques [54], and coupling with Lagrange shape functions [55]. We refer to [52] for a rigorous review on these methods.

One can possibly generalize any of the above techniques based on GNURBS and incorporate it in adaptive *w*-refinement. In the following section, we develop such an extension for least-square fitting method and elaborate how this technique could be properly incorporated in the proposed adaptive framework.

3.6.2.1. Least-square minimization using GNURBS

Suppose a Dirichlet condition $u_D(x,y)$ over an arbitrary boundary of the domain Γ_D in Eq. (8) is specified. The problem can be simply posed as a curve fitting problem where a given height function u_D needs to be approximated with u_D^h defined in Eq. (26) over a fixed planar NURBS curve as in Eq. (27)

$$u_D(\rho) \approx u_D^h(\rho) = \sum_{i=0}^{n} R_{i,p}^{u}\left(\rho\right) u_i, \quad 0 \le \rho \le 1 \tag{26}$$

$$\mathbf{C}(\rho) = \begin{Bmatrix} x(\rho) \\ y(\rho) \end{Bmatrix} = \sum_{i=0}^{n} R_{i,p}^{xy}\left(\rho\right) \begin{Bmatrix} x_i \\ y_i \end{Bmatrix}, \quad 0 \le \rho \le 1 \tag{27}$$

where $\rho \in \{\xi, \eta\}$. This can be easily posed as a least-square approximation problem leading to optimal accuracy in L^2*-norm*. Assuming $\{\rho_s \to (x_s, y_s, u_{D,s}) : s \in \mathcal{S}\}$ is the set of N_s collocation points, the error function *E* to be minimized is defined as

$$E(\boldsymbol{\lambda}) = \frac{1}{2}\sum_{s\in\mathcal{S}} \left\| u_D^h(\rho_s) - u_D(\rho_s) \right\|^2 = \frac{1}{2}\sum_{s\in\mathcal{S}} \left\| \sum_{L\in\mathcal{L}^s} R_L(\rho_s) u_L - u_{D,s} \right\|^2 \tag{28}$$

where ρ_s are the corresponding collocation points in the parametric space, $\mathcal{L}^s$ is the set of indices of non-zero basis functions at ρ_s, and $u_{D,s} = u_D(\rho_s)$. Unlike classic NURBS, the vector of design variables $\boldsymbol{\lambda}$ here includes both boundary control variables as well as boundary control weights, i.e. $\boldsymbol{\lambda} = \{u_0, ..., u_n, w_0^u, ..., w_n^u\}$, where similar bounding-box constraints on control weights as in Eq. (22) are to be satisfied. With this set of design variables, Eq. (28) becomes a constrained non-linear least-square problem which can be solved using any of the existing solvers such as trust-region-reflective available in MATLAB. Alternatively, to avoid solving a non-linear problem, one can employ a two-step algorithm developed by the authors in [3] via the extension of the original algorithm for NURBS approximation proposed by Ma [41,43]. This algorithm leads to two separate linear systems of equations; a homogenous system which yields the optimal control weights and a non-homogenous one that yields the corresponding optimal control variables. Further details of this algorithm can be found in [3].

The reported numerical results in [3] suggest a dramatic improvement in the accuracy compared to least-square fitting with NURBS in both cases of smooth and rapidly varying height functions. In particular, it has been shown that for smooth functions, the above approximation yields one order faster convergence than approximation with NURBS.

Another significant advantage of including the boundary control weights as additional design variables for the approximation of Dirichlet conditions is the possibility of exact satisfaction of these conditions for a wider range of functions compared to NURBS. In particular, any Dirichlet conditions specified as a function of the form

$$u_D(x,y) = \frac{f(x,y)}{g(x,y)} \tag{29}$$

where f and g are arbitrary polynomials, can be exactly represented via solving Eq. (28) by appropriate choice of degrees of basis functions (p, q). We will provide a numerical example later and clarify why making direct use of NURBS-based IGA does not naturally allow for the exact imposition of the entire functions in this class.

3.6.2.2. Incorporation in adaptive w-refinement

We provide here further details on how to impose the obtained optimal boundary conditions from previous section on the problem. The process of imposing these conditions on Eq. (15) is quite similar to other classic ways of strongly imposing the Dirichlet conditions, apart from the subtle fact that, in the current case, one should first solve the problem in Eq. (28) and feed the obtained optimal boundary control weights to the basis functions used for defining the field variable function space in Eq. (12) prior to the construction of stiffness matrix in Eq. (15).

Moreover, strictly speaking, these boundary control weights (associated with non-homogeneous essential boundary) must be excluded from the design variables in Eq. (22). We emphasize here that, while using any of the *strong* methods, discussed in Section 3.6.2, for the imposition of non-homogeneous essential conditions, including these boundary control weights in Eq. (22) as design variables will result in erroneous results.

Finally, it needs to be mentioned that merely imposing the optimal boundary conditions obtained by the above-discussed algorithm before executing *w*-adaptivity may not result in improved accuracy, or may even deteriorate the solution compared to imposing the Dirichlet conditions using classic linear least-square fitting. This could be attributed to the fact that these boundary basis functions are shared with the interior ones; hence, while improving the accuracy on the boundary (e.g. along ξ), manipulating the control weights on the boundary may deteriorate the natural balance between basis functions inward the domain (along η). However, this issue will be fully resolved after performing *w*-adaptivity.

3.7. Numerical integrations

Another significant aspect for effective implementation of *w*-refinement which should be carefully addressed is the employed quadrature rule. It is important to note that in order for *w*-adaptivity algorithm to perform effectively and lead to optimal solution, all the numerical integrations need to be performed with adequate accuracy. This includes integration of the stiffness matrix in (16), the load vector in (17), the estimated error in (21), as well as the sensitivity expressions in (24) and (25).

Several studies have been carried out for the development of efficient quadrature rules in tensor product splines; see e.g. [56–61]. Nevertheless, to our knowledge, all these studies ignore the variation of the weight function in the denominator and devise quadrature rules which satisfy the exactness condition for the non-rational basis. The argument is that "*often the weight appearing in the denominator of the NURBS basis functions changes slowly (compared to the polynomial numerator of the NURBS basis functions) because they are piecewise smooth functions on the initial coarse mesh where the geometry is exactly represented. Then, it is a common practice to select quadrature rules that give exact integration when the NURBS denominator is constant*." [61].

Nonetheless, as will be seen later in numerical results, this assumption will not hold true in *w*-adaptive IGA, especially when the adaptivity is applied to problems with sharply varying solutions. In the current work, we will simply use a finer quadrature globally in such cases, to make sure all the integrations are calculated with adequate accuracy. Although this does not seem to be an efficient way for addressing this issue, we emphasize here that devising an efficient quadrature rule for these highly rational tensor product splines has not been a concern of us here and is beyond the scope of this paper. This is in fact an interesting subject for further research in *w*-adaptive IGA.

On the other hand, another observation reported by Hughes et al. [1] is consistent with our experiments with *w*-adaptive IGA. To assess the validity of assuming NURBS as B-splines of the same polynomial order for deciding on the number of quadrature points, they perform tests in which they systematically increase the number of quadrature points. They report that "*for sufficiently fine meshes no differences in results were discernible. However, coarse meshes required more integration points due to large variations in the geometrical mapping. More research needs to be done to determine a robust strategy covering all situations.*" [1]

Similarly, our numerical experiments with *w*-adaptive IGA suggest that more quadrature points are required to achieve the same level of accuracy in integration when coarser meshes are used. However, in our case, this is mainly caused by large variations in the denominator of solution basis, that is

$$W^{u}(\xi,\eta)=\sum_{k=0}^{n_1}\sum_{l=0}^{n_2}N_{kl}^{p,q}(\xi,\eta)w_{kl}^{u} \tag{30}$$

rather than in geometrical mapping. Therefore, even in cases where the geometrical mapping is constant, a larger number of quadrature points will be required when *w*-adaptivity is performed on coarse meshes. The details of selection of quadrature rule will be included for all the numerical studies in this paper.

3.8. Computer implementation aspects

The flowchart in Fig. 6 summarizes the process of adaptive *w*-refinement.

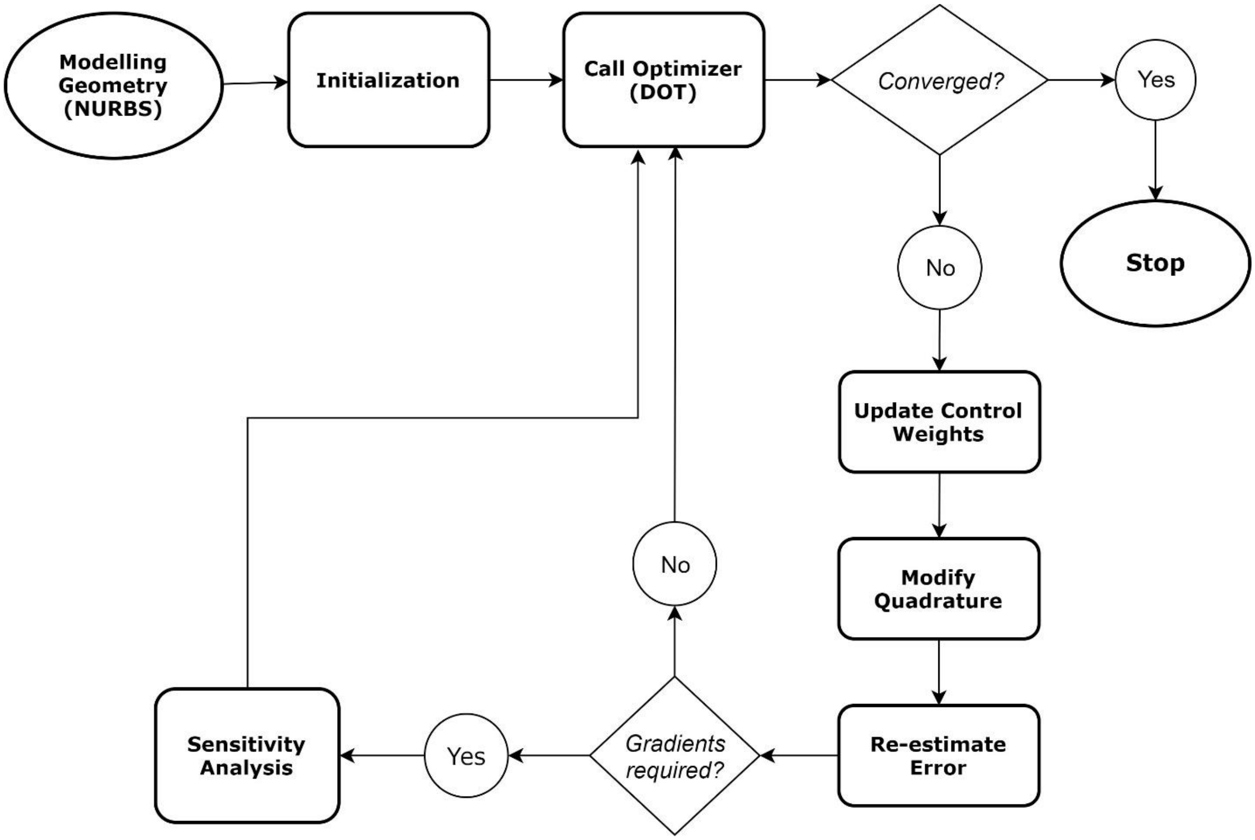


Fig. 6. The procedure of adaptive *w*-refinement.

The implementation of this procedure is quite straightforward and can be easily included in an existing IGA package. Certain details, however, should be considered for the efficient implementation of this algorithm. For instance, at the initialization step, one can evaluate and pre-store all the entities constructed using the basis functions of the geometry as these entities are invariant during the optimization process. Further, if non-homogeneous boundary conditions exist, they need to be evaluated only once at this stage. On the other hand, the solution basis functions as well as all their resultants, such as natural boundary conditions, change at each iteration and need to be evaluated iteratively prior to re-estimation of the error. Finally, as shown in Fig. 6, we have included a particular step for the modification of quadrature followed by updating the control weights in the flowchart of adaptivity for the sake of completeness, even though as discussed in previous section, it has not been implemented in this work and will be studied in a future research.

3.9. Analogy with other refinement techniques

Recalling the definition of function space refinement techniques in Section 1.1, we can see that the proposed method conforms to this definition as it adaptively enriches the function space and improves the accuracy without perturbing the underlying geometry or its parameterization. However, unlike *h* and *p* refinements, this procedure preserves the number of basis functions and only changes their contribution to the function space by adjusting their weights. It is important to mention that since the solution space is always simply discretized by NURBS basis functions, during this adaptive procedure, all the properties of basis functions such as their linear independence and partition of unity are naturally preserved. The proof of linear independence of NURBS basis functions can be found in [62]. Further, since no additional basis functions are introduced, the structure of stiffness matrix remains unchanged in terms of sparsity and bandwidth. The condition number, however, will change and needs to be monitored. In the following numerical experiments, we will study the effect of performing *w*-adaptivity on the conditioning of the stiffness matrix in different types of problems.

Another important aspect of this adaptive method is that, unlike many of existing methods for addressing problems with irregularities such as DPG [63,64], anisotropic NURBS approximation [65] etc., it does not require any prior knowledge of the solution behavior. The proposed method is in some sense quite similar to *r*-refinement as both methods attempt to minimize an estimation of the error by solving an optimization problem. However, unlike *r*-refinement, the design variables in proposed method are transferred to the solution space. This change of variable eliminates all the deficiencies of *r*-refinement discussed in Section 1.1 and makes it a competitive algorithm with existing adaptivity techniques.

3.10. Limitations

It is clear that performing *w*-refinement provides a trade-off between the achieved improvement in accuracy and the extra computational cost for solving the adaptivity problem illustrated in Fig. 6. Of course, the algorithm is fruitful when this trade-off is reasonable, i.e. when the obtained accuracy dominates the additional computational cost. We note here that this trade-off depends on multiple factors such as the type of governing PDE, the behavior of the solution (number and orientation of existing layers, in particular), the accuracy of an initial guess for control weights, the employed degree of basis functions, the efficiency of the employed non-linear optimization solver etc. Detailed investigation of all these factors is beyond the scope of this study. However, considering the flowchart of *w*-refinement in Fig. 6, we can identify the main sources of the additional computational cost as discussed below.

According to our numerical studies, the primary source of this additional cost is due to re-solving the governing PDE at each iteration followed by updating the control weights. Another major portion of the computational expense is due to re-evaluation of the objective function, i.e. estimated error in Eq. (21), as well as its sensitivities in Eqs. (23)-(25), repeatedly. We recall that the calculation of these expressions requires evaluating higher order derivatives of rational basis functions which are more complicated and expensive compared to polynomial basis functions. In the following numerical studies, we will show how these components of the algorithm contribute to the overall computational expense.

Consequently, we can see that the cost of adaptive *w*-refinement directly depends on the number of optimization iterations, where the cost of each iteration mainly relies on the number of quadrature points as well as the employed number of design variables. Therefore, the key factors for improving efficiency are to reduce the number of optimization iterations, the number of quadrature points, as well as the number of considered design variables. We will later suggest some preliminary ideas for improving these factors in Section 5.

4. Numerical Experiments

To demonstrate the performance of the proposed adaptivity technique, in this section, we apply this method to a variety of problems with different solution behaviors and compare the obtained results with those of NURBS-based IGA.

The presented numerical results are obtained by the implementation of this adaptive algorithm in PGI Visual FORTRAN [66]. The conjugate gradient method with Incomplete Lower-Upper (ILU) preconditioner has been used for solving the system of equations. All the reported condition numbers of the stiffness matrix are measured in L^1-norm. The optimization problem is solved by the BFGS method available in the Design Optimization Tool (DOT) [67]. The initial guess, bounding constraints on design variables, and termination criteria are selected as follows.

Initial guess: It is well-known that in non-linear optimization problems, starting with a good initial guess can significantly improve the performance and efficiency of the algorithm. Not only can this make the algorithm converge faster, but also it increases the possibility of achieving the optimal results. Finding a suitable initial guess of control weights for a steady transport problem which is studied here, however, is non-trivial. In the following numerical experiments, unless stated otherwise, we will start the optimization process assuming unit values for all design variables, i.e. B-spline basis functions. Nevertheless, in certain cases, we will suggest effective ideas which can be used to obtain an improved initial guess.

Bounding constraints: Theoretically, selecting an infinitesimal positive value for $w_{\min}$ and an arbitrary larger value for $w_{\max}$ in Eq. (22) will cover the whole search space. However, these values affect the performance of the algorithm. In particular, selecting too small values for $w_{\min}$, if taken by design variables during adaptivity, will result in deterioration of the conditioning of the stiffness matrix which is not desired. Our experiments show that with the following suggested bounds on design variables, the algorithm works effectively

$$10^{-4} \le w_L^u \le 3.0,\ L = 1,\ldots,N \tag{31}$$

Further research, however, is needed for finding the optimal bounds in different types of problems.

Termination criteria: As illustrated in Fig. 6, the optimization process is terminated whenever a convergence criterion is satisfied. The employed design optimization tool (DOT) uses several criteria to make the decision when to stop. These include: if a maximum number of 100 iterations is reached; Kuhn–Tucker conditions are satisfied 'reasonably', defined as when all components of the gradient of objective are less than 1*e*-3; and the so-called diminishing returns criterion when either the relative or absolute change in the objective between two consecutive iterations is less than the specified tolerance 1*e*-6. Further details on these criteria can be found in [67].

4.1. Test Case 1- Poisson equation with a smooth solution

In the first numerical experiment, we investigate the performance of *w*-adaptivity on problems with smooth solutions. For this purpose, we consider the following governing Poisson equation with homogeneous boundary conditions on all edges

$$\begin{aligned} -\Delta u &= 2\pi^2 \sin(\pi x)\sin(\pi y), \qquad (x,y) \in [0,1]\otimes[0,1] \\ u_D(x,0) &= u_D(x,1) = u_D(0,y) = u_D(1,y) = 0 \end{aligned} \tag{32}$$

whose closed-form solution is given by:

$$u(x,y) = \sin(\pi x)\sin(\pi y) \tag{33}$$

which is illustrated in Fig. 7.

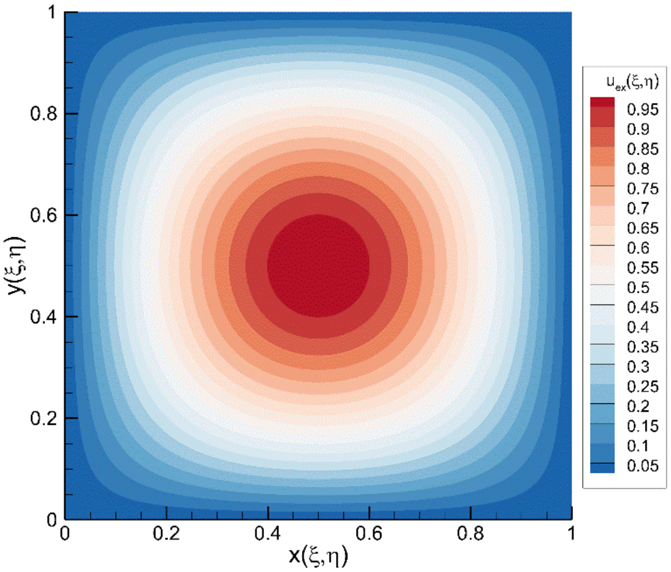
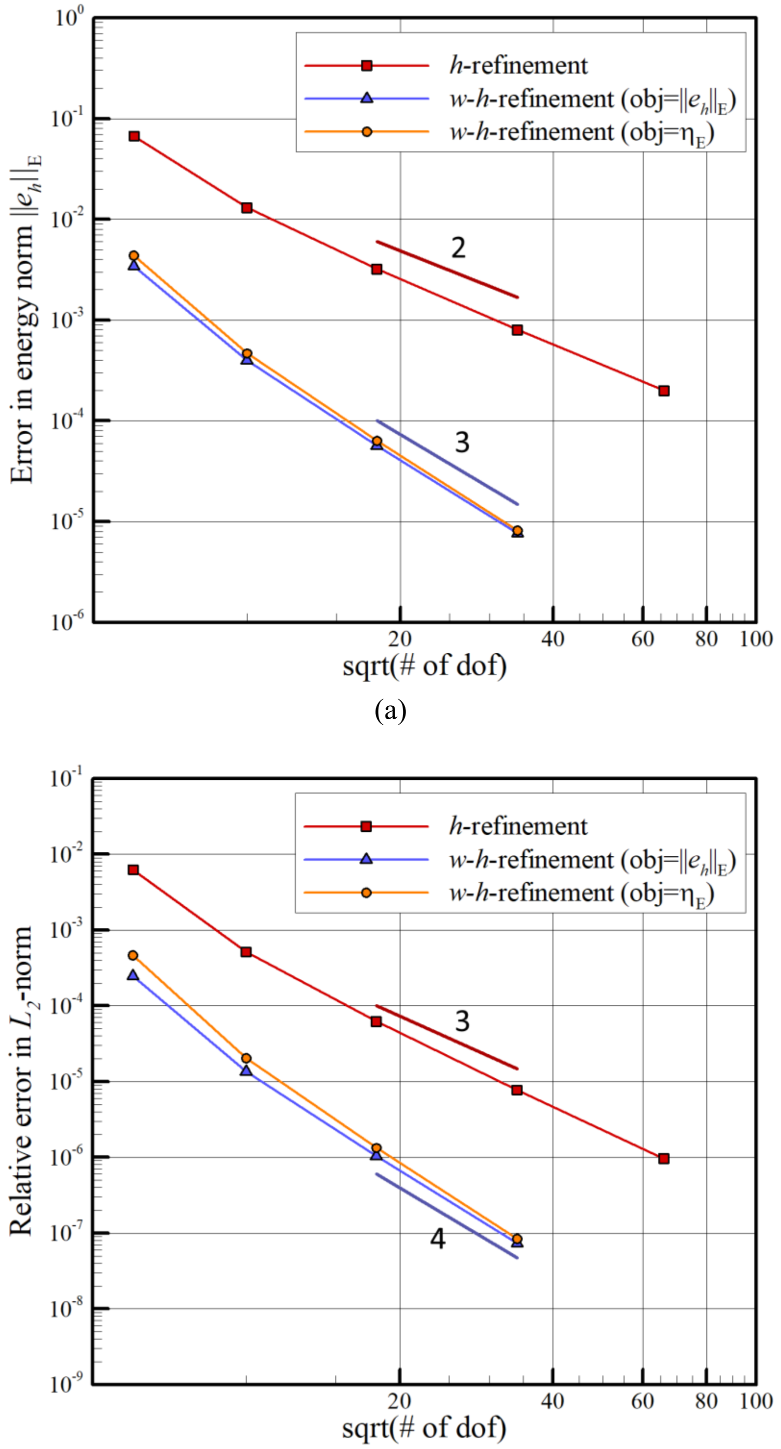


Fig. 7. Exact solution of the Poisson equation in (33).

We perform a convergence study by starting with a mesh of 4×4 bi-quadratic elements with linear parametrization, and refining up to 64×64 elements. At each level of discretization, followed by *h*-refinement, we perform *w*-adaptivity and refer to this procedure as *w*-*h*-refinement. In all cases, we use a set of 4×4 quadrature points per element for integration. Further, to examine the reliability of the employed residual based a posteriori error estimator, we drive the adaptivity process with both the exact error in (20), as well as the estimator in (21). The obtained results are represented in Fig. 8.



(a)

(b)

Fig. 8. Convergence rates of *h*-refinement versus *w*-*h*-refinement in (a) energy norm, and (b) relative L^2-norm.

It is interesting to notice that, followed by performing *w*-adaptivity, the convergence rates have been improved by one order in both energy as well as L^2 norms, indefinitely. We note here that these are representative results of our studies with different degrees of basis functions on a variety of second order elliptic PDEs with a smooth solution. Our numerical results suggest the optimal rate of convergence of NURBS basis with optimal weights are $\mathcal{O}(p+2)$ in L^2 and $\mathcal{O}(p+1)$ in energy norm for these problems.

Moreover, the figure shows a good agreement between the results obtained with guiding the adaptivity process by the exact error as well as the estimated error. As can be seen, this agreement improves as the resolution of the mesh increases. This is expected since the performance of estimator is directly related to the accuracy of solving the PDE. That is, the more accurate the PDE is solved, a better estimation of the error is provided by the estimator.

The history of adaptivity process guided by the estimator is depicted in Fig. 9 for the 4×4 mesh. The plotted results are obtained by assuming the unknown constant $C=1$ in Eq. (21). It is interesting to notice that by decreasing the objective function (η_E) at each iteration, the exact error $\|e_h\|_E$ also diminishes by a quite similar rate, which implies the reliability of the employed residual based a posteriori estimator. Note that the offset between the estimated and exact error is unimportant and is caused by the assumption made for the unknown constant *C*.

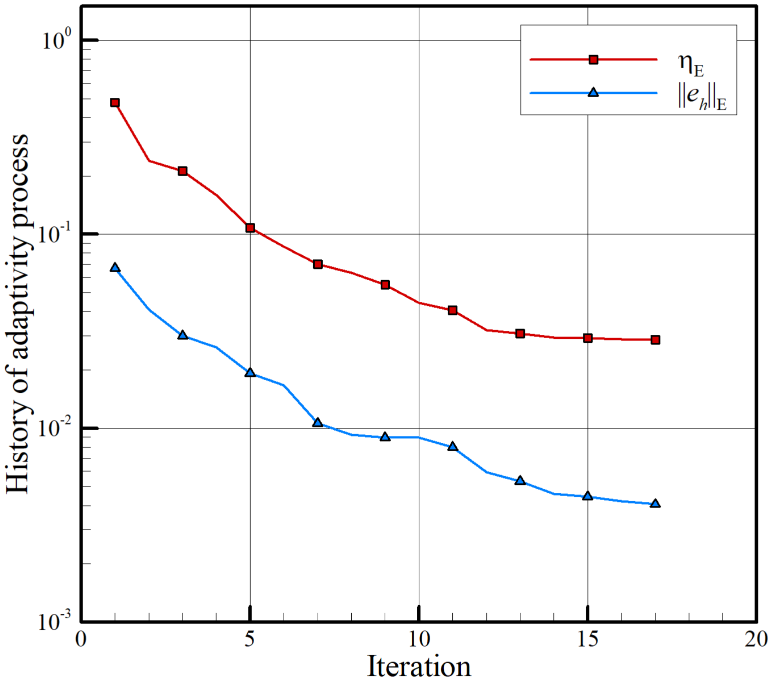
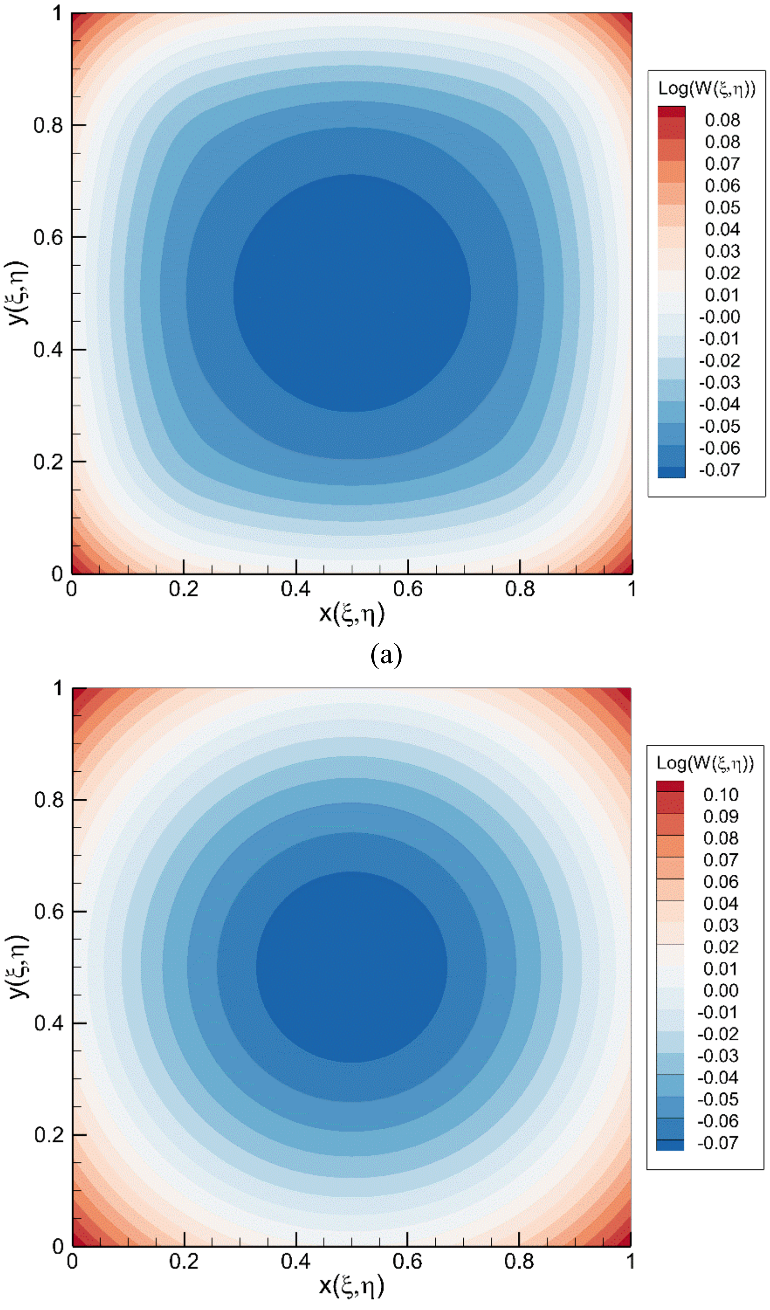


Fig. 9. History of adaptivity process on the 4×4 mesh.

Furthermore, the optimal variation of the denominator of solution $W^{u}(\xi,\eta)$ is depicted in Figs. 10(a) and (b), for the 4×4 and 32×32 meshes, respectively. The adaptivity process is driven by the estimator in both cases.



(a)

(b)

Fig. 10. Optimal variation of $W^{u}(\xi,\eta)$ after performing *w*-adaptivity on the (a) 4×4 mesh, and (b) 32×32 mesh.

Comparing these figures, we can see that the optimal variation of denominator is converging towards a perfectly circular distribution, suggesting the possibility of existence of a closed-form expression. Moreover, we can see that in both cases the variation of the denominator is very small. This is expected to be always the case as long as the solution of the PDE is smooth.

The condition numbers of the stiffness matrix before and after adaptivity are presented in Table 1 for different mesh resolutions. As the table shows, no noticeable change in the condition number of the system of equations for any of the presented cases has occurred followed by performing *w*-adaptivity.

Table 1. Condition number of the stiffness matrix for different meshes.

Mesh	IGA	*w*-adaptive IGA	$\frac{w\text{-IGA}}{\text{IGA}}$
4×4	6.07E+00	6.48E+00	1.1
32×32	1.86E+02	1.86E+02	1.0

Finally, the relative amount of computational time consumed by different steps of the algorithm throughout the adaptivity process are shown in Table 2. The reported numbers are an average of the times calculated for different mesh resolutions.

Table 2. Relative computational times of different steps of *w*-refinement.

Basis	Objective & Sensitivities	Analysis	Other
30%	20%	45%	5%

According to this table, the largest computational time is consumed by the analysis step (construction of the stiffness matrix and solving the system of equations). The second time-consuming part is the evaluation of the basis functions together with their higher order spatial derivatives. Sensitivity analysis and evaluation of the objective function are also the next major time-consuming component of the algorithm.

4.2. Test Case 2- Reaction-diffusion equation with a rough solution

In this example, we examine the effectiveness of the proposed adaptive method on a problem with a rough solution. To this end, we consider a singularly perturbed reaction-diffusion equation with homogeneous essential boundary conditions on all edges

$$\begin{aligned} -\varepsilon^2 \Delta u + u = f, \qquad & (x,y) \in [0,1] \otimes [0,1] \\ u_D(x,0) = u_D(x,1) = u_D(0,y) = u_D(1,y) = 0 & \end{aligned} \tag{34}$$

where ε is the diffusion parameter, and f is determined by the exact solution

$$u(x,y) = \sin(\pi y)(1 - e^{-x/\varepsilon})(1-x) \tag{35}$$

which is depicted in Fig. 11 for $\varepsilon = 0.01$.

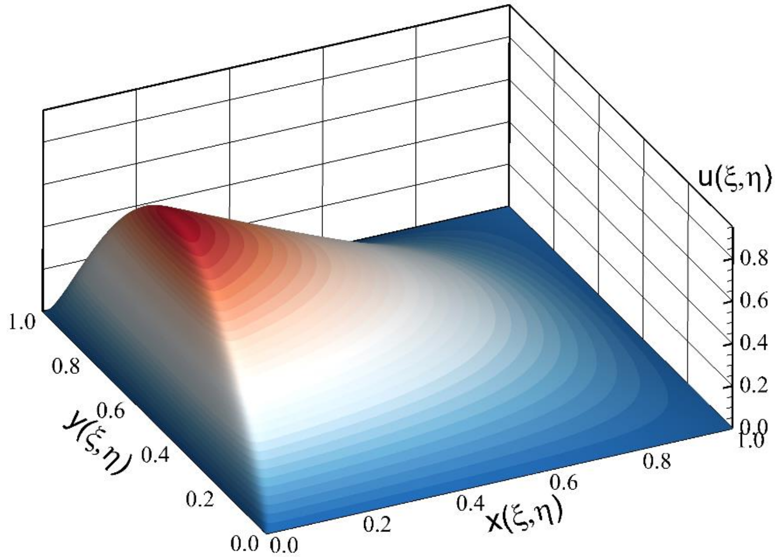
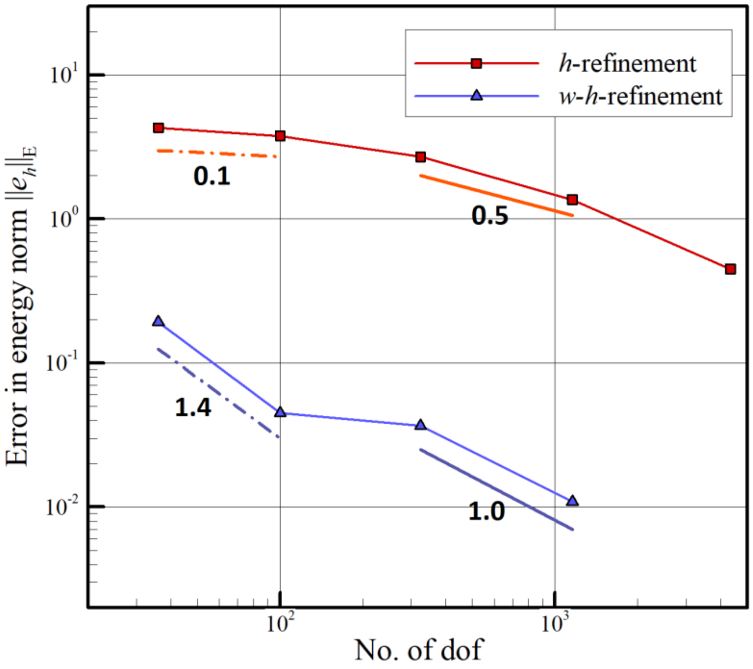


Fig. 11. Exact solution of reaction-diffusion equation in (35) for $\varepsilon = 0.01$.

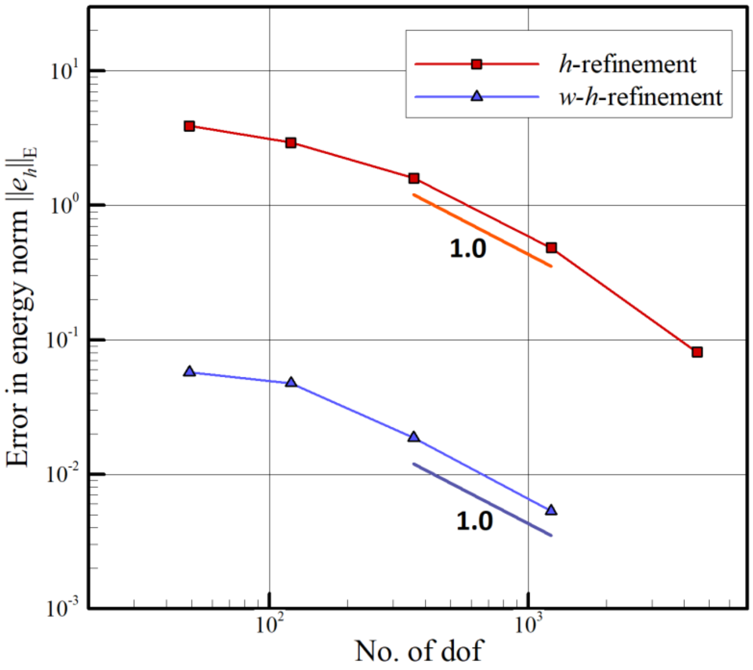
As the figure shows, the solution features a steep boundary layer of width ε near the left edge. Similar to the previous example, we study the convergence of *h* and *w*-*h*-refinement by performing *w*-adaptivity on different meshes of 4×4 to 32×32 elements. We employ sets of $15\times15, 12\times12, 10\times10$ and 8×8 quadrature points per knot-element for integration on the coarsest to finest mesh, respectively. Our experiments show no discernable change in results with using finer quadratures. However, as discussed earlier, a more systematic way is needed for deciding on the appropriate number of quadrature points.

In this example, we start the adaptivity process with an improved initial guess for control weights in the case of finer meshes, which is obtained by *h*-refining the optimal rational function space of the coarser mesh obtained by *w*-adaptivity. For this purpose, *h*-refinement is performed separately on the geometry and optimal solution function spaces of a coarser mesh. Our experiments indicate that this procedure leads to improved results and reduced number of iterations in problems with sharply varying solutions, such as the current example, where large variations in the denominator occur.

The obtained convergence rates are represented in Figs. 12(a) and (b) for bi-quadratic and bi-cubic basis functions, respectively.



(a)

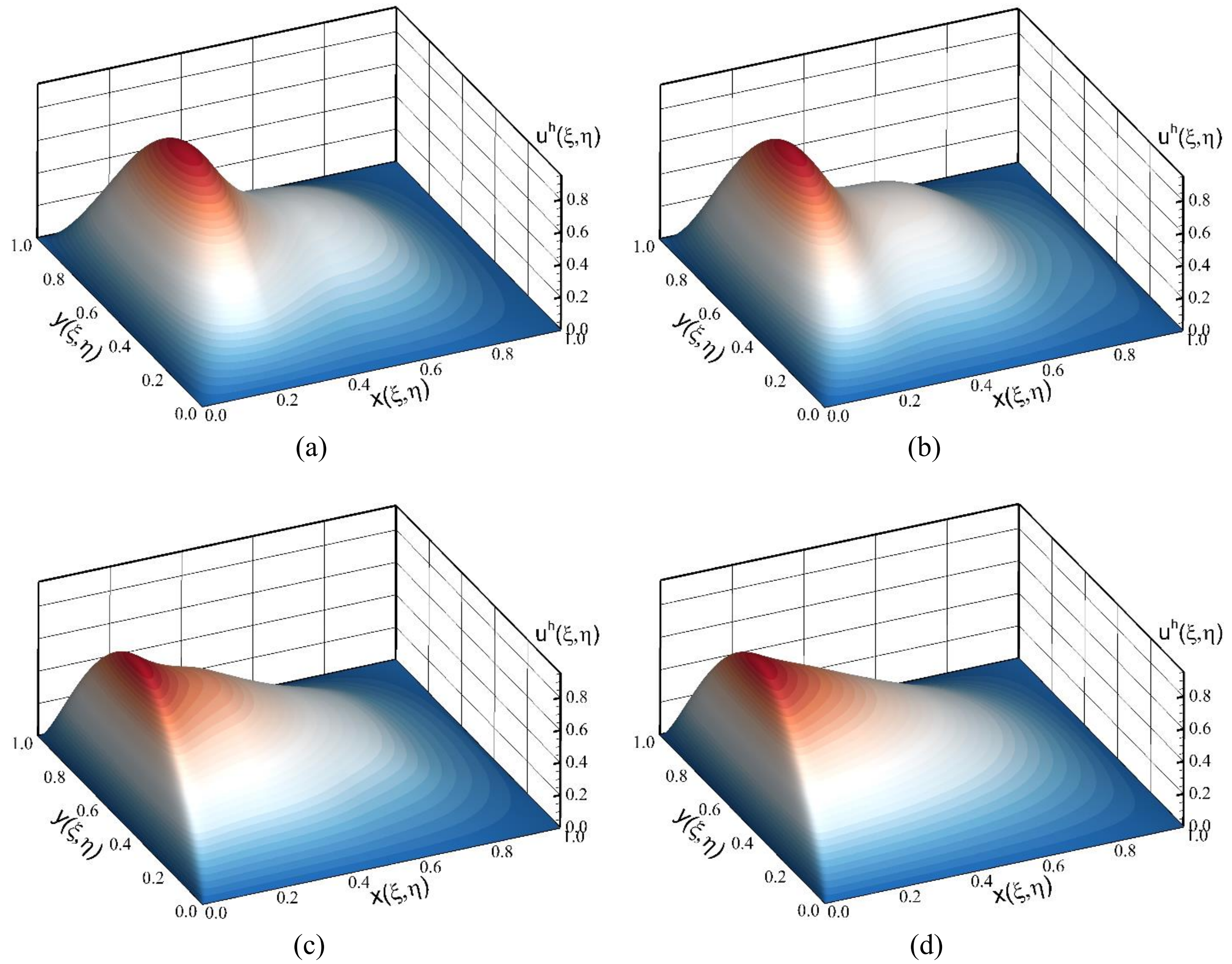


(b)

Fig. 12. Convergence rates of *h*-refinement versus *w*-*h*-refinement in energy norm for (a) quadratic, and (b) cubic case.

As observed, in both cases a considerable improvement in the accuracy of approximation has been achieved. The improvement of the convergence rate, however, is not persistent and also differs for different degrees. Comparing Figs. 12(a) and (b), we can see that the improvement in the convergence rate with quadratic basis is more evident, although this improvement is not persistent for all levels of refinement. It is worth noting that this non-uniform behavior of the convergence rate is common in other types of adaptive methods such as adaptive local *h*-refinement; see e.g. [11]. Further study is required to better perceive the effect of *w*-adaptivity on the convergence rate in problems with sharp layers. It is interesting to notice that in both cases, the obtained accuracy on a 4×4 mesh, after performing *w*-adaptivity, is better than that of a 64×64 mesh with B-spline functions. However, we reiterate that there is an additional computational cost for solving the optimization problem in *w*-adaptive IGA.

For better insight, the plots of approximated solution before and after adaptivity are depicted in Fig. 13. Also, the corresponding distributions of error are represented in Fig. 14.



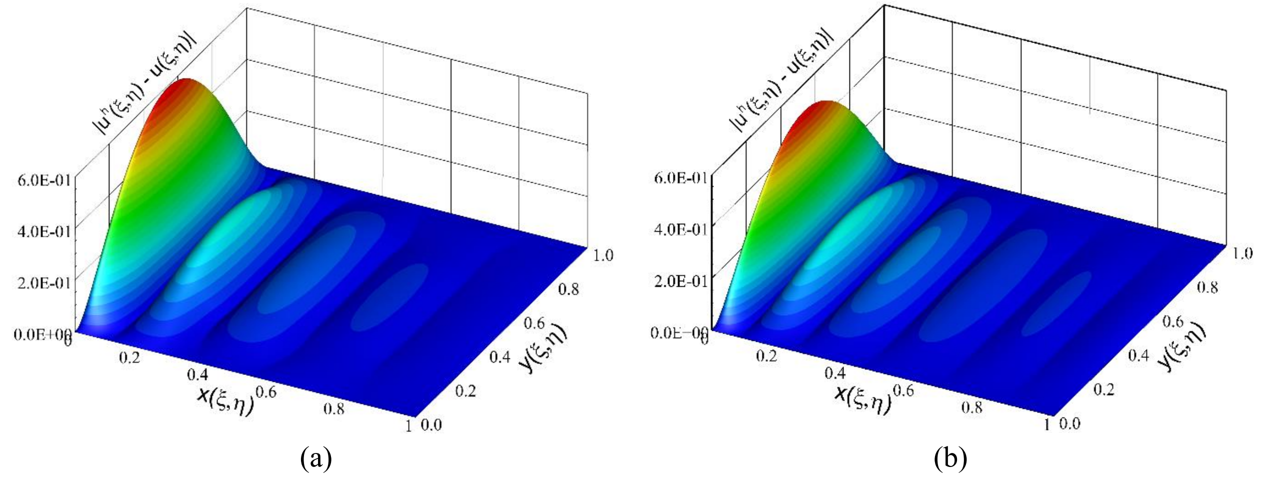
(a)

(b)

(c)

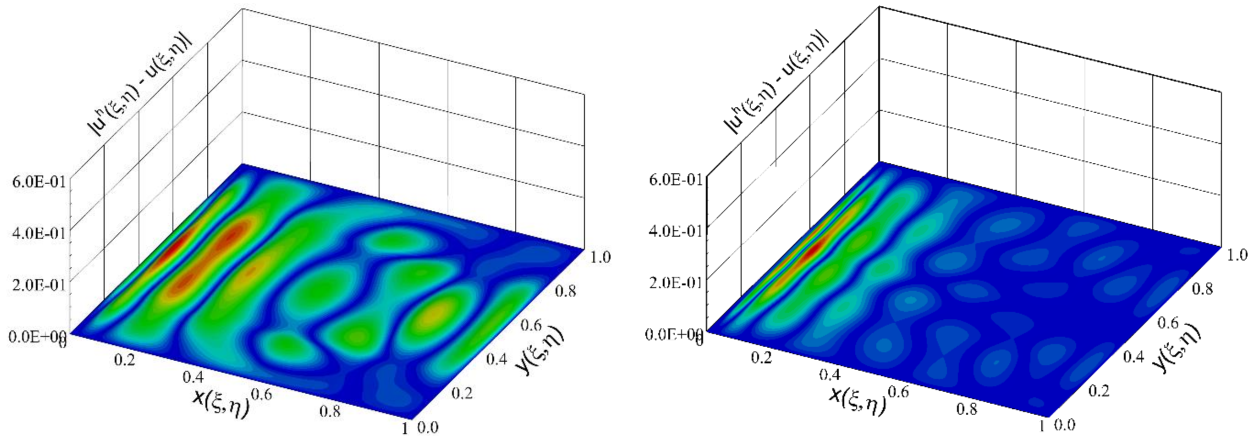
(d)

Fig. 13. Approximate solution of reaction-diffusion problem with a 4×4 mesh using (a) quadratic IGA, (b) cubic IGA, (c) quadratic *w*-adaptive IGA, and (d) cubic *w*-adaptive IGA.



(a)

(b)

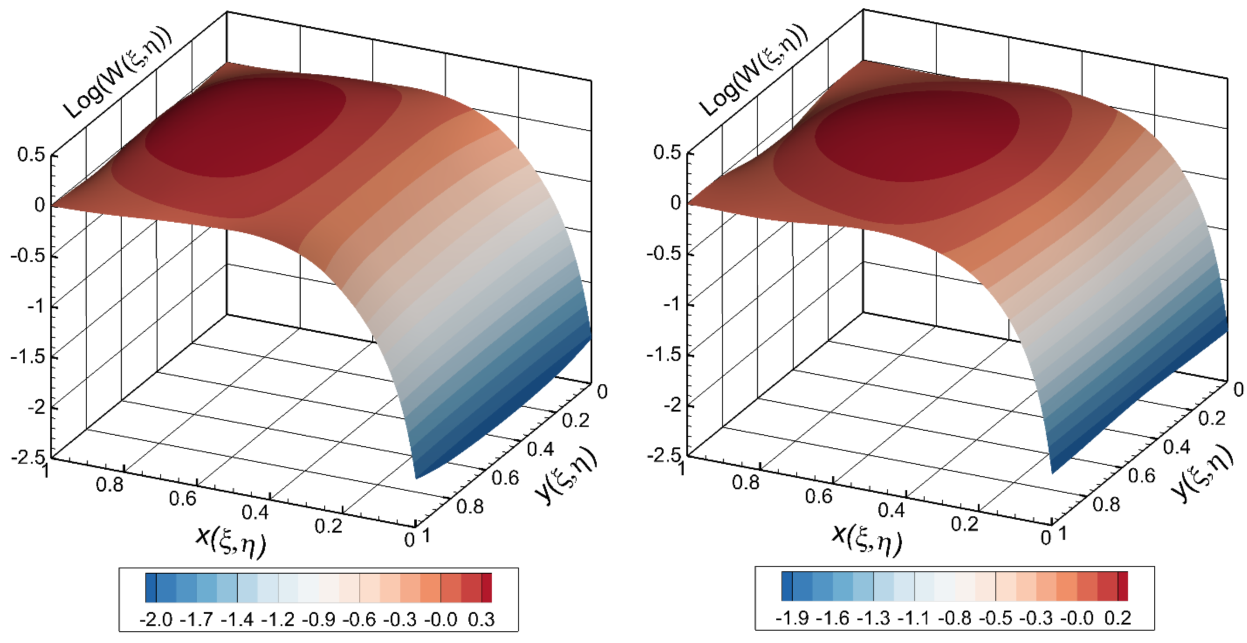


(c) (d)

Fig. 14. Error distribution of solving reaction-diffusion problem with a 4×4 mesh using (a) quadratic IGA, (b) cubic IGA, (c) quadratic *w*-adaptive IGA, and (d) cubic *w*-adaptive IGA.

Having examined these figures, we can observe that after performing adaptivity, in both cases, the boundary layer has almost been completely resolved and a monotone distribution of error has been achieved.

The variation of denominator function $W^u(\xi,\eta)$ of the optimal solutions are demonstrated in Fig. 15. It is interesting to notice that, unlike the previous example with a smooth solution, the variation of denominator in this case is very large. As the figure shows, the magnitude of this variation exceeds two orders of magnitude in the vicinity of the boundary layer. This experiment, in fact, reveals another significant aspect of using rational splines as a basis for analysis. While piecewise smooth polynomials are inherently poor for the approximation of fields with steep localized gradients, approximation with rational bases promises an effective tool for addressing this deficiency.



(a) (b)

Fig. 15. Variation of solution denominator after *w*-adaptivity on the 4×4 mesh with (a) quadratic, and (b) cubic basis functions.

The condition numbers of the stiffness matrix before and after adaptivity are presented in Table 3 for different mesh resolutions. Comparing the condition numbers in Table 3 with those of the previous example in Table 1, we can observe moderate increase of the condition numbers in the current example.

Table 3. Condition number of the stiffness matrix for different meshes.

Mesh	Degree	IGA	*w*-adaptive IGA	$\frac{w\text{-IGA}}{\text{IGA}}$
4×4	2	5.65E+01	2.23E+02	3.9
	3	8.04E+02	1.79E+03	2.2
32×32	2	1.97E+01	3.14E+01	1.6
	3	1.77E+02	2.56E+02	1.4

Our experiments indicate that the increase of condition number is related to the variation of solution denominator, that is, larger variations of the denominator result in further increase of the condition number. Nevertheless, despite having orders of magnitude variation in the denominator of basis functions in the current case, according to Table 3, the condition numbers have not increased by any more than 4 times for any of the presented experiments. Finally, it is worth noting that for more complex problems with meshes of practical scale, the increase of condition number can possibly be larger. Suitable preconditioners perhaps need to be developed and employed in such scenarios.

4.3. Test Case 3- Poisson equation with a closed-form solution in rational space

In this example, we attempt to reveal another crucial merit of *w*-adaptivity, which is its capability to solve problems whose exact solution lie in the rational space with machine precision, irrespective of how coarse the discretization is. Towards this goal, we consider the following Poisson equation over a quarter ring with Dirichlet conditions on all edges

$$\begin{aligned} -\Delta u &= f(r,\theta) \\ u &= u_D \quad \text{on}\, \Gamma_D \end{aligned} \tag{36}$$

where the source term *f* and boundary conditions u_D are specified by the closed-form solution (see Fig. 16)

$$u(r,\theta) = \frac{\cos(\theta)}{r^2}. \tag{37}$$

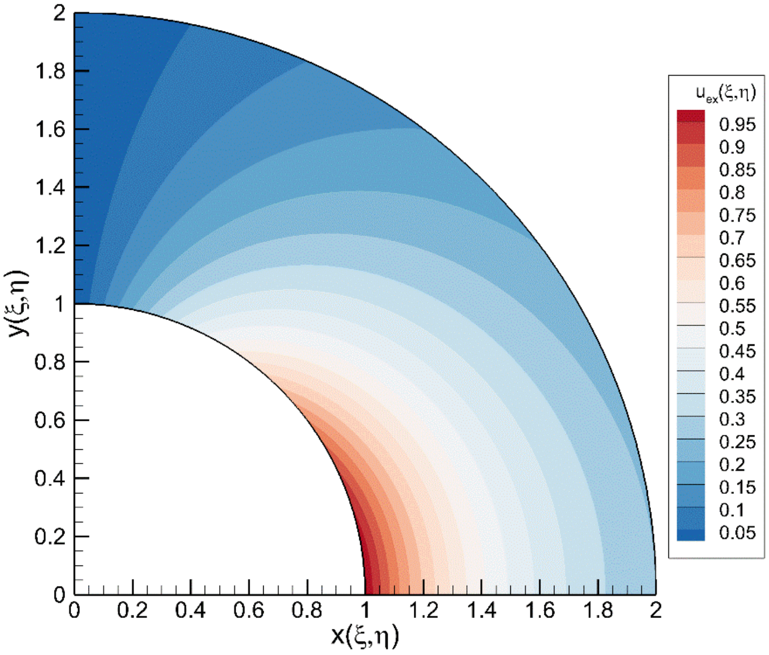


Fig. 16. Exact solution of Poisson equation in (37).

The configuration of the problem is illustrated in Fig. 17.

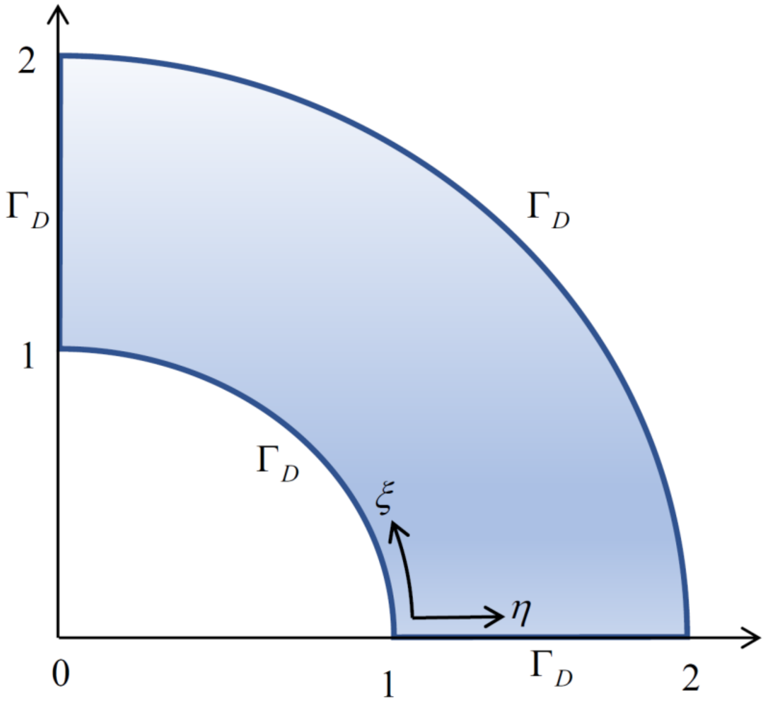


Fig. 17. Configuration and boundary conditions of the quarter ring.

We construct a computational model with two bi-quadratic NURBS elements with normal parameterization as shown in Fig. 18. The analytical values for the coordinates and weights of control points are provided in Table 2. Also, the knot-vectors are selected as $\mathbf{\Xi}=\{0,0,0,0.5,1,1,1\}$ and $\mathbf{H}=\{0,0,0,1,1,1\}$. It can be shown that the exact solution in (37) can be recovered by using the analytical values shown in Table 2 for control variables and control weights. According to this table, one can clearly see that the control weights required for recovering the exact solution in (37) have nothing to do with those of the underlying geometry. This is trivial as these control weights are determined by the governing PDE in Eq. (36).

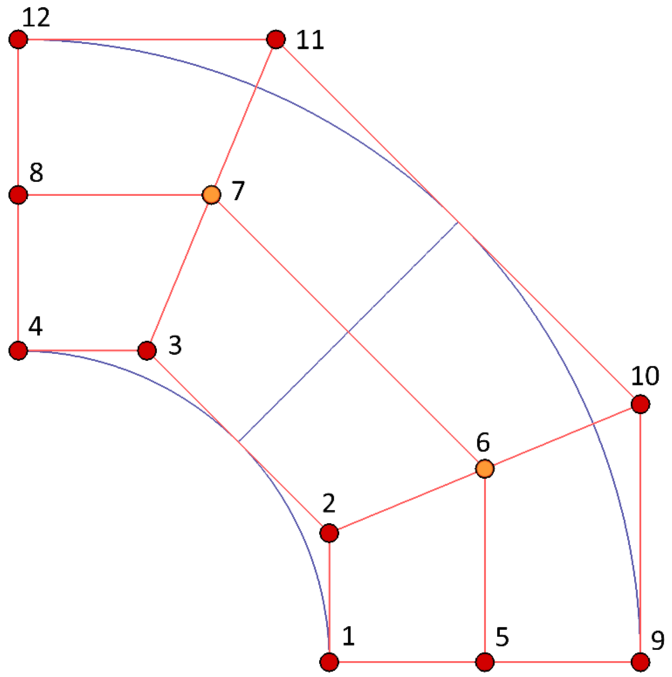


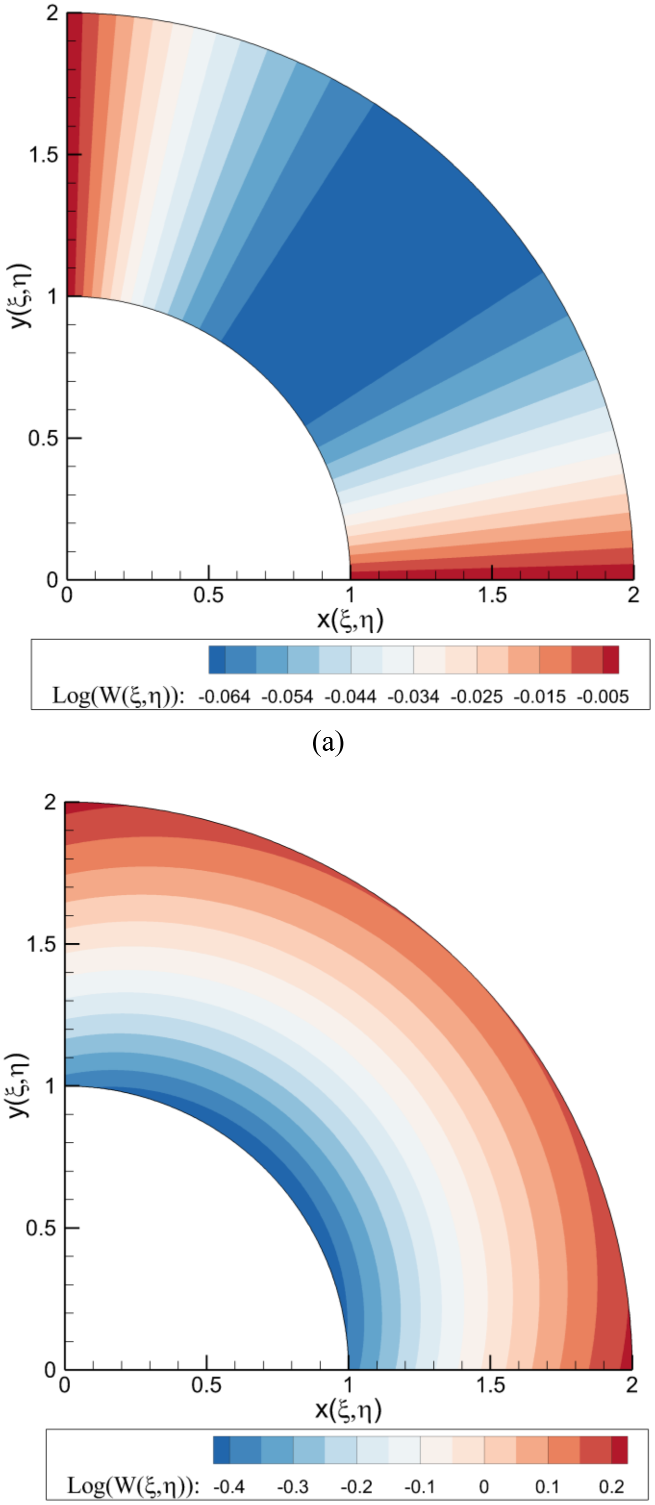
Fig. 18. Control and physical mesh of the quarter ring with normal parameterization.

Table 2. Analytical values for exact modelling and solution of the problem.

L	x_L	y_L	w_L^G	u_L	w_L^u
1	1.0	0.0	1.0	1.0	a
2	1.0	a	b^2	1.0	$0.5\,c$
3	a	1.0	b^2	a	$0.5\,c$
4	0.0	1.0	1.0	0.0	a
5	1.5	0.0	1.0	0.5	$2a$
6	1.5	$1.5\,a$	b^2	0.5	c
7	$1.5\,a$	1.5	b^2	$0.5\,a$	c
8	0.0	1.5	1.0	0.0	$2a$
9	2.0	0.0	1.0	0.25	$4\,a$
10	2.0	$2a$	b^2	0.25	$2\,c$
11	$2a$	2.0	b^2	$0.25\,a$	$2\,c$
12	0.0	2.0	1.0	0.0	$4\,a$

$a = \tan(\frac{\pi}{8}), b = \cos(\frac{\pi}{8}), c = \cos(\frac{\pi}{4})$

For better insight, the variation of denominator of the geometry as well as that of the field variable, obtained by the reported values in Table 2, are illustrated in Fig. 19.



(a)

(b)

Fig. 19. The variation of denominator of the (a) geometry, and (b) field variable.

Comparing these figures, we can see that there is no meaningful relationship between the variations of these two functions. Now, we attempt to solve the problem by using classic NURBS-based IGA, where the control weights are assigned identical values with the weights of the geometry, as well as the proposed *w*-adaptive IGA. To ensure that all integrations are calculated with machine precision, in both cases, we use a set of 15×15 quadrature points per knot-element.

In the case of IGA, we impose the essential boundary conditions by using classic least-square fitting. Note that the only remaining unknown control variables to be determined by analysis are (u_6, u_7). The obtained distribution of error of IGA solution is represented in Fig. 20. As observed, the error is of order 10^{-2} over most regions of the domain. The energy norm of this error distribution is $\|e_h\|_E = 1.62e-1$. One may attempt to perform *h*-refinement to achieve a better accuracy with the expected optimal convergence rate of $\mathcal{O}(2)$ in energy norm, studied in test case 1.

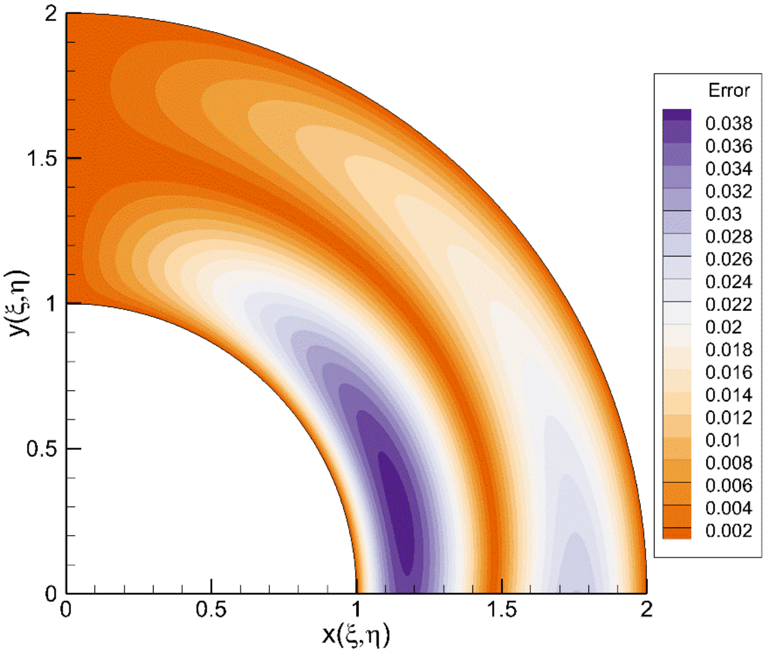


Fig. 20. Error distribution of NURBS-based IGA solution.

Next, we study the performance of *w*-adaptivity for solving the same problem. In this case, the essential boundary conditions are imposed exactly by using the analytical values for boundary control variables and control weights, reported in Table 2, prior to performing the adaptivity. The only remaining unknowns to be determined by analysis here are (u_6, w_6^u) and (u_7, w_7^u). We start the adaptivity process by the initial guess of assuming unit values for both unknown control weights, i.e. $\boldsymbol{\chi}_0 = \left\{ w_6^u, w_7^u \right\}_0 = \{1,1\}$. The initial distribution of error prior to performing the adaptivity is indicated in Fig. 21. The energy norm of error in this case is $\|e_h\|_E = 1.05e-1$, which is slightly better than the result of previous case with IGA due to exact satisfaction of boundary conditions on all edges here.

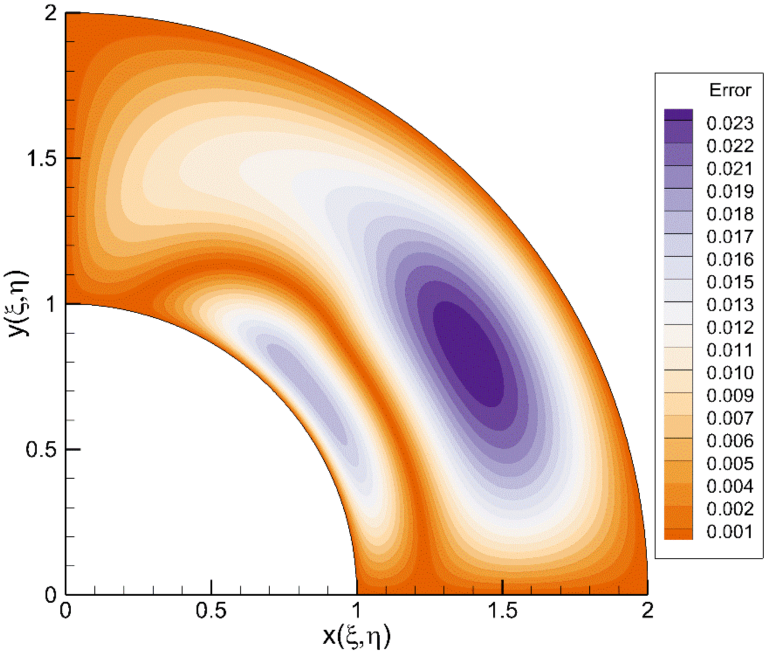


Fig. 21. Error distribution of the initial solution before performing *w*-adaptivity.

Starting with this solution, we now execute the adaptivity process driven by the estimator. The history of estimated error as well as the exact error (in energy norm) during adaptivity are plotted in Fig. 22. It is surprising to notice that although the adaptivity process has been guided by the estimator, the exact error has diminished to machine precision ($1.54e-13$) after 38 iterations, which implies the high reliability of the proposed adaptive framework.

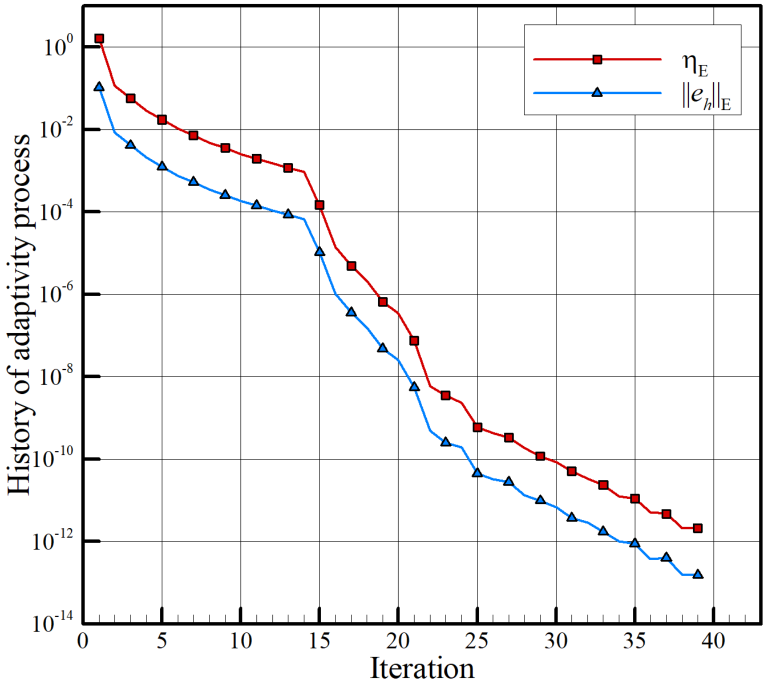


Fig. 22. The histories of estimated and exact error during *w*-adaptivity.

The obtained optimal values of unknown control variables after adaptivity are presented in Table 3, alongside their analytical values. Considering this table, we can see that the obtained optimal values are in agreements with the analytical ones within 12 digits after the decimal place.

Table 3. The obtained optimal control weights by *w*-adaptivity together with the analytical values.

Design variable	Initial	Optimal	Analytical
w_6^u	1.0	0.707106781185937	0.707106781186548
w_7^u	1.0	0.707106781186656	0.707106781186548

Finally, it must be mentioned here that inspired by Theorem 1, it might seem tempting that one would be able to achieve this accuracy simply by performing the analysis using a higher order basis. We emphasize here that although Theorem 1 establishes that GNURBS can always be transformed to higher order classic NURBS, obtaining these results by directly making use of NURBS-based IGA using any order of basis functions is not possible. The reader can consult [3] for more details on this apparent inconsistency.

4.4. Test Case 4: Patch test

As the last numerical experiment, we study the satisfaction of patch test by the proposed *w*-adaptive isogeometric method. For this purpose, we consider the quarter annulus from previous example with a perturbed mesh as shown in Fig. 23.

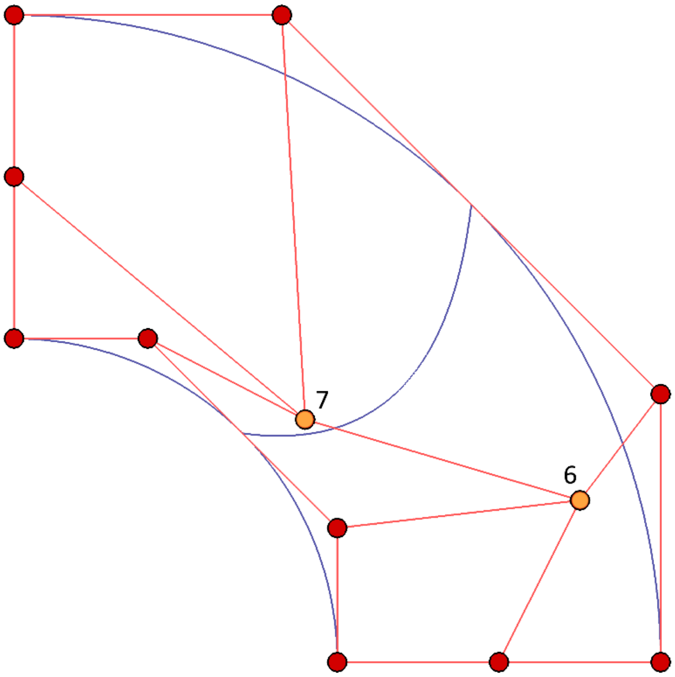


Fig. 23. The quarter ring with a perturbed mesh.

We consider two cases of the standard patch test, as well as an extended patch test in rational space discussed below. In both cases, we use 12×12 quadrature points per knot-element for integration.

4.4.1. Standard patch test

We recall here that NURBS-based IGA satisfies the standard patch test as reported by Hughes et al. in [1,2]. This is in fact one of the key advantages of using an isoparametric basis. We investigate here if this test will also be satisfied by *w*-adaptive IGA. Towards this end, we consider a Poisson equation with Dirichlet boundary conditions on all edges

$$\begin{aligned} -\Delta u &= f \\ u &= u_D \quad \text{on}\,\Gamma_D \end{aligned} \tag{38}$$

where f and u_D are specified by the following exact solution

$$u(x,y) = 2x + y + 1. \tag{39}$$

First, we assign the boundary control weights the same values with those of the geometry. This ensures satisfaction of all essential conditions exactly. The boundary control variables are obtained by using linear least square fitting and enforced strongly.

Next, similar to previous example, we execute *w*-adaptivity with the initial guess of $\boldsymbol{\chi}_0 = \left\{ w_6^u, w_7^u \right\}_0 = \{1,1\}$ for interior control weights. The history of adaptivity process is illustrated in Fig. 24. As observed, the error is diminished to machine precision ($3.26e-13$) after 31 iterations.

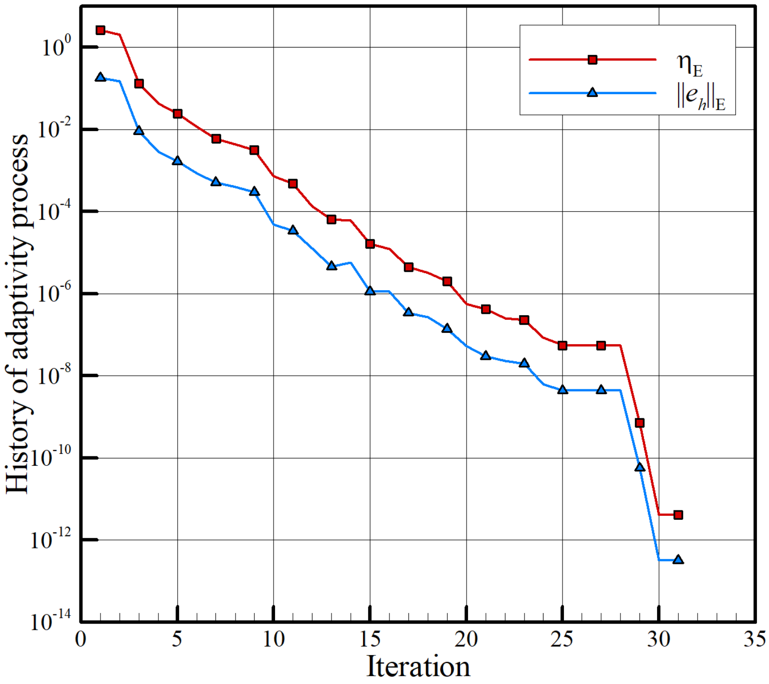


Fig. 24. History of *w*-adaptivity for standard patch test.

Moreover, the obtained optimal values for control weights are presented in Table 4. According to this table, the obtained optimal values are in agreements with the analytical values, i.e. w_6^G, w_7^G, within 12 digits after the decimal place.

Table 4. The obtained optimal control weights by *w*-adaptivity as well as the analytical values for standard patch test.

Design variable	Initial	Optimal	Analytical
w_6^u	1.0	0.853553390593643	0.853553390593274
w_7^u	1.0	0.853553390592845	0.853553390593274

The results of this experiment are consistent with the assertions of Hughes et al. [1] regarding the fact that affine covariance, which is ensured by isoparametric concept, is an essential property for satisfying patch tests.

4.4.2. Rational patch test

As the final numerical experiment, we repeat the previous test on a Poisson problem with the following exact solution

$$u(x,y)=\frac{2x+y}{x+y+1} \tag{40}$$

Dirichlet boundary conditions specified by the exact solution are assumed on all edges. The exact boundary control weights and control variables are obtained by solving Eq. (28) and enforced strongly. Note that in this case, due to rational structure of the exact solution, the optimal control weights, unlike previous case, have nothing to do with those of the geometry. Similar to previous case, we conduct *w*-adaptivity with the initial guess of $\boldsymbol{\chi}_0=\left\{w_6^u,w_7^u\right\}_0=\{1,1\}$ for interior control weights. The history of adaptivity process is depicted in Fig. 25. As observed, the exact error is diminished to machine precision ($1.49e-14$) after 36 iterations.

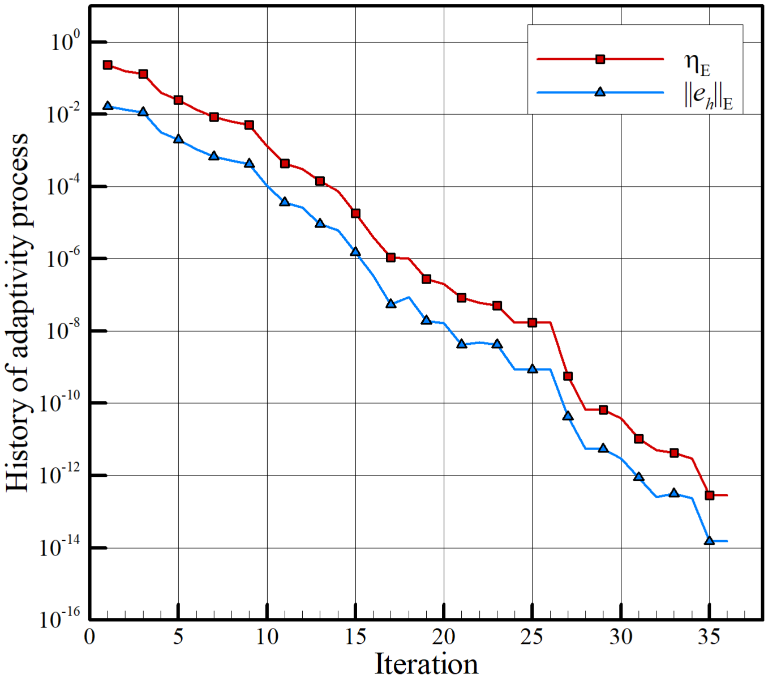


Fig. 25. History of *w*-adaptivity for rational patch test.

This study reveals another significant fact about using rational splines for analysis, that is, over an arbitrary parameterization, *w*-adaptive IGA is able to reproduce the exact solution of a problem whose closed-form solution is a first order rational expression. The results of this experiment also lead to interesting questions for more studies on correlation between isoparametric concept or affine-invariance and satisfaction of different patch tests. For instance, as mentioned earlier, Cottrell et al. [2] refer to affine-invariance, ensured by using an isoparametric basis, as an essential property for satisfying patch test. While this is consistent with our obtained results for the standard patch test, the devised experiment of rational patch test seems to be violating this necessity.

5. Further improvements

The present study opens doors to multiple new areas of research in IGA. While the numerical results are promising, there are many challenges which require further research to be addressed. We review a few of these aspects in this section in view of providing insights for interested readers.

Proof of convergence: While in this paper, we only numerically studied the rate of convergence of *w*-adaptive IGA, possible mathematical proofs of these results need to be developed. Specifically, in the case of second order elliptic PDEs with smooth solutions, the presented numerical results suggest that the NURBS basis of a particular degree with optimal weights has the same rate of convergence with the B-spline basis of one higher degree. To our knowledge, this has not been reported or proved in the literature. It also needs to be investigated if these results hold true for higher dimensions (3D) as well.

Adaptive Quadrature: One of the main aspects which needs to be more systematically addressed is devising efficient quadrature rules for integration over the arising highly-rational elements in *w*-adaptive IGA. In particular, two common strategies of devising a weighted quadrature rule [68], or employing an adaptive subdivision technique can be exploited for improved efficiency. Quadrature rules that restore variational consistency, or the so-called VCI domain integration methods which have been extended to IGA in [69], also provide an alternative promising technique for addressing this issue. An appropriate index, based on the total variation of denominator in Eq. (30) for instance, needs to be defined to identify elements where the quadrature needs to be improved.

Localization: Another aspect which can help improving the efficiency of the proposed adaptive method is the selection of the vector of design variables in Eq. (22). Although we only studied global *w*-adaptivity in this paper, as discussed earlier, this is unnecessary. For instance, it seems plausible to only consider the control weights of the elements in selective regions of the domain as design variables especially in the case of problems with irregularities such as test case 2. Robust and effective marking techniques, however, need to be developed and studied for any localization.

Stabilization: While the most promising scope of the proposed *w*-adaptive algorithm is perhaps on problems with sharp layers, instabilities may occur in many problems of this type with Galerkin approach [44]. Stabilization techniques, such as SUPG [70] need to be used in such scenarios. Incorporation of these techniques within the proposed *w*-adaptive framework, however, is non-trivial and needs further research.

Combination with other refinement techniques: Similar to other refinement techniques, the proposed *w*-refinement method can plausibly be combined with any of existing refinement methods for improved efficiency. Combination of this method with order elevation, in particular, requires employing an alternative generalization of NURBS, which is already introduced in [3] for performing optimal order elevation on parametric curves by the authors. In contrast to the introduced method in this paper, this combination will allow for introducing 'customized rationality' to the solution space which is expected to improve efficiency; please see [3] for more details.

Application to other rational splines: Finally, although we only studied here the application of the proposed *w*-adaptive method to GNURBS-based IGA, it seems plausible to all other variations of IGA based on rational forms of splines such as T-splines, U-splines etc.

6. Conclusion

A novel adaptivity technique in isogeometric analysis, referred to as adaptive *w*-refinement, was introduced. The proposed adaptive method allows for the approximation of solution with optimal rational basis by treating the control weights as additional degrees of freedoms. It was shown that this procedure effectively alleviates the deficiencies of NURBS for analysis and leads to superior results at the expense of solving an unconstrained optimization problem. The performance of the proposed method on elliptic problems with smooth and sharp solutions was studied. It was observed that *w*-adaptive IGA results in one order faster convergence than classic IGA in the case of smooth problems, while significant improvement of accuracy is achieved in problems with sharply varying solutions. Moreover, unlike classic IGA, the proposed adaptive method was demonstrated to be able to reproduce the exact solution of problems whose closed-form solutions lie in rational space, revealing a new critical aspect of using rational splines for analysis. Overall, the proposed adaptive *w*-refinement procedure provides a new effective technique for the enrichment of function space in isogeometric method and possibly a competitive tool with hierarchical splines.

Acknowledgements

The authors would like to thank the support of National Science Foundation through grant CMMI-1661597.

Appendix A. Proof of Theorem 1

We first review two lemmas on the multiplication of Bézier, as well as B-spline bivariate functions. The proofs of these lemmas can be found in [71]

Lemma 1:

Let $f_b(\xi,\eta)$ and $g_b(\xi,\eta)$ be two $(p, q)^{\text{th}}$-degree bivariate Bézier functions defined as

$$\begin{aligned} f_b(\xi,\eta) &= \sum_{i=0}^{p}\sum_{j=0}^{q} B_{i,p}(\xi)B_{j,q}(\eta)f_{ij} \\ g_b(\xi,\eta) &= \sum_{i=0}^{p}\sum_{j=0}^{q} B_{i,p}(\xi)B_{j,q}(\eta)g_{ij} \end{aligned} \tag{B.1}$$

Their product function $h_b(\xi,\eta)$ is a Bézier function of degree (2*p*, 2*q*) which can be computed as

$$h_b(\xi,\eta) = f_b(\xi,\eta)g_b(\xi,\eta) = \sum_{i=0}^{2p}\sum_{j=0}^{2q} B_{i,2p}(\xi)B_{j,2q}(\eta)H_{ij}^{b} \tag{B.2}$$

where the ordinates of the product Bézier function are

$$H_{ij}^{b} = \sum_{k=\max(0,i-p)}^{\min(p,i)} \sum_{l=\max(0,j-q)}^{\min(q,j)} \frac{\binom{p}{k}\binom{p}{i-k}}{\binom{2p}{i}} \frac{\binom{q}{l}\binom{q}{j-l}}{\binom{2q}{j}} f_{kl}\, g_{i-k,j-l} \tag{B.3}$$

Lemma 2:

Let $f(\xi,\eta)$ and $g(\xi,\eta)$ be two (p, q)th-degree bivariate B-spline functions defined as

$$\begin{aligned} f(\xi,\eta) &= \sum_{i=0}^{n_1}\sum_{j=0}^{n_2} N_{ij}^{p,q}(\xi,\eta)\, f_{ij} \\ g(\xi,\eta) &= \sum_{i=0}^{n_1}\sum_{j=0}^{n_2} N_{ij}^{p,q}(\xi,\eta)\, g_{ij} \end{aligned} \tag{B.4}$$

Their product function $h(\xi,\eta)$ is a B-spline function of degree (2*p*, 2*q*), that is

$$h(\xi,\eta) = f(\xi,\eta)g(\xi,\eta) = \sum_{i=0}^{n_1^h}\sum_{j=0}^{n_2^h} N_{ij}^{2p,2q}(\xi,\eta)H_{ij} \tag{B.5}$$

where H_{ij} are the ordinates of the product B-spline function.

Specific to this lemma, many algorithms have been proposed in the literature for evaluating the ordinates H_{ij}; see e.g. [71–74]. One can use, for instance, a straightforward algorithm proposed by Piegl and Tiller [75] including three steps of

- Performing Bézier extraction
- Computation of the product of Bézier functions
- Recomposition of the Bézier product functions into B-spline form using knot removal.

where the product of Bézier functions in the second step can be computed analytically employing Lemma 1. Further, the knot vector of $h(\xi)$ can be constructed as described in [75]. Alternatively, one may use a more advanced algorithm referred to as Sliding Windows Algorithm (SWA) recently proposed by Chen et al. [72].

Proof. The proof relies on the lemma that the summation of two NURBS surfaces is a higher order NURBS surface [75]. We rewrite Eq. (5) in Section 2.2 in the following form:

$$\begin{Bmatrix} \mathrm{x}(\xi,\eta) \\ y(\xi,\eta) \\ z(\xi,\eta) \end{Bmatrix} = \frac{\sum_{i=0}^{n_1}\sum_{j=0}^{n_2} N_{ij}^{p,q}(\xi,\eta)\, w_{ij}^{xy}}{\sum_{k=0}^{n_1}\sum_{l=0}^{n_2} N_{kl}^{p,q}(\xi,\eta)\, w_{kl}^{xy}} \begin{Bmatrix} x_{ij} \\ y_{ij} \\ 0 \end{Bmatrix} + \frac{\sum_{i=0}^{n_1}\sum_{j=0}^{n_2} N_{ij}^{p,q}(\xi,\eta)\, w_{ij}^{z}}{\sum_{k=0}^{n_1}\sum_{l=0}^{n_2} N_{kl}^{p,q}(\xi,\eta)\, w_{kl}^{z}} \begin{Bmatrix} 0 \\ 0 \\ z_{ij} \end{Bmatrix} \tag{B.6}$$

Extracting the common denominator yields

$$\begin{aligned}
x(\xi,\eta) &= \frac{\left(\sum_{i=0}^{n_1}\sum_{j=0}^{n_2} N_{ij}^{p,q}(\xi,\eta)\, w_{ij}^{xy} x_{ij}\right)\left(\sum_{k=0}^{n_1}\sum_{l=0}^{n_2} N_{kl}^{p,q}(\xi,\eta)\, w_{kl}^{z}\right)}{D(\xi,\eta)} \\
y(\xi,\eta) &= \frac{\left(\sum_{i=0}^{n_1}\sum_{j=0}^{n_2} N_{ij}^{p,q}(\xi,\eta)\, w_{ij}^{xy} y_{ij}\right)\left(\sum_{k=0}^{n_1}\sum_{l=0}^{n_2} N_{kl}^{p,q}(\xi,\eta)\, w_{kl}^{z}\right)}{D(\xi,\eta)} \\
z(\xi,\eta) &= \frac{\left(\sum_{i=0}^{n_1}\sum_{j=0}^{n_2} N_{ij}^{p,q}(\xi,\eta)\, w_{ij}^{z} z_{ij}\right)\left(\sum_{k=0}^{n_1}\sum_{l=0}^{n_2} N_{kl}^{p,q}(\xi,\eta)\, w_{kl}^{xy}\right)}{D(\xi,\eta)}
\end{aligned} \tag{B.7}$$

where

$$D(\xi,\eta) = \left(\sum_{k=0}^{n_1}\sum_{l=0}^{n_2} N_{kl}^{p,q}(\xi,\eta)\, w_{kl}^{xy}\right)\left(\sum_{k=0}^{n_1}\sum_{l=0}^{n_2} N_{kl}^{p,q}(\xi,\eta)\, w_{kl}^{z}\right) \tag{B.8}$$

As can be observed, evaluation of the above expressions involves performing the multiplication of bivariate B-spline functions. According to Lemma 2, all the product functions in (A.7) are B-spline functions of degree (2*p*, 2*q*). Therefore, we can obtain the equivalent higher order NURBS representation of (A.6) in the following form

$$\begin{Bmatrix} x(\xi,\eta) \\ y(\xi,\eta) \\ z(\xi,\eta) \end{Bmatrix} = \sum_{i=0}^{\hat{n}_1}\sum_{i=0}^{\hat{n}_2} R_{ij}^{2p,2q}(\xi,\eta) \begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} \tag{B.9}$$

where

$$R_{ij}^{2p,2q}(\xi,\eta) = \frac{N_{ij}^{2p,2q}(\xi,\eta) W_{ij}}{\sum_{k=0}^{\hat{n}_1}\sum_{l=0}^{\hat{n}_2} N_{kl}^{2p,2q}(\xi,\eta) W_{kl}} \tag{B.10}$$

in which $(X_{ij}, Y_{ij}, Z_{ij}, W_{ij})$ are the coordinates and weights of the $(\hat{n}_1+1)\times(\hat{n}_2+1)$ control points of the equivalent higher order NURBS surface, which can be obtained using any of the mentioned algorithms followed by Lemma 2. □

In the special case of Rational Bézier (R-Bézier) surfaces, one can obtain straightforward analytical expressions for the coefficients of the equivalent higher order R-Bézier surface in (A.9). For this case, Eqs. (A.9) and (A.10) can be written as

$$\begin{Bmatrix} x(\xi,\eta) \\ y(\xi,\eta) \\ z(\xi,\eta) \end{Bmatrix} = \sum_{i=0}^{2p}\sum_{j=0}^{2q} R_{ij}^{2p,2q}\left(\xi,\eta\right)\begin{Bmatrix} X_{ij} \\ Y_{ij} \\ Z_{ij} \end{Bmatrix} \tag{B.11}$$

where

$$R_{ij}^{2p,2q}\left(\xi,\eta\right) = \frac{B_{ij}^{2p,2q}\left(\xi,\eta\right)W_{ij}}{\sum_{k=0}^{2p}\sum_{l=0}^{2q} B_{kl}^{2p,2q}\left(\xi,\eta\right)W_{kl}} \tag{B.12}$$

Using relations (A.2) and (A.3) in Lemma 1, the coordinates and weights of control points in these equations can be obtained as

$$\begin{aligned} W_{ij} &= \sum_{k=\max(0,i-p)}^{\min(p,i)} \; \sum_{l=\max(0,j-q)}^{\min(q,j)} \lambda_{ik}^{p}\,\lambda_{jl}^{q}\, w_{i-k,j-l}^{xy}\, w_{kl}^{z} \\ X_{ij} &= \frac{1}{W_{ij}} \sum_{k=\max(0,i-p)}^{\min(p,i)} \; \sum_{l=\max(0,j-q)}^{\min(q,j)} \lambda_{ik}^{p}\,\lambda_{jl}^{q}\, w_{kl}^{xy}\, x_{kl} w_{i-k,j-l}^{z} \\ Y_{ij} &= \frac{1}{W_{ij}} \sum_{k=\max(0,i-p)}^{\min(p,i)} \; \sum_{l=\max(0,j-q)}^{\min(q,j)} \lambda_{ik}^{p}\,\lambda_{jl}^{q}\, w_{kl}^{xy}\, y_{kl} w_{i-k,j-l}^{z} \\ Z_{ij} &= \frac{1}{W_{ij}} \sum_{k=\max(0,i-p)}^{\min(p,i)} \; \sum_{l=\max(0,j-q)}^{\min(q,j)} \lambda_{ik}^{p}\,\lambda_{jl}^{q}\, w_{i-k,j-l}^{xy}\, z_{kl} w_{kl}^{z} \end{aligned} \tag{B.13}$$

where $\lambda_{ij}^{n} = \frac{\begin{pmatrix} n \\ j \end{pmatrix}\begin{pmatrix} n \\ i-j \end{pmatrix}}{\begin{pmatrix} 2n \\ i \end{pmatrix}}$.

Appendix B. Derivation of the second order derivatives of basis functions and field variable

We recall the following relations for the transformation of the first and second order derivatives of an arbitrary variable $\mathcal{A}$ between physical and parametric coordinates:

$$\begin{bmatrix} \dfrac{\partial \mathcal{A}}{\partial x} \\ \dfrac{\partial \mathcal{A}}{\partial y} \end{bmatrix} = \left(J\right)^{-1} \begin{bmatrix} \dfrac{\partial \mathcal{A}}{\partial \xi} \\ \dfrac{\partial \mathcal{A}}{\partial \eta} \end{bmatrix} \tag{C.1}$$

where the Jacobian matrix J is

$$J = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} \tag{C.2}$$

Also,

$$\begin{Bmatrix} \dfrac{\partial^2 \mathcal{A}}{\partial x^2} \\ \dfrac{\partial^2 \mathcal{A}}{\partial x \partial y} \\ \dfrac{\partial^2 \mathcal{A}}{\partial y^2} \end{Bmatrix} = \left(J^{II}\right)^{-1} \begin{Bmatrix} \dfrac{\partial^2 \mathcal{A}}{\partial \xi^2} - \dfrac{\partial \mathcal{A}}{\partial x} x_{,\xi\xi} - \dfrac{\partial \mathcal{A}}{\partial y} y_{,\xi\xi} \\ \dfrac{\partial^2 \mathcal{A}}{\partial \xi \, \partial \eta} - \dfrac{\partial \mathcal{A}}{\partial x} x_{,\xi\eta} - \dfrac{\partial \mathcal{A}}{\partial y} y_{,\xi\eta} \\ \dfrac{\partial^2 \mathcal{A}}{\partial \eta^2} - \dfrac{\partial \mathcal{A}}{\partial x} x_{,\eta\eta} - \dfrac{\partial \mathcal{A}}{\partial y} y_{,\eta\eta} \end{Bmatrix} \tag{C.3}$$

where

$$J^{II} = \begin{bmatrix} x_{,\xi}^2 & 2x_{,\xi} y_{,\xi} & y_{,\xi}^2 \\ x_{,\xi} x_{,\eta} & x_{,\xi} y_{,\eta} + x_{,\eta} y_{,\xi} & y_{,\xi} y_{,\eta} \\ x_{,\eta}^2 & 2x_{,\eta} y_{,\eta} & y_{,\eta}^2 \end{bmatrix}, \tag{C.4}$$

Note that Eqs. (B.1) and (B.3) can be directly used for the calculation of spatial derivatives of the basis functions as well as the field variable. We will later use the above expressions for sensitivity analysis.

Appendix C. Derivation of sensitivity expressions

For brevity, we first define the following simplified notations:

$$\begin{aligned} &\rho \in \{\xi, \eta\} \\ &W \equiv W^u(\xi,\eta) = \sum_{k=0}^{n_1} \sum_{l=0}^{n_2} N_{kl}^{p,q}(\xi,\eta) w_{kl}^u \\ &W_{,\rho} \equiv \frac{\partial W^u(\xi,\eta)}{\partial \rho} \\ &W_{,\rho\rho} \equiv \frac{\partial^2 W^u(\xi,\eta)}{\partial \rho^2} \\ &N_L \equiv N_{ij}(\xi,\eta) \\ &u^h \equiv u^h(\xi,\eta) \\ &w_L \equiv w_{ij}^u \end{aligned} \tag{D.1}$$

Having the above expressions, we can find the required sensitivities in (24) and (25) as described below. The sensitivities of the field variable simplify to

$$\frac{\partial u^h}{\partial w_L} = \frac{N_L}{W}\left(u^h - u_L\right) \tag{D.2}$$

Further, the sensitivities of the first order spatial derivatives of the field variable can be derived as

$$\begin{bmatrix} \dfrac{\partial u^h_{,x}}{\partial w_L} \\ \dfrac{\partial u^h_{,y}}{\partial w_L} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial^2 u^h}{\partial x\,\partial w_L} \\ \dfrac{\partial^2 u^h}{\partial y\,\partial w_L} \end{bmatrix} = (J)^{-1} \begin{bmatrix} \dfrac{\partial^2 u^h}{\partial \xi\,\partial w_L} \\ \dfrac{\partial^2 u^h}{\partial \eta\,\partial w_L} \end{bmatrix} \tag{D.3}$$

where

$$\frac{\partial^2 u^h}{\partial \rho\,\partial w_L} = \frac{1}{W}\left[\left(\frac{N_L}{W} W_{,\rho} - \frac{\partial N_L}{\partial \rho}\right)(u^h - u_L) - N_L \frac{\partial u^h}{\partial \rho}\right] \tag{D.4}$$

By having the above relations, we can evaluate the sensitivities of the second order spatial derivatives of the field variable as follows

$$\left\{\begin{matrix} \dfrac{\partial u^h_{,xx}}{\partial w_L} \\ \dfrac{\partial u^h_{,xy}}{\partial w_L} \\ \dfrac{\partial u^h_{,yy}}{\partial w_L} \end{matrix}\right\} = \left\{\begin{matrix} \dfrac{\partial^3 u^h}{\partial x^2 \partial w_L} \\ \dfrac{\partial^3 u^h}{\partial x \partial y\,\partial w_L} \\ \dfrac{\partial^3 u^h}{\partial y^2 \partial w_L} \end{matrix}\right\} = (J^{II})^{-1} \left\{\begin{matrix} \dfrac{\partial u^h_{,\xi\xi}}{\partial w_L} - \dfrac{\partial u^h_{,x}}{\partial w_L} x_{,\xi\xi} - \dfrac{\partial u^h_{,y}}{\partial w_L} y_{,\xi\xi} \\ \dfrac{\partial u^h_{,\xi\eta}}{\partial w_L} - \dfrac{\partial u^h_{,x}}{\partial w_L} x_{,\xi\eta} - \dfrac{\partial u^h_{,y}}{\partial w_L} y_{,\xi\eta} \\ \dfrac{\partial u^h_{,\eta\eta}}{\partial w_L} - \dfrac{\partial u^h_{,x}}{\partial w_L} x_{,\eta\eta} - \dfrac{\partial u^h_{,y}}{\partial w_L} y_{,\eta\eta} \end{matrix}\right\} \tag{D.5}$$

where the required sensitivities of the higher order parametric derivatives of the field variable can be computed using the following expressions

$$\frac{\partial u^h_{,\rho\rho}}{\partial w_L} = \frac{\partial^3 u^h}{\partial \rho^2 \partial w_L} = \frac{1}{W^2}\left\{\begin{array}{l} \left[2\left(\dfrac{\partial N_L}{\partial \rho} - \dfrac{N_L}{W} W_{,\rho}\right) W_{,\rho} + N_L W_{,\rho\rho} - W \dfrac{\partial^2 N_L}{\partial \rho^2}\right]\left(u^h - u_L\right) \\ +2\left(N_L W_{,\rho} - W \dfrac{\partial N_L}{\partial \rho}\right) \dfrac{\partial u^h}{\partial \rho} \end{array}\right\} - \frac{N_L}{W}\left(\frac{\partial^2 u^h}{\partial \rho^2}\right) \tag{D.6}$$

and

$$\frac{\partial u^{h}_{,\xi\eta}}{\partial w_L}=\frac{\partial^{3}u^{h}}{\partial\xi\,\partial\eta\,\partial w_L}=\frac{1}{W^{2}}\left\{\begin{array}{l}\left[\left(\frac{\partial N_L}{\partial\eta}-\frac{2N_L}{W}W_{,\eta}\right)W_{,\xi}+N_L W_{,\xi\eta}+W_{,\eta}\frac{\partial N_L}{\partial\xi}-W\frac{\partial^{2}N_L}{\partial\xi\partial\eta}\right]\left(u^{h}-u_L\right)+\\ \left[\left(N_L W_{,\xi}-W\frac{\partial N_L}{\partial\xi}\right)\frac{\partial u^{h}}{\partial\eta}-\left(N_L W_{,\eta}-W\frac{\partial N_L}{\partial\eta}\right)\frac{\partial u^{h}}{\partial\xi}\right]\end{array}\right\}$$

$$-\frac{N_L}{W}\left(\frac{\partial^{2}u^{h}}{\partial\xi\partial\eta}\right)$$

(D.7)

References

[1] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Eng. 194 (2005) 4135–4195. https://doi.org/10.1016/j.cma.2004.10.008.

[2] J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, John Wiley & Sons, 2009. https://doi.org/10.1002/9780470749081.

[3] A.H. Taheri, S. Abolghasemi, K. Suresh, Generalizations of non-uniform rational B-splines via decoupling of the weights: theory, software and applications, Eng. Comput. (2019). https://doi.org/10.1007/s00366-019-00799-w.

[4] D. Schillinger, E. Rank, An unfitted hp-adaptive finite element method based on hierarchical B-splines for interface problems of complex geometry, Comput. Methods Appl. Mech. Eng. 200 (2011) 3358–3380. https://doi.org/10.1016/j.cma.2011.08.002.

[5] D. Schillinger, L. Dedè, M.A. Scott, J.A. Evans, M.J. Borden, E. Rank, T.J.R. Hughes, An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces, Comput. Methods Appl. Mech. Eng. 249–252 (2012) 116–150. https://doi.org/10.1016/j.cma.2012.03.017.

[6] C. Giannelli, B. Jüttler, S.K. Kleiss, A. Mantzaflaris, B. Simeon, J. Špeh, THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis, Comput. Methods Appl. Mech. Eng. 299 (2016) 337–365. https://doi.org/10.1016/j.cma.2015.11.002.

[7] T. Kanduč, C. Giannelli, F. Pelosi, H. Speleers, Adaptive isogeometric analysis with hierarchical box splines, Comput. Methods Appl. Mech. Eng. 316 (2017) 817–838. https://doi.org/10.1016/j.cma.2016.09.046.

[8] E.J. Evans, M.A. Scott, X. Li, D.C. Thomas, Hierarchical T-splines: Analysis-suitability, Bézier extraction, and application as an adaptive basis for isogeometric analysis, Comput. Methods Appl. Mech. Eng. 284 (2015) 1–20. https://doi.org/10.1016/j.cma.2014.05.019.

[9] D. Schillinger, J.A. Evans, A. Reali, M.A. Scott, T.J.R. Hughes, Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations, Comput. Methods Appl. Mech. Eng. 267 (2013) 170–232. https://doi.org/10.1016/j.cma.2013.07.017.

[10] A. Vuong, C. Giannelli, B. Jüttler, B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis, Comput. Methods Appl. Mech. Eng. 200 (2011) 3554–3567. https://doi.org/10.1016/j.cma.2011.09.004.

[11] P. Wang, J. Xu, J. Deng, F. Chen, Adaptive isogeometric analysis using rational PHT-splines, Comput. Aided Des. 43 (2011) 1438–1448. https://doi.org/10.1016/j.cad.2011.08.026.

[12] B.S. Michael R. Dörfel, Bert Jüttler, Adaptive isogeometric analysis by local h-refinement with T-splines, Comput. Methods Appl. Mech. Eng. 199 (2010) 264–275. https://doi.org/10.1016/j.cma.2008.07.012.

[13] T. Lyche, C. Manni, H. Speleers, eds., Splines and PDEs: From Approximation Theory to Numerical Linear Algebra, Springer, Cetraro, Italy, 2017. https://doi.org/10.1007/978-3-319-94911-6.

[14] X. Li, F.L. Chen, H.M. Kang, J.S. Deng, A survey on the local refinable splines, Sci. China Math. 59 (2016) 617–644. https://doi.org/10.1007/s11425-015-5063-8.

[15] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, T.W. Sederberg, Isogeometric analysis using T-splines, Comput. Methods Appl. Mech. Eng. 199 (2010) 229–263. https://doi.org/10.1016/j.cma.2009.02.036.

[16] H. Casquero, X. Wei, D. Toshniwal, A. Li, T.J.R. Hughes, J. Kiendl, Y.J. Zhang, Seamless integration of design and Kirchhoff–Love shell analysis using analysis-suitable unstructured T-splines, Comput. Methods Appl. Mech. Eng. 360 (2020) 112765. https://doi.org/10.1016/j.cma.2019.112765.

[17] L. Liu, H. Casquero, H. Gomez, Y.J. Zhang, Hybrid-degree weighted T-splines and their application in isogeometric analysis, Comput. Fluids. 141 (2016) 42–53. https://doi.org/10.1016/j.compfluid.2016.03.020.

[18] H. Casquero, L. Liu, Y. Zhang, A. Reali, H. Gomez, Isogeometric collocation using analysis-suitable T-splines of arbitrary degree, Comput. Methods Appl. Mech. Eng. 301 (2016) 164–186. https://doi.org/10.1016/j.cma.2015.12.014.

[19] H. Casquero, L. Liu, Y. Zhang, A. Reali, J. Kiendl, H. Gomez, Arbitrary-degree T-splines for isogeometric analysis of fully nonlinear Kirchhoff–Love shells, CAD Comput. Aided Des. 82 (2017) 140–153. https://doi.org/10.1016/j.cad.2016.08.009.

[20] M.A. Scott, X. Li, T.W. Sederberg, T.J.R. Hughes, Local refinement of analysis-suitable T-splines, Comput. Methods Appl. Mech. Eng. 213–216 (2012) 206–222. https://doi.org/10.1016/j.cma.2011.11.022.

[21] W. Wang, Y. Zhang, L. Liu, T.J.R. Hughes, Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology, CAD Comput. Aided Des. 45 (2013) 351–360. https://doi.org/10.1016/j.cad.2012.10.018.

[22] L. Liu, Y. Zhang, T.J.R. Hughes, M.A. Scott, T.W. Sederberg, Volumetric T-spline construction using Boolean operations, Eng. Comput. 30 (2013) 425–439. https://doi.org/10.1007/s00366-013-0346-6.

[23] Y. Zhang, W. Wang, T.J.R. Hughes, Conformal solid T-spline construction from boundary T-spline representations, Comput. Mech. 51 (2013) 1051–1059. https://doi.org/10.1007/s00466-012-0787-6.

[24] Y. Zhang, W. Wang, T.J.R. Hughes, Solid T-spline construction from boundary representations for genus-zero geometry, Comput. Methods Appl. Mech. Eng. 249–252 (2012) 185–197. https://doi.org/10.1016/j.cma.2012.01.014.

[25] W. Wang, Y. Zhang, G. Xu, T.J.R. Hughes, Converting an unstructured quadrilateral/hexahedral mesh to a rational T-spline, Comput. Mech. 50 (2012) 65–84. https://doi.org/10.1007/s00466-011-0674-6.

[26] W. Wang, Y. Zhang, M.A. Scott, T.J.R. Hughes, Converting an unstructured quadrilateral mesh to a standard T-spline surface, Comput. Mech. 48 (2011) 477–498. https://doi.org/10.1007/s00466-011-0598-1.

[27] X. Wei, Y. Zhang, L. Liu, T.J.R. Hughes, Truncated T-splines: Fundamentals and methods, Comput. Methods Appl. Mech. Eng. 316 (2017) 349–372. https://doi.org/10.1016/j.cma.2016.07.020.

[28] X. Wei, Y. Zhang, T.J.R. Hughes, M.A. Scott, Truncated hierarchical Catmull-Clark subdivision with local refinement, Comput. Methods Appl. Mech. Eng. 291 (2015) 1–20. https://doi.org/10.1016/j.cma.2015.03.019.

[29] X. Wei, Y.J. Zhang, T.J.R. Hughes, M.A. Scott, Extended Truncated Hierarchical Catmull-Clark Subdivision, Comput. Methods Appl. Mech. Eng. 299 (2016) 316–336. https://doi.org/10.1016/j.cma.2015.10.024.

[30] X. Li, X. Wei, Y.J. Zhang, Hybrid non-uniform recursive subdivision with improved convergence rates, Comput. Methods Appl. Mech. Eng. 352 (2019) 606–624. https://doi.org/10.1016/j.cma.2019.04.036.

[31] K.A. Johannessen, T. Kvamsdal, T. Dokken, Isogeometric analysis using LR B-splines, Comput. Methods Appl. Mech. Eng. 269 (2014) 471–514. https://doi.org/10.1016/j.cma.2013.09.014.

[32] X. Wei, Y.J. Zhang, T.J.R. Hughes, Truncated hierarchical tricubic C0 spline construction on unstructured hexahedral meshes for isogeometric analysis applications, Comput. Math. with Appl. 74 (2017) 2203–2220. https://doi.org/10.1016/j.camwa.2017.07.043.

[33] X. Wei, Y.J. Zhang, D. Toshniwal, H. Speleers, X. Li, C. Manni, J.A. Evans, T.J.R. Hughes, Blended B-spline construction on unstructured quadrilateral and hexahedral meshes with optimal convergence rates in isogeometric analysis, Comput. Methods Appl. Mech. Eng. 341 (2018) 609–639. https://doi.org/10.1016/j.cma.2018.07.013.

[34] A. Qarariyah, F. Deng, T. Yang, Y. Liu, J. Deng, Isogeometric analysis on implicit domains using weighted extended PHT-splines, J. Comput. Appl. Math. 350 (2019) 353–371. https://doi.org/10.1016/j.cam.2018.10.012.

[35] M. Scott, U-splines for Unstructured IGA Meshes in LS-DYNA ®, (2018) 1–5.

[36] Y. Lai, Y.J. Zhang, L. Liu, X. Wei, E. Fang, J. Lua, Integrating CAD with Abaqus: A practical isogeometric analysis software platform for industrial applications, Comput. Math. with Appl. 74 (2017) 1648–1660. https://doi.org/10.1016/j.camwa.2017.03.032.

[37] E. Atroshchenko, S. Tomar, G. Xu, S.P.A. Bordas, Weakening the tight coupling between geometry and simulation in isogeometric analysis: From sub- and super-geometric analysis to Geometry-Independent Field approximaTion (GIFT), Int J Numer Methods Eng. 114 (2018) 1131–1159. https://doi.org/10.1002/nme.5778.

[38] U. Basappa, A. Rajagopal, J.N. Reddy, Adaptive Isogeometric Analysis Based on a Combined r-h Strategy, Int. J. Comput. Methods Eng. Sci. Mech. 2287 (2016). https://doi.org/10.1080/15502287.2016.1153171.

[39] A. Mirzakhani, B. Hassani, A. Ganjali, Adaptive analysis of three-dimensional structures using an isogeometric control net refinement approach, Acta Mech. 226 (2015) 3425–3449. https://doi.org/10.1007/s00707-015-1376-5.

[40] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Parameterization of computational domain in isogeometric analysis : Methods and comparison, Comput. Methods Appl. Mech. Eng. 200 (2011) 2021–2031. https://doi.org/10.1016/j.cma.2011.03.005.

[41] W. Ma, J.-P. Kruth, NURBS curve and surface fitting for reverse engineering, Int. J. Adv. Manuf. Technol. 14 (1998) 918–927. https://doi.org/10.1007/BF01179082.

[42] N. Carlson, NURBS Surface Fitting with Gauss-Newton, Lulea University of Technology, 2009.

[43] W. Ma, NURBS-based computer aided design modelling from measured points of physical models, Catholic University of Leuven, 1994.

[44] J. Donea, A. Huerta, Finite Element Methods for Flow Problems., John Wiley & Sons, 2003. https://doi.org/10.1002/0470013826.

[45] J.T.O. Mark Ainsworth, A Posteriori Error Estimation in Finite Element Analysis, Wiley-Interscience [John Wiley & Sons], New York, 2000.

[46] T. Chen, R. Mo, Z.W. Gong, Imposing Essential Boundary Conditions in Isogeometric Analysis with Nitsche's Method, Appl. Mech. Mater. 121–126 (2011) 2779–2783. https://doi.org/10.4028/www.scientific.net/AMM.121-126.2779.

[47] D. Wang, J. Xuan, An improved NURBS-based isogeometric analysis with enhanced treatment of essential boundary conditions, Comput. Methods Appl. Mech. Eng. 199 (2010) 2425–2436. https://doi.org/10.1016/j.cma.2010.03.032.

[48] A. Embar, J. Dolbow, I. Harari, Imposing Dirichlet boundary conditions with Nitsche's method and spline-based finite elements, Int. J. Numer. Methods Eng. 83 (2010) 877–898. https://doi.org/10.1002/nme.2863.

[49] S. Shojaee, E. Izadpenah, A. Haeri, Imposition of Essential Boundary Conditions in Isogeometric Analysis Using the Lagrange Multiplier Method, Int. J. Optim. Civ. Eng. 2 (2012) 247–271. http://ijoce.iust.ac.ir/browse.php?a_code=A-10-1-65&slc_lang=en&sid=1.

[50] M. Ruess, D. Schillinger, Y. Bazilevs, V. Varduhn, E. Rank, Weakly enforced essential boundary conditions for NURBS-embedded and trimmed NURBS geometries on the basis of the finite cell method, Int. J. Numer. Methods Eng. 95 (2013) 811–846. https://doi.org/10.1002/nme.4522.

[51] Y. Guo, M. Ruess, Weak Dirichlet boundary conditions for trimmed thin isogeometric shells, Comput. Math. with Appl. 70 (2015) 1425–1440. https://doi.org/10.1016/j.camwa.2015.06.012.

[52] S. Fernandez-Mendez, A. Huerta, Imposing essential boundary conditions in mesh-free methods, Comput. Methods Appl. Mech. Eng. 193 (2004) 1257–1275. https://doi.org/10.1016/j.cma.2003.12.019.

[53] J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, Isogeometric analysis, John Wiley & Sons, 2009.

[54] P. Costantini, C. Manni, F. Pelosi, M.L. Sampoli, Quasi-interpolation in isogeometric analysis based on generalized B-splines, Comput. Aided Geom. Des. 27 (2010) 656–668. https://doi.org/10.1016/j.cagd.2010.07.004.

[55] J. Lu, G. Yang, J. Ge, Blending NURBS and Lagrangian representations in isogeometric analysis, Comput. Methods Appl. Mech. Eng. 257 (2013) 117–125. https://doi.org/10.1016/j.cma.2013.01.012.

[56] V. Calo, Q. Deng, V. Puzyrev, Quadrature blending for isogeometric analysis, Procedia Comput. Sci. 108 (2017) 798–807. https://doi.org/10.1016/j.procs.2017.05.143.

[57] R.R. Hiemstra, F. Calabrò, D. Schillinger, T.J.R. Hughes, Optimal and reduced quadrature rules for tensor product and hierarchically refined splines in isogeometric analysis, 316 (2017) 966–1004. https://doi.org/10.1016/j.cma.2016.10.049.

[58] F. Calabrò, G. Sangalli, M. Tani, Fast formation of isogeometric Galerkin matrices by weighted quadrature, 316 (2017) 606–622. https://doi.org/10.1016/j.cma.2016.09.013.

[59] M. Barton, V.M. Calo, Optimal quadrature rules for odd-degree spline spaces and their application

to tensor-product-based isogeometric analysis, 305 (2016) 217–240. https://doi.org/10.1016/j.cma.2016.02.034.

[60] F. Auricchio, F. Calabrò, T.J.R. Hughes, A. Reali, G. Sangalli, A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis, Comput. Methods Appl. Mech. Eng. 249–252 (2012) 15–27. https://doi.org/10.1016/j.cma.2012.04.014.

[61] T.J.R. Hughes, A. Reali, G. Sangalli, Efficient quadrature for NURBS-based isogeometric analysis, Comput. Methods Appl. Mech. Eng. 199 (2010) 301–313. https://doi.org/10.1016/j.cma.2008.12.004.

[62] L. Piegl, W. Tiller, The NURBS book, Springer Science & Business Media, 2012.

[63] C. Carstensen, L. Demkowicz, J. Gopalakrishnan, Breaking spaces and forms for the DPG method and applications including Maxwell equations, Comput. Math. with Appl. 72 (2016) 494–522. https://doi.org/10.1016/j.camwa.2016.05.004.

[64] J. Chan, N. Heuer, T. Bui-Thanh, L. Demkowicz, A robust DPG method for convection-dominated diffusion problems II: Adjoint boundary conditions and mesh-dependent test norms, Comput. Math. with Appl. 67 (2014) 771–795. https://doi.org/10.1016/j.camwa.2013.06.010.

[65] L. Beirao da Veiga, D. Cho, G. Sangalli, Anisotropic NURBS approximation in isogeometric analysis, Comput. Methods Appl. Mech. Eng. 209–212 (2012) 1–11. https://doi.org/10.1016/j.cma.2011.10.016.

[66] PGI Visual FORTRAN Reference Guide, NVIDIA Corporation, Hillsboro, OR, 2018.

[67] Vanderplaats, DOT – Design Optimization Tools, Users Manual, Vanderplaats Research & Development, Inc., Colorado Springs, CO., 1999.

[68] W. Gautschi, The use of rational functions in numerical quadrature, J. Comput. Appl. Math. 133 (2001) 111–126. https://doi.org/10.1016/S0377-0427(00)00637-3.

[69] M. Hillman, J.S. Chen, Y. Bazilevs, Variationally consistent domain integration for isogeometric analysis, Comput. Methods Appl. Mech. Eng. 284 (2015) 521–540. https://doi.org/10.1016/j.cma.2014.10.004.

[70] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1982) 199–259.

[71] X. Che, G. Farin, Z. Gao, D. Hansford, The product of two B-spline functions, Adv. Mater. Res. 186 (2011) 445–448. https://doi.org/10.4028/www.scientific.net/AMR.186.445.

[72] X. Chen, R.F. Riesenfeld, E. Cohen, An algorithm for direct multiplication of B-splines, IEEE Trans. Autom. Sci. Eng. 6 (2009) 433–442. https://doi.org/10.1109/TASE.2009.2021327.

[73] E.T.Y. Lee, Computing a chain of blossoms, with application to products of splines, Comput. Aided Geom. Des. 11 (1994) 597–620.

[74] K. Mørken, Some identities for products and degree raising of splines, Constr. Approx. 7 (1991) 195–208. https://doi.org/10.1007/BF01888153.

[75] L. Piegl, W. Tiller, Symbolic operators for NURBS, Comput. Aided Des. 29 (1997) 361–368. https://doi.org/10.1016/S0010-4485(96)00074-7.