Generalizations of non-uniform rational B-splines via decoupling of the weights: Theory, software and applications

Alireza H. Taheri[1], Saeed Abolghasemi[2], Krishnan Suresh[1*]

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

[2] *School of Mechanical Engineering, Shahrood University of Technology, Shahrood, Iran.*

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

Abstract:

We introduce a new class of curves and surfaces by exploring multiple variations of Non-Uniform Rational B-Splines. These variations which are referred to as Generalized Non-Uniform Rational B-Splines (GNURBS) serve as an alternative interactive shape design tool, and provide improved approximation abilities in certain applications. GNURBS are obtained by decoupling the weights associated with control points along different physical coordinates. This unexplored idea brings the possibility of treating the weights as additional degrees of freedoms. It will be seen that this proposed concept effectively improves the capability of NURBS, and circumvents its deficiencies in special applications. Further, it is proven that these new representations are merely disguised forms of classic NURBS, guaranteeing a strong theoretical foundation, and facilitating their utilization. A few numerical examples are presented which demonstrate superior approximation results of GNURBS compared to NURBS in both cases of smooth and non-smooth fields. Finally, in order to better demonstrate the behavior and abilities of GNURBS in comparison to NURBS, an interactive MATLAB toolbox has been developed and introduced.

Keywords: NURBS, isoparametric, weights, decoupling, generalization.

1. Introduction

Non-Uniform Rational B-Splines (NURBS) are perhaps the most popular curve and surface representation method in Computer-Aided Design/Computer-Aided Manufacturing (CAD/CAM). They were first introduced in 1975 by Versprille [1] as rational extension of B-splines. NURBS form the backbone of CAD, and are considered the dominant technology for engineering design [2]; further, they have also been extensively used in several applications including isogeometric analysis (IGA) [3], NURBS-augmented finite element analysis [4], shape optimization [5, 6], topology optimization [7, 8], material modeling [9, 10], reverse engineering [11], G-code generation [12] etc.

Recent generalizations of NURBS-based technology include T-splines [13, 14] which constitute a superset of NURBS, and provide the local refinement properties by allowing for some unstructured-ness. An alternative generalization of NURBS, referred to as Generalized Hierarchical NURBS (H-NURBS), were introduced in 2008 by Chen et al. [15] by extending the idea of hierarchical B-splines to NURBS. Similar to T-splines, H-NURBS primarily bring the possibility of local refinement with tensor-product surfaces. A novel shape-adjustable generalized Bézier curve with multiple shape parameters has been recently proposed by Hu et al. [16], and its applications to surface modelling in engineering has been studied. Most recent class of splines which removes the limitations of T-splines are Unstructured-splines (U-splines) that have been developed by Scott [17].

Other generalizations of NURBS have also been suggested in the literature, even though these representations have not gained popularity. For instance, Wang et al. [18] propose a generalized NURBS curve and surface representation with the primary advantage of representing smooth surfaces with genus zero using only one surface patch. This also provides a new method to exactly generate conic curves and revolution surfaces. Further, it simplifies modelling local features such as creases and ruled patches.

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. This is achieved by the introduction of weights into the basis functions in a rational manner. The applications of this rational form, however, is not limited to precise construction of conics. According to the literature, there are other applications where the weights have been employed as additional degrees of freedom for improved flexibility.

For instance, the weights can be employed as additional design variables for interactive shape design so that one can utilize both control point movement, and weight modification to attain local shape control [19]. Many studies suggest employing the weights as additional design variables in data-fitting for better accuracy [11, 20]. Carlson [20] 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 is in fact one of the other main advantages of NURBS over B-splines. While smooth piecewise polynomials such as B-splines are poor in the approximation of rapidly varying data and discontinuities, employing rational functions is an effective tool for addressing this class of problems [20]. In order to avoid solving a non-linear optimization problem, Ma [11, 21] develops a two-step linear algorithm for data approximation using NURBS.

Despite being an effective technique for improving the performance of NURBS, there is a wide range of applications where treating the weights as extra design variables is either impossible or can be problematic. For instance, Dimas and Briassoulis [22] point out that a bad choice of weights in approximation can lead to poor curve/surface parameterization. Piegl [23] mentions that "*improper application of the weights can result in a very bad parameterization, which can destroy subsequent surface constructions*". Further, there are numerous applications where employing the weights as additional design variables is essentially impossible. We will discuss some of these applications in Section 4. The focus of this paper is to develop new generalizations of NURBS to primarily address this shortcoming. These proposed generalizations improve the performance of NURBS, and provide an alternative concept for removing these deficiencies of NURBS. It will be shown that, unlike T-splines, these generalizations are only variations of classic NURBS, and do not constitute a new superset of NURBS, making it easy to integrate and deploy them in modern CAD/CAM systems.

The remainder of this paper is organized as follows: in Sections 2 and 3, we introduce two different generalizations of NURBS, and develop their theoretical properties. We explore some of the applications of GNURBS in Section 4, and compare their performance against classic NURBS.

Further potential areas of applications and extensions of GNURBS are also discussed in this section. An interactive MATLAB toolbox for GNURBS is discussed in Section 5, and finally conclusions are drawn in Section 6.

2. Generalized NURBS Curves: a non-isoparametric approach

We recall that the equation of a NURBS curve is parametrically defined as

$$\mathbf{C}(\xi)=\sum_{i=0}^{n} R_{i,p}(\xi)\mathbf{P}_i\,, \qquad a\le\xi\le b \tag{1}$$

where $\mathbf{P}_i$ are a set of $n+1$ control points and $R_{i,p}(\xi)$ are the corresponding rational basis functions associated with i^{th} control point defined as

$$R_{i,p}(\xi)=\frac{N_{i,p}(\xi)\,w_i}{\sum_{j=0}^{n} N_{j,p}(\xi)\,w_j} \tag{2}$$

where w_i are the weights associated with control points, and $N_{i,p}(\xi)$ are the B-spline basis functions of degree p, defined on a set of non-decreasing real numbers $\boldsymbol{\Xi}=\{\xi_0,\xi_1,\ldots,\xi_{n+p}\}$ called knot vector. $N_{i,p}(\xi)$ is recursively defined as:

$$\begin{aligned} N_{i,0}(\xi)&=\begin{cases}1 & if \quad \xi_i\le\xi<\xi_{i+1}\\ 0 & \quad otherwise\end{cases}\\ N_{i,p}(\xi)&=\frac{\xi-\xi_i}{\xi_{i+p}-\xi_i}N_{i,p-1}(\xi)+\frac{\xi_{i+p+1}-\xi}{\xi_{i+p+1}-\xi_{i+1}}N_{i+1,p-1}(\xi)\end{aligned} \tag{3}$$

The NURBS curve in (1) is a vector equation which, assuming $\mathbf{P}_i=[x_i\;\;y_i\;\;z_i]^T$, could be written in the following expanded form in 3D space

$$\begin{Bmatrix} x(\xi)\\ y(\xi)\\ z(\xi)\end{Bmatrix}=\sum_{i=0}^{n}R_{i,p}(\xi)\begin{Bmatrix} x_i\\ y_i\\ z_i\end{Bmatrix} \tag{4}$$

Observe that NURBS curves 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, e.g. all different types of splines;

and ensures significant properties such as affine invariance and convex hull which are of interest in geometric modelling.

We introduce here the concept of Generalized Non-Uniform Rational B-Splines (GNURBS) by the extension of the above equation as follows

$$\begin{Bmatrix} x(\xi) \\ y(\xi) \\ z(\xi) \end{Bmatrix} = \sum_{i=0}^{n} \begin{Bmatrix} R_{i,p}^{x}(\xi)\, x_i \\ R_{i,p}^{y}(\xi)\, y_i \\ R_{i,p}^{z}(\xi)\, z_i \end{Bmatrix} \tag{5}$$

where $[R_{i,p}^{x}(\xi),\, R_{i,p}^{y}(\xi),\, R_{i,p}^{z}(\xi)]^T$ is now a vector set of basis functions which is defined as

$$\begin{bmatrix} R_{i,p}^{x}(\xi) \\ R_{i,p}^{y}(\xi) \\ R_{i,p}^{z}(\xi) \end{bmatrix} = \left[\frac{N_{i,p}(\xi)\, w_i^x}{\sum_{j=0}^{n} N_{j,p}(\xi)\, w_j^x},\ \frac{N_{i,p}(\xi)\, w_i^y}{\sum_{j=0}^{n} N_{j,p}(\xi)\, w_j^y},\ \frac{N_{i,p}(\xi)\, w_i^z}{\sum_{j=0}^{n} N_{j,p}(\xi)\, w_j^z} \right]^T \tag{6}$$

where $\left(w_i^x, w_i^y, w_i^z\right)$ is the set of coordinate-dependent weights associated with i^{th} control point. Denoting the vector set of basis functions in (6) by $\mathbf{R}_{i,p}(\xi) = [R_{i,p}^{x}(\xi),\, R_{i,p}^{y}(\xi),\, R_{i,p}^{z}(\xi)]^T$, the equation of a GNURBS curve can be written in the following compact form

$$\mathbf{C}(\xi) = \sum_{i=0}^{n} \mathbf{R}_{i,p}(\xi) \odot \mathbf{P}_i, \qquad a \le \xi \le b \tag{7}$$

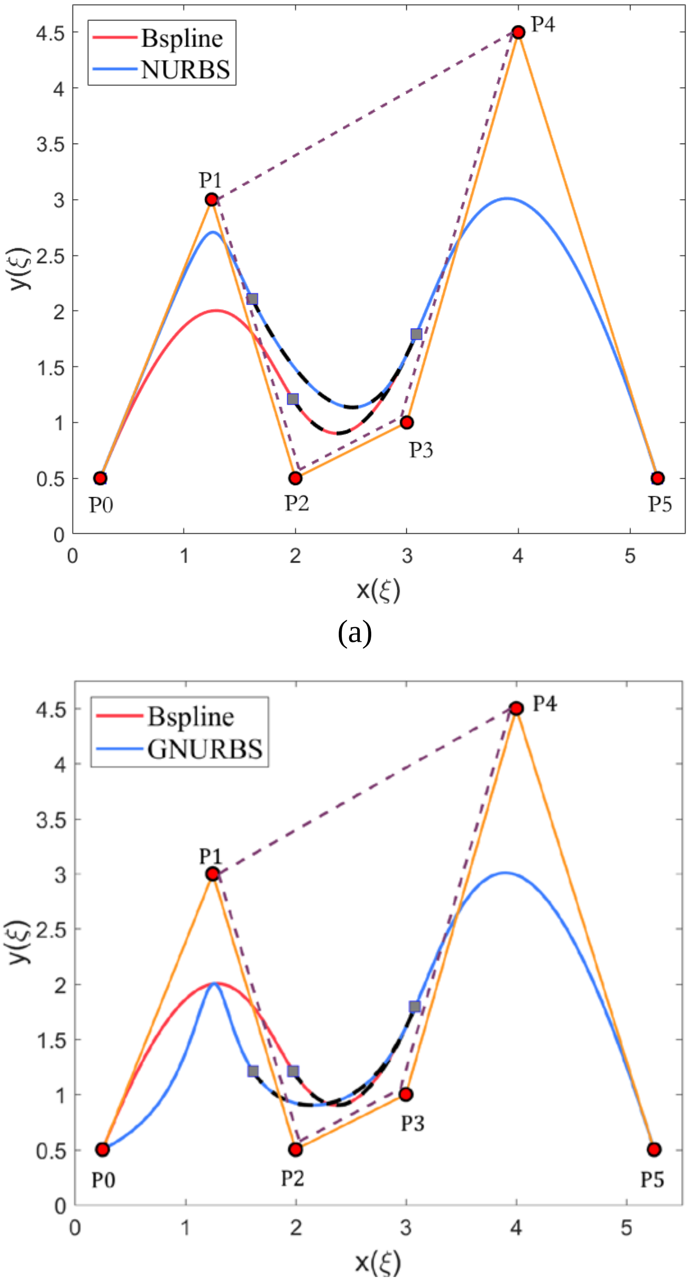
where $\odot$ denotes Hadamard (entry-wise) product of two vector variables.

Comparison of the above equation with that of classic NURBS 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 modification results in loss of properties such as strong convex hull and affine invariance. However, it will be established that GNURBS are only disguised forms of higher-order classic NURBS, i.e., all the properties of NURBS can be recovered through a suitable transformation, thus ensuring a strong theoretical foundation. In the following section, we develop the theory of GNURBS, and discuss how the properties of this non-isoparametric representation compare to those of NURBS.

2.1 Theory and properties

It can be easily shown that many properties of NURBS curves elaborated in [19] such as end-points interpolation, continuity, etc. are similarly satisfied in GNURBS. However, when treated in the direct form, some of the NURBS properties will be modified or even violated. We first discuss these, and later show how a simple transformation can be applied to recover all NURBS properties.

1. Affine invariance: Due to coordinate-dependence of the basis functions in GNURBS, applying an affine transformation directly to the control points will not result in the same curve as applying the same transformation to the curve; hence, this property is not satisfied.
2. Strong convex hull: A GNURBS curve need not lie in the convex hull of its control points. We demonstrate this graphically in Fig. 1 for a cubic curve ($p = 3$) constructed on the knot vector $\boldsymbol{\Xi} = \left\{\xi_0, \xi_1, ..., \xi_9\right\} = \left\{0,0,0,0,\tfrac{1}{3},\tfrac{2}{3},1,1,1,1\right\}$. Fig. 1(a) shows a B-spline curve and a NURBS curve with $\left\{w_0, ..., w_5\right\} = \left\{1,5,1,1,1,1\right\}$ constructed using the same control polygon. As observed, by increasing w_1 the middle knot span $\xi \in [\xi_4, \xi_5)$ always lies within the convex hull of control points {$\boldsymbol{P}_1$, $\boldsymbol{P}_2$, $\boldsymbol{P}_3$, $\boldsymbol{P}_4$}. Fig. 1(b) illustrates an example where the same knot span of a cubic GNURBS curve constructed with the same control polygon but a decoupled set of weights $\left\{w_0^x, ..., w_5^x\right\} = \left\{1,5,1,1,1,1\right\}$ and $\left\{w_0^y, ..., w_5^y\right\} = \left\{1,1,1,1,1,1\right\}$ exits the convex hull of its control points. However, we prove that it satisfies a weaker condition referred to as "*axis-aligned bounding box*" property described below.



(a)

(b)

Fig. 1. (a) A NURBS knot-span lies inside the *convex hull* of its control points. (b) A GNURBS knot-span need not lie inside the convex hull of its control points.

The function spaces corresponding to Fig. 1 are depicted in Fig. 2. Observe that the function space associated with the NURBS curve in Fig. 1(a) is identical for both *x* and *y* physical components, i.e. $R(\xi)$. Nevertheless, in the case of GNURBS curve shown in Fig. 1(b), the *x*-coordinate is constructed using the rational set of basis functions $R(\xi)$, while the *y*-coordinate is constructed using the set of B-spline basis functions $N(\xi)$.

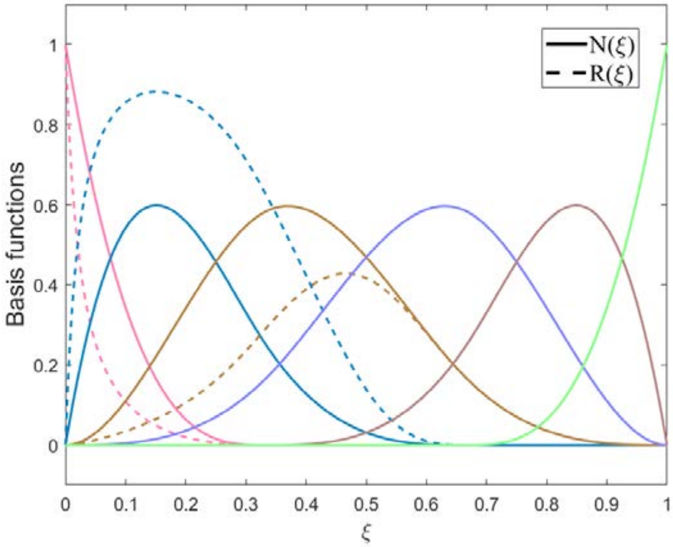


Fig. 2. Cubic function spaces corresponding to Fig. 1: B-spline function space $N(\xi)$, and NURBS function space $R(\xi)$ with $\left\{w_0,...,w_5\right\} = \left\{1,5,1,1,1,1\right\}$.

3. Axis-aligned bounding box (AABB): Every GNURBS knot span lies within the *axis-aligned bounding box* of its corresponding control points. That is, if $\xi \in \left[\xi_i, \xi_{i+1}\right)$, then $\mathbf{C}(\xi)$ lies within the bounding box of the control points $\left\{\mathbf{P}_{i-p},...,\mathbf{P}_i\right\}$.

   Proof:

   Eq. (5) can be easily written in the following form:

$$\begin{Bmatrix} x(\xi) \\ y(\xi) \\ z(\xi) \end{Bmatrix} = \sum_{i=0}^{n} R^{x}_{i,p}(\xi) \begin{Bmatrix} x_i \\ 0 \\ 0 \end{Bmatrix} + \sum_{i=0}^{n} R^{y}_{i,p}(\xi) \begin{Bmatrix} 0 \\ y_i \\ 0 \end{Bmatrix} + \sum_{i=0}^{n} R^{z}_{i,p}(\xi) \begin{Bmatrix} 0 \\ 0 \\ z_i \end{Bmatrix} \tag{8}$$

   Accordingly, Eq. (7) could be written as

$$\mathbf{C}(\xi) = \mathbf{C}_x(\xi) + \mathbf{C}_y(\xi) + \mathbf{C}_z(\xi), \qquad a \le \xi \le b \tag{9}$$

   where $\mathbf{C}_x(\xi)$, $\mathbf{C}_y(\xi)$ and $\mathbf{C}_z(\xi)$ are simply classic NURBS curves. From a geometric standpoint, each of these curves is the projection of the original non-isoparametric curve onto the corresponding physical axis. The following figure shows a graphical representation of above equations for a 2D cubic curve constructed over the knot vector $\mathbf{\Xi} = \left\{0,0,0,0,\tfrac{1}{3},\tfrac{2}{3},1,1,1,1\right\}$.

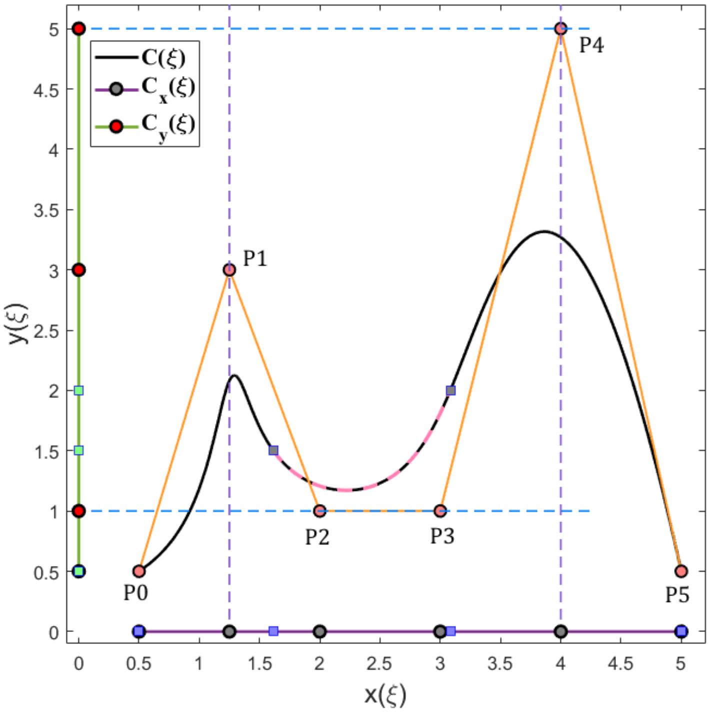


Fig. 3. Graphical representation of the bounding box property of a 2D cubic GNURBS curve with $\{w_0^x, ..., w_5^x\} = \{1,5,1,1,1,1\}$ and $\{w_0^y, ..., w_5^y\} = \{1,1,1,1,1,1\}$.

Since each of these curves is a classic NURBS curve, they satisfy the convex hull property. Therefore, the middle knot span of the curve which is marked in Fig. 3, must lie within the convex hulls of its corresponding control points on both projected curves. That is, if $\xi \in \left[\frac{1}{3}, \frac{2}{3} \right)$, then $\mathbf{C}_x(\xi)$ lies within the convex hull of the control points $\{x_1, ..., x_4\}$ which is the region between the two vertical lines in Fig. 3. Similarly, $\mathbf{C}_y(\xi)$ lies within the convex hull of the control points $\{y_1, ..., y_4\}$ which is the area between the two horizontal lines in this figure. Consequently, $\mathbf{C}(\xi)$ is contained in the intersection of these two convex hulls, which is the rectangular area shown in Fig. 3, referred to as the bounding box of $\{\mathbf{P}_1, ..., \mathbf{P}_4\}$.

4. <u>Local Modification</u>: Similar to NURBS, one can show that, in GNURBS, if a control point $\mathbf{P}_i$ is moved, or if any of the weights $w_i^d \, (d = x, y, z)$ is changed, it affects only the curve segment over the interval $[\xi_i, \xi_{i+p-1})$. However, unlike NURBS, changing the weights will only affect the parameterization of the curve along the corresponding physical coordinate

d , while the curve parameterization in the other directions will be preserved. This is, in fact, the key difference between GNURBS and NURBS which increases control. Assuming $\xi \in [\xi_i, \xi_{i+p-1})$, if w_i^d is increased (decreased), the curve will move closer to (farther from) $\mathbf{P}_i$. Further, for a fixed ξ, a point on $\mathbf{C}(\xi)$ moves along a horizontal (vertical) straight line as a weight w_i^x (w_i^y) is modified; see Fig. 1(b). This can be easily concluded from the proposed decomposition in (8) and the properties of classic NURBS curves.

5. Variation Diminishing Property: Due to loss of convex-hull property, this property is also not preserved in the direct form of GNURBS; that is, since the curve does not need to lie within the convex hull of its control points, there can be a plane (line in 2D) which intersects the curve multiple times without having any intersections with the control polygon.
6. NURBS Inclusion: If the weights in all directions are equal for each control point, then the GNURBS curve reduces to a NURBS curve.

Having discussed the properties of GNURBS in the direct form, we now develop a transformation of GNURBS into an equivalent NURBS of a higher order. Towards this end, we first review two lemmas on the multiplication of Bézier, as well as B-spline functions. The proofs of these lemmas can be found in [24].

Lemma 1:

Let $f_b(\xi)$ and $g_b(\xi)$ be two Bézier functions of degree *p* and *q*, respectively. Their product function $h_b(\xi)$ is a Bézier function of degree *p*+*q* which can be computed as [25]

$$h_b(\xi) = f_b(\xi) g_b(\xi) = \sum_{k=0}^{p+q} B_{k,p+q}\left(\xi\right) h_k^b \tag{10}$$

where $B_{k,p+q}\left(\xi\right)$ denotes k^{th} Bézier basis function of degree *p*+*q*, and

$$h_k^b = \sum_{j=\max(0,k-q)}^{\min(p,k)} \frac{\begin{pmatrix} p \\ j \end{pmatrix}\begin{pmatrix} q \\ k-j \end{pmatrix}}{\begin{pmatrix} p+q \\ k \end{pmatrix}} f_j g_{k-j} \tag{11}$$

End of Lemma 1

Lemma 2:

Let $f(\xi)$ and $g(\xi)$ be two univariate B-spline functions of degree *p* and *q*, respectively. Their product function $h(\xi)$ is a B-spline function of degree *p*+*q*, i.e.

$$h(\xi) = f(\xi)g(\xi) = \sum_{k=0}^{n_h} N_{k,p+q}(\xi)h_k \tag{12}$$

where h_k are the ordinates of the product B-spline function.

End of Lemma 2

Specific to Lemma 2, numerous algorithms have been proposed in the literature for evaluating the ordinates; see [26–29], for instance. In this paper, we will use a straightforward algorithm proposed by Piegl and Tiller [25] 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.

The product of Bézier functions in the second step can be computed analytically employing Lemma 1. Further, one can construct the knot vector of $h(\xi)$ as described in [25]. A more advanced algorithm referred to as Sliding Windows Algorithm (SWA) recently proposed by Chen et al. could be found in [27].

The decomposition in (8) together with the above two Lemmas lead to the following interesting theorem on the equivalence of NURBS and GNURBS.

Theorem: Every GNURBS curve of degree *p* and dimension m can be transformed exactly into a NURBS curve of degree $m \times p$.

Proof. We provide the proof here for a 2D curve, however, it can easily be extended to any higher dimension. The proof relies on the lemma that the summation of two NURBS curves is a higher order NURBS curve [25]. We rewrite Eq. (8) for a 2D curve in the following form:

$$\begin{Bmatrix} x(\xi) \\ y(\xi) \end{Bmatrix} = \frac{\sum_{i=0}^{n} N_{i,p}(\xi) w_i^x}{\sum_{j=0}^{n} N_{j,p}(\xi) w_j^x} \begin{Bmatrix} x_i \\ 0 \end{Bmatrix} + \frac{\sum_{i=0}^{n} N_{i,p}(\xi) w_i^y}{\sum_{j=0}^{n} N_{j,p}(\xi) w_j^y} \begin{Bmatrix} 0 \\ y_i \end{Bmatrix} \tag{13}$$

Extracting the common denominator leads to:

$$
\begin{aligned}
x(\xi) &= \frac{\left(\sum_{i=0}^{n} N_{i,p}\left(\xi\right) w_i^x x_i\right)\left(\sum_{j=0}^{n} N_{j,p}\left(\xi\right) w_j^y\right)}{\left(\sum_{j=0}^{n} N_{j,p}\left(\xi\right) w_j^x\right)\left(\sum_{j=0}^{n} N_{j,p}\left(\xi\right) w_j^y\right)} \\
y(\xi) &= \frac{\left(\sum_{i=0}^{n} N_{i,p}\left(\xi\right) w_i^y y_i\right)\left(\sum_{j=0}^{n} N_{j,p}\left(\xi\right) w_j^x\right)}{\left(\sum_{j=0}^{n} N_{j,p}\left(\xi\right) w_j^y\right)\left(\sum_{j=0}^{n} N_{j,p}\left(\xi\right) w_j^x\right)}
\end{aligned}
\tag{14}
$$

As can be observed, evaluation of (14) involves performing the multiplication of univariate B-spline functions. According to Lemma 2, the product functions in (14) are B-spline functions of degree 2*p*. Therefore, we can obtain the equivalent higher order NURBS representation of (13) in the following form

$$
\begin{Bmatrix} x\left(\xi\right) \\ y\left(\xi\right) \end{Bmatrix} = \sum_{i=0}^{\hat{n}} \widehat{R}_{i,2p} \begin{Bmatrix} X_i \\ Y_i \end{Bmatrix} \tag{15}
$$

where

$$
\widehat{R}_{i,2p} = \frac{N_{i,2p}\left(\xi\right) W_i}{\sum_{i=0}^{\hat{n}} N_{i,2p}\left(\xi\right) W_i} \tag{16}
$$

in which (X_i, Y_i, W_i) are the coordinates and weights of the equivalent higher order NURBS curve, which can be obtained using the algorithm described in Lemma 2, and $\hat{n}+1$ is the number of control points.

End of proof

In the special case of Rational Bézier (R-Bézier) curves, one can obtain straightforward analytical expressions for the coefficients of the equivalent higher order R-Bézier curve in (15). For this case, Eqs. (15) and (16) can be written as

$$
\begin{Bmatrix} x(\xi) \\ y(\xi) \end{Bmatrix} = \sum_{i=0}^{2p} \widehat{R}_{i,2p}\left(\xi\right) \begin{Bmatrix} X_i \\ Y_i \end{Bmatrix} \tag{17}
$$

where

$$\widehat{R}_{i,2p}=\frac{B_{i,2p}(\xi)W_i}{\sum_{i=0}^{\hat{n}}B_{i,2p}(\xi)W_i} \tag{18}$$

Using relations (10) and (11) in Lemma 1, the weights and control points in these equations are obtained as

$$\begin{aligned} W_i &= \sum_{j=\max(0,i-n)}^{\min(n,i)} \lambda_{ij} w_j^x w_{i-j}^y \\ X_i &= \frac{1}{W_i}\sum_{j=\max(0,i-n)}^{\min(n,i)} \lambda_{ij} x_j w_j^x w_{i-j}^y \\ Y_i &= \frac{1}{W_i}\sum_{j=\max(0,i-n)}^{\min(n,i)} \lambda_{ij} y_j w_j^y w_{i-j}^x \end{aligned} \tag{19}$$

where $\lambda_{ij}=\frac{\binom{n}{j}\binom{n}{i-j}}{\binom{2n}{i}}$.

Figure 4 shows a quadratic GNURBS curve, and its equivalent quartic NURBS curve obtained using the above theorem.

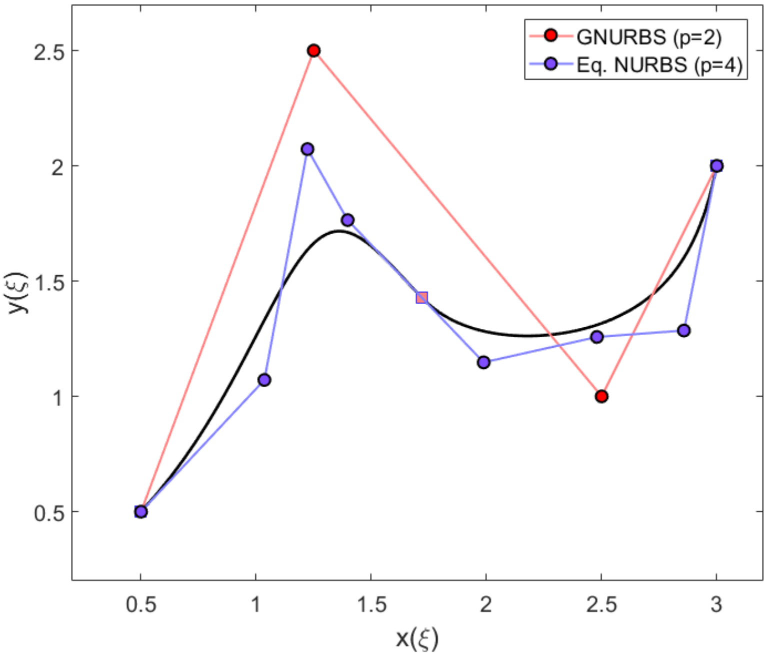


Fig. 4. Equivalence of a 2D quadratic GNRUBS curve with $\left\{w_0^x, ..., w_3^x\right\} = \{1, 2.5, 1.5, 3\}$ and $\left\{w_0^y, ..., w_3^y\right\} = \{1, 1, 2.5, 2\}$, with a quartic NURBS curve with $\left\{w_0, ..., w_7\right\} = \{1.00, 1.75, 2.30, 3.19, 3.81, 4.04, 5.25, 6.00\}$.

It needs to be pointed out that, despite the apparent violation of some properties of NURBS, the above theorem establishes that GNURBS are merely disguised form of higher order classic NURBS, thereby inheriting all the properties of NURBS indirectly. For instance, as can be seen in Fig. 4, the curve violates the global convex-hull of the original control polygon of GNURBS, however, it does lie within the convex-hull of the control polygon associated with its equivalent higher order classic NURBS.

2.2 Partial decoupling for 3D curves

One can easily extend the above theorem and formulation to 3D curves with independent weights along all three physical directions. However, a more practical case, which will be the emphasis for the rest of this paper, is to perform partial decoupling of the weights. In particular, in 3D, one can use the same set of weights in *x* and *y* directions, denoted by w^{xy}, and a different set of weights in *z* direction w^z. Accordingly, Eq. (5) could be written as

$$\begin{Bmatrix} x(\xi) \\ y(\xi) \\ z(\xi) \end{Bmatrix} = \sum_{i=0}^{n} \begin{Bmatrix} R_{i,p}^{xy}(\xi)\, x_i \\ R_{i,p}^{xy}(\xi)\, y_i \\ R_{i,p}^{z}(\xi)\, z_i \end{Bmatrix} \tag{20}$$

where

$$R_{i,p}^{xy}(\xi) = \frac{N_{i,p}(\xi)\, w_i^{xy}}{\sum_{j=0}^{n} N_{j,p}(\xi)\, w_j^{xy}} \tag{21}$$

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 curve in *x*-*y* plane. For better insight, we provide a graphical visualization of designing a 3D curve with an in-plane circular shape in Fig. 5.

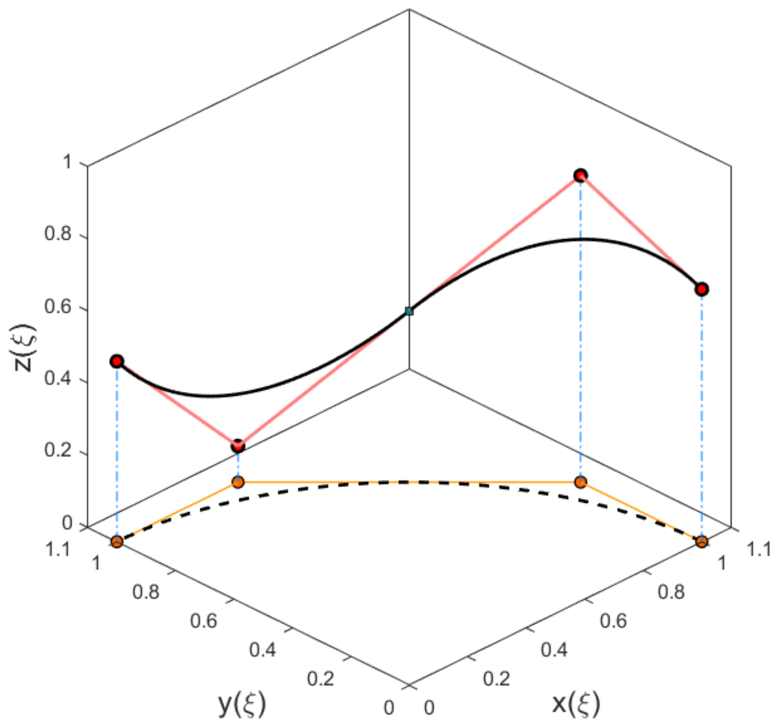


Fig. 5. A 3D GNURBS curve with an underlying precise circular arc: $\left\{w_0^{xy},\ldots,w_3^{xy}\right\}=\left\{1,0.8536,0.8536,1\right\}$ and $\left\{w_0^{z},\ldots,w_3^{z}\right\}=\left\{1,1,1,1\right\}$.

As can be clearly seen in Fig. 5, treating the independent set of out of plane weights can provide better flexibility and control. As a simple example, one can use this representation as an intermediate interactive shape design tool, and finally convert it to a higher order classic NURBS, if desired, to recover affine invariance and other properties. In this paper, we will focus on demonstrating superior approximation abilities of this representation in certain applications where a height function, field or set of data points need to be approximated over an underlying 2D curve.

To derive the equivalent higher order NURBS representation of (20), we rewrite this equation in the following form

$$\begin{Bmatrix} x(\xi) \\ y(\xi) \\ z(\xi) \end{Bmatrix} = \frac{\sum_{i=0}^{n} N_{i,p}(\xi) w_i^{xy}}{\sum_{j=0}^{n} N_{j,p}(\xi) w_j^{xy}} \begin{Bmatrix} x_i \\ y_i \\ 0 \end{Bmatrix} + \frac{\sum_{i=0}^{n} N_{i,p}(\xi) w_i^{z}}{\sum_{j=0}^{n} N_{j,p}(\xi) w_j^{z}} \begin{Bmatrix} 0 \\ 0 \\ z_i \end{Bmatrix} \tag{22}$$

Following a very similar procedure as for 2D curves, we can easily derive the expressions for the equivalent higher order NURBS curve to the generalized form in (20) as

$$\begin{Bmatrix} x(\xi) \\ y(\xi) \\ z(\xi) \end{Bmatrix} = \sum_{i=0}^{\hat{n}} \hat{R}_{i,2p} \begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} \tag{23}$$

where

$$\hat{R}_{i,2p} = \frac{N_{i,2p}(\xi) W_i}{\sum_{i=0}^{\hat{n}} N_{i,2p}(\xi) W_i} \tag{24}$$

(X_i, Y_i, Z_i, W_i) in these equations can be obtained using a similar algorithm as for 2D curves in the following form

$$W_i = \sum_{j=\max(0,i-n)}^{\min(n,i)} \lambda_{ij} w_j^{xy} w_{i-j}^{z} \tag{25}$$

and

$$\begin{aligned} X_i &= \frac{1}{W_i} \sum_{j=\max(0,i-n)}^{\min(n,i)} \lambda_{ij} x_j w_j^{xy} w_{i-j}^{z} \\ Y_i &= \frac{1}{W_i} \sum_{j=\max(0,i-n)}^{\min(n,i)} \lambda_{ij} y_j w_j^{xy} w_{i-j}^{z} \\ Z_i &= \frac{1}{W_i} \sum_{j=\max(0,i-n)}^{\min(n,i)} \lambda_{ij} z_j w_j^{z} w_{i-j}^{xy} \end{aligned} \tag{26}$$

where $\lambda_{ij} = \frac{\binom{n}{j}\binom{n}{i-j}}{\binom{2n}{i}}$.

It should be noted here that the properties of classic NURBS which are lost in this proposed generalization are not critical or even of interest in many applications of NURBS. Nevertheless, in some applications, these properties can be crucial. In order to make GNURBS applicable to such applications, we develop an alternative variation of NURBS which can be directly derived from the generalization proposed above.

3. Generalized NURBS curves: an isoparametric approach via order-elevation

Note that the equivalent higher order NURBS representation in (15) or (23) itself provides another variation of NURBS which can be directly employed as another alternative to NURBS with better flexibility in some applications.

In order to clarify how these equations provide additional flexibility than classic NURBS, we first derive a more generic form of these equations via an alternative approach using an extension of order elevation technique.

Assume a 2D R-Bézier curve of degree *p* is given as follows

$$\begin{Bmatrix} x(\xi) \\ y(\xi) \end{Bmatrix} = \frac{\sum_{i=0}^{p} B_{i,p}(\xi) w_i^{xy}}{\sum_{j=0}^{p} B_{j,p}(\xi) w_j^{xy}} \begin{Bmatrix} x_i \\ y_i \end{Bmatrix} \tag{27}$$

In order to elevate the degree of this curve by *q*, we can simply multiply both numerator and denominator of this equation by any arbitrary expression in the following form

$$f(\xi) = \sum_{i=0}^{q} B_{i,q}(\xi) w_i^z \tag{28}$$

Recalling Lemma 1, we can obtain the higher order R-Bézier curve with *q* degree elevations as

$$\begin{Bmatrix} x(\xi) \\ y(\xi) \end{Bmatrix} = \sum_{i=0}^{\hat{n}} \widehat{R}_{i,p+q} \begin{Bmatrix} X_i \\ Y_i \end{Bmatrix} \tag{29}$$

where

$$\widehat{R}_{i,p+q} = \frac{B_{i,p+q}(\xi) W_i}{\sum_{i=0}^{\hat{n}} B_{i,p+q}(\xi) W_i} \tag{30}$$

in which $\hat{n} = p + q$ and (X_i, Y_i, W_i) can be obtained using (31) and (32)

$$W_i = \sum_{j=\max(0,i-q)}^{\min(p,i)} \lambda_{ij} w_j^{xy} w_{i-j}^z \tag{31}$$

$$X_i = \frac{1}{W_i} \sum_{j=\max(0,i-q)}^{\min(p,i)} \lambda_{ij} x_j w_j^{xy} w_{i-j}^z$$
$$Y_i = \frac{1}{W_i} \sum_{j=\max(0,i-q)}^{\min(p,i)} \lambda_{ij} y_j w_j^{xy} w_{i-j}^z \tag{32}$$

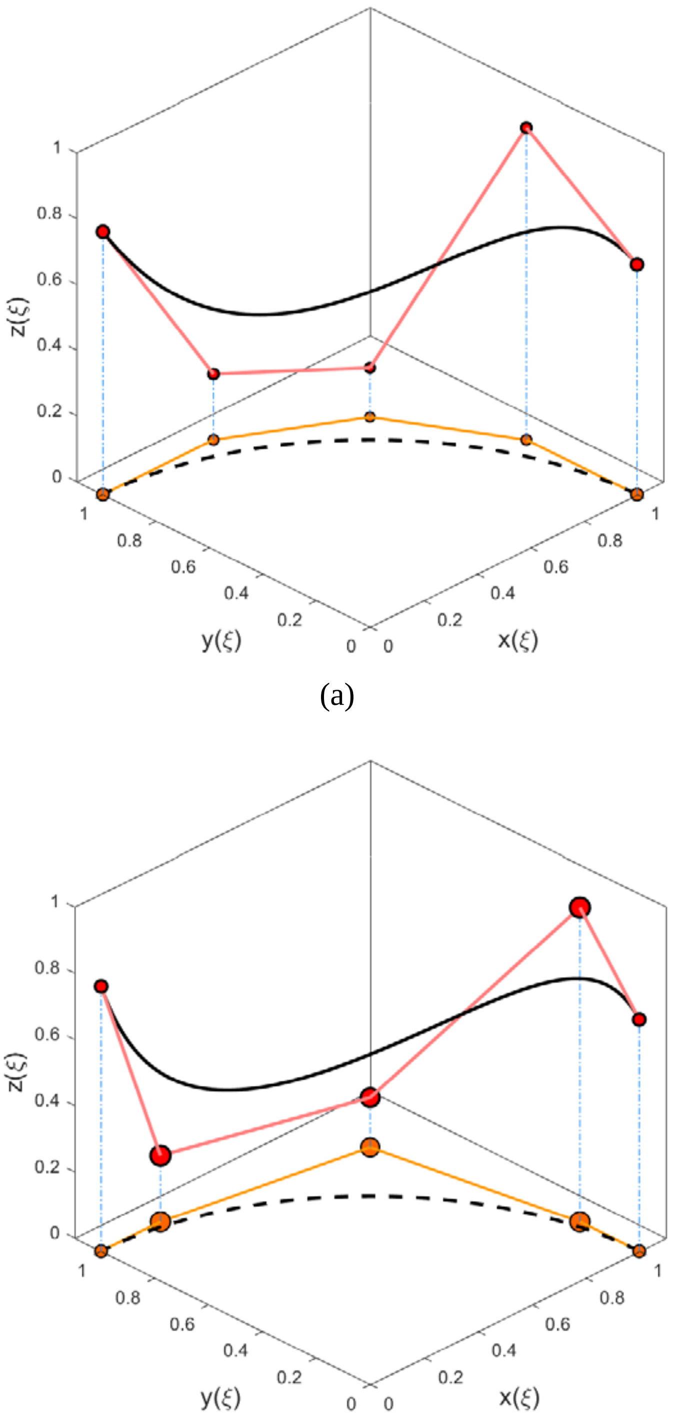
where $\lambda_{ij} = \dfrac{\binom{p}{j}\binom{q}{i-j}}{\binom{p+q}{i}}$.

Observe that this procedure can be seen as a trivial extension of the classic order elevation techniques in the literature [19, 30]. In fact, one can simply recover the common order elevation algorithm by assigning $w_i^z = 1, \forall i$ in (28). We will refer to this procedure as *generalized order elevation* hereafter. Now suppose we intend to add another dimension to the representation in (29) in an isoparametric manner. Again, this extra dimension can be viewed as the height function of a parametric curve in 2D, or may represent a field or set of data points which needs to be approximated over a 2D curve. For this purpose, we extend (29) as

$$\begin{Bmatrix} x(\xi) \\ y(\xi) \\ z(\xi) \end{Bmatrix} = \sum_{i=0}^{\hat{n}} \widehat{R}_{i,p+q} \begin{Bmatrix} X_i \\ Y_i \\ Z_i \end{Bmatrix} \tag{33}$$

It is interesting to notice that, although Eq. (33) apparently seems to be a classic R-Bézier curve, it provides additional flexibility. Observe that in the above procedure, w_i^z are arbitrary variables which can be freely chosen without perturbing the geometry or parameterization of the underlying curve in *x*-*y* plane.

In order to better demonstrate the effect of these weights on the behavior of GNURBS curves, we generate a 3D quartic GR-Bézier curve by performing the above process with $q = 2$ on a quadratic R-Bézier circular arc and assigning the heights of control points as shown in Fig. 6.



(a)

(b)

Fig. 6. A 3D isoparametric GNURBS curve with (a) $\left\{w_1^z, w_2^z, w_3^z\right\} = \{1,1,1\}$, and (b) $\left\{w_1^z, w_2^z, w_3^z\right\} = \{1, 2.5, 1\}$.

The obtained results with $\left\{w_1^z, w_2^z, w_3^z\right\} = \{1,1,1\}$ (classic order elevation) and $\left\{w_1^z, w_2^z, w_3^z\right\} = \{1, 2.5, 1\}$ are represented in Figs. 6(a) and (b), respectively. As observed, the heights of control points in both cases are identical. For more clarity, the size of control points is plotted proportional to their weights. Further, the corresponding sets of basis functions are plotted in Fig. 7.

Comparing Figs. 6(a) and (b), it can be noticed that by increasing w_2^z, the weights of the three interior control points are increased which results in out of plane deformation of the curve as depicted in Fig. 6(b). However, as this figure shows, this leads to automatic in-plane re-arrangement of control points in such a manner that the in-plane geometry of the curve (as well as its parameterization) remains unchanged.

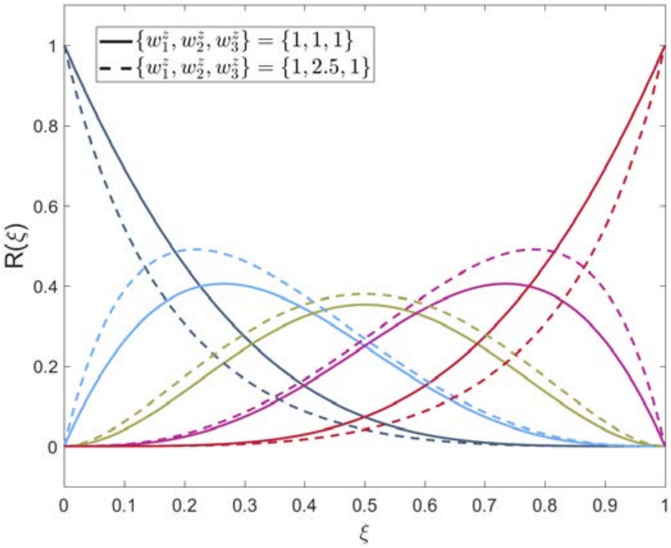


Fig. 7. The function spaces corresponding to GNURBS curves in Fig. 6.

The above algorithm can be extended to NURBS in a straightforward manner using a similar three step algorithm explained in Lemma 2. That is, Eq. (33) also holds true for NURBS with the rational basis functions defined as

$$\widehat{R}_{i,p+q} = \frac{N_{i,p+q}(\xi) W_i}{\sum_{i=0}^{\hat{n}} N_{i,p+q}(\xi) W_i} \tag{34}$$

We here note that while the variables w_i^z in (33) or (34) can be directly treated as design variables for improved flexibility, the physical meaning and local support of the weights in this variation are lost. Hence, it might not be suitable for being used as an interactive shape design tool. However, as will be shown in the next section, it can still be effectively employed as an enhanced tool for approximation purposes where the decision on the optimal values of the weights is made by a numerical algorithm.

4. Applications

The proposed generalizations of NURBS in (20) and (33) provide alternative tools to NURBS which can be useful in certain applications such as IGA. Exploring these advanced applications, however, is beyond the scope of this paper. In this section, we however investigate function approximations as an application. Hereafter, we will persistently refer to (20) as the first generalization of NURBS or non-isoparametric GNURBS, while we will refer to (33) as the second generalization of NURBS or isoparametric GNURBS.

Both these variations primarily provide the common and significant possibility of treating the out-of-plane weights as additional design variables, without perturbing the underlying geometry or its parameterization. However, the difference between them should be clear since the first form is obtained via explicit decoupling of the weights along different physical coordinates resulting in a non-isoparametric representation with the properties elaborated in Section 2, while the second variation is obtained by implicit decoupling of the weights within the isoparametric set of basis functions; thereby preserving the properties of NURBS. As discussed above, the generation of these implicitly decoupled set of weights in the second variation requires order elevation a priori.

Finally, we emphasize that although these new representations finally lie in the NURBS space, obtaining their results in certain class of applications by directly making use of NURBS does not seem possible.

4.1 Approximation over curved domains

There are various applications where the data or a function needs to be approximated over a parametric curved domain. For instance, there are numerous studies in the literature for the approximation of scattered data or functions on curved surfaces; see [31, 32] for a rigorous review. A similar problem arises in other applications such as modelling helical curves and surfaces [33–35], treating the non-homogenous essential boundary conditions in IGA [36–39] etc. In all these applications the limitation of preserving the underlying parameterization applies. Therefore, employing the weights as additional design variables is disallowed. In this section, we investigate the performance of GNURBS versus NURBS in this class of problems for two cases of approximating a smooth function as well as a rapidly varying one.

4.1.1 Least-square minimization using NURBS and GNURBS

Suppose an in-plane circular arc is given in the following parametric form

$$\mathbf{C}(\xi) = r\left(\cos(\frac{\pi}{2}\xi), \sin(\frac{\pi}{2}\xi)\right) \qquad 0 \le \xi \le 1 \tag{35}$$

where r is the radius of the circular arc. Eq. (35) can be precisely constructed using NURBS. Now, assume a height function $z(\xi)$ needs to be approximated over this arc with minimum error. This can be easily posed as a least-square approximation problem leading to optimal accuracy in *L2-norm*. Assuming $\left\{\xi_s \to (\overline{x}_s, \overline{y}_s, \overline{z}_s) : s \in \mathcal{S}\right\}$ is the set of ns collocation points, the error function f to be minimized is defined as

$$f = \frac{1}{2}\sum_{s\in\mathcal{S}} \left\|\hat{z}(\xi_s) - \overline{z}_s\right\|^2 = \frac{1}{2}\sum_{s\in\mathcal{S}} \left\| \sum_{L\in\mathcal{L}^s} R_L(\xi_s) z_L - \overline{z}_s \right\|^2 \tag{36}$$

where $\hat{z}(\xi)$ is the approximated NURBS function, ξ_s are the corresponding collocation points in the parametric space, $\mathcal{L}^s$ is the set of indices of non-zero basis functions at ξ_s and $\overline{z}_s = z(\xi_s)$.

In the case of NURBS, the only unknowns to consider are *control variables* z_L and the problem leads to a linear least square problem in the following matrix form

$$\sum_{s\in\mathcal{S}} \begin{pmatrix} R_0(\xi_s)R_0(\xi_s) & \cdots & R_0(\xi_s)R_n(\xi_s) \\ \vdots & \ddots & \vdots \\ R_n(\xi_s)R_0(\xi_s) & \cdots & R_n(\xi_s)R_n(\xi_s) \end{pmatrix} \begin{pmatrix} z_0 \\ \vdots \\ z_n \end{pmatrix} = \sum_{s\in\mathcal{S}} z(\xi_s) \begin{pmatrix} R_0(\xi_s) \\ \vdots \\ R_n(\xi_s) \end{pmatrix} \tag{37}$$

which can be solved for the $n+1$ unknowns $\boldsymbol{\lambda} = \left\{z_0, ..., z_n\right\}$ by proper choice of collocation points.

To improve the accuracy of approximation, invoking the proposed variations of NURBS, we can treat the out-of-plane weights w_i^z as extra design variables without perturbing the geometry or parameterization of the underlying precise circular arc. We may refer to these variables as *control weights* hereafter. With the first generalization in (20), the vector of design variables becomes $\boldsymbol{\lambda} = \left\{z_0, ..., z_n, w_0^z, ..., w_n^z\right\}$, where the positivity constraints on control weights ($w_i^z > 0, \forall i$) are often desired to be satisfied for numerical stability. Considering the new set of design variables, Eq. (37) now becomes a non-linear least-square problem which can be solved using any of the existing solvers such as Levenberg-Marquardt.

To avoid solving a non-linear problem, one can alternatively employ a two-step algorithm developed by Ma [11, 21], which 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. The development of this algorithm for GNURBS is provided below.

Employing the concept of homogeneous coordinates, the third component of GNURBS curve in (20) can be written in the following matrix form

$$z(\xi) = \frac{\mathbf{N}^T(\xi)\mathbf{z}^w}{\mathbf{N}^T(\xi)\mathbf{w}^z} \tag{38}$$

where the vector variables are defined as

$$\begin{aligned} \mathbf{N} &= [N_0(\xi), N_1(\xi), ..., N_n(\xi)]^T \\ \mathbf{z}^w &= [z_0^w, z_1^w, ..., z_n^w]^T = [z_0 w_0^z, z_1 w_1^z, ..., z_n w_n^z]^T \\ \mathbf{w}^z &= [w_0^z, w_1^z, ..., w_n^z]^T \end{aligned} \tag{39}$$

We may refer to z_i^w in this equation as weighted control variables. Also, we have dropped the subscript p in denoting the B-spline basis functions, for brevity. Eq. (38) can be written at the collocation points in the following form

$$\mathbf{N}^T(\xi_s)\mathbf{z}^w = z(\xi_s)\mathbf{N}^T(\xi_s)\mathbf{w}^z \qquad \forall s \in \mathcal{S} \tag{40}$$

Denoting the set of data points and B-spline basis functions in the matrix forms of (41) and (42), respectively

$$\bar{Z} = diag\left\{\bar{z}_1, ..., \bar{z}_{ns}\right\} \tag{41}$$

$$\tilde{N} = \begin{bmatrix} N_0(\xi_1) & N_1(\xi_1) & \cdots & N_n(\xi_1) \\ N_0(\xi_2) & N_1(\xi_2) & \cdots & N_n(\xi_2) \\ \vdots & \vdots & & \vdots \\ N_0(\xi_{ns}) & N_1(\xi_{ns}) & \cdots & N_n(\xi_{ns}) \end{bmatrix}_{ns\times(n+1)} \tag{42}$$

Eq. (40) can be written in the following compact form

$$\tilde{N}\,\mathbf{z}^w = \bar{Z}\,\tilde{N}\,\mathbf{w}^z \tag{43}$$

which can be re-written as

$$\tilde{\mathrm{A}}\begin{bmatrix}\mathbf{z}^{w}\\ \mathbf{w}^{z}\end{bmatrix}=\left[0\right]_{2n\times1} \tag{44}$$

where

$$\tilde{\mathrm{A}}=\begin{bmatrix}\tilde{N} & -\overline{Z}\,\tilde{N}\end{bmatrix}_{ns\times2n} \tag{45}$$

Eq. (44) is an over-determined system of equations and now represents a linear least-square problem. Multiplying the sides of this equation by $\tilde{\mathrm{A}}^{T}$ yields

$$\begin{bmatrix}\tilde{N}^{T}\tilde{N} & -\tilde{N}^{T}\overline{Z}\,\tilde{N}\\ -\tilde{N}^{T}\overline{Z}\,\tilde{N} & \tilde{N}^{T}\overline{Z}^{2}\tilde{N}\end{bmatrix}\begin{bmatrix}\mathbf{z}^{w}\\ \mathbf{w}^{z}\end{bmatrix}=\left[0\right]_{2n\times1} \tag{46}$$

It is possible to separate the control weights from the control variables by eliminating the lower left element of (46), which yields

$$\begin{bmatrix}\tilde{N}^{T}\tilde{N} & -\tilde{N}^{T}\overline{Z}\,\tilde{N}\\ 0 & M\end{bmatrix}\begin{bmatrix}\mathbf{z}^{w}\\ \mathbf{w}^{z}\end{bmatrix}=\left[0\right]_{2n\times1} \tag{47}$$

where

$$M=\tilde{N}^{T}\overline{Z}^{2}\tilde{N}-(\tilde{N}^{T}\overline{Z}\,\tilde{N})(\tilde{N}^{T}\tilde{N})^{-1}(\tilde{N}^{T}\overline{Z}\,\tilde{N}) \tag{48}$$

According to (47), the control weights are now decoupled from the control variables and can be obtained via solving the following homogeneous system of equations

$$M\,\mathbf{w}^{z}=\left[0\right]_{n\times1} \tag{49}$$

Further details on different algorithms for solving (49) and extracting the optimal real or positive weights can be found in [21]. Once the unknown weights are found, the optimal control variables can be subsequently obtained via solving (43).

With the second generalization in (33), however, the development of a linear algorithm does not seem easily possible. Therefore, a non-linear least square algorithm needs to be used to find the optimal set of design variables. Further, since the derivation of analytical Jacobian matrix becomes complicated in case of having internal knots, we limit our study to GR-Bézier. The vector of design variables for this simplified case becomes $\boldsymbol{\lambda}=\left\{Z_{0},...,Z_{n},w_{0}^{z},...,w_{q}^{z}\right\}$ where $n=p+q$. The imposition of the least square problem is quite straightforward; hence, we do not present it here. The derivation of Jacobian matrix components with respect to control weights, however, is non-trivial and requires evaluating the sensitivity using the following expressions

$$\frac{\partial W_i}{\partial w_k^z} = \begin{cases} \dfrac{\binom{p}{i-k}\binom{q}{k}}{\binom{p+q}{i}} w_{i-k}^{xy}, & if \quad (i-p) \le k \le i \\ 0 & Otherwise \end{cases} \tag{50}$$

The initial conditions for solving the least square problem are specified as follows

$$\boldsymbol{\lambda}_0 = \left\{ \underbrace{0, 0, \ldots, 0}_{n+1}, \underbrace{1, 1, \ldots, 1}_{q+1} \right\} \tag{51}$$

As previously discussed, by changing w_i^z during the optimization process, the in-plane coordinates of control points also vary at each iteration. However, since the in-plane geometry and parameterization are always fixed, one may only re-evaluate and update these coordinates after the termination of the optimization process according to the obtained optimal set of isoparametric basis functions. It is important to note that *this algorithm yields the combination of optimal weights and the corresponding arrangement of control points which results in the best approximation over a given parameterization*. To our knowledge, no such investigation has been reported in the literature thus far.

In the next section, we approximate various height functions over the circular arc in (35) modelled precisely with NURBS. In all cases, the interpolating end control points are prescribed to lie on the height function. Further, we employ 100 uniformly distributed sample points in the parametric space for setting up the least square problem. The numerical implementations are performed in MATLAB. Finally, the relative *L_2-norms* of the error are calculated using the following relation

$$error = \frac{\left(\int \left(\hat{z}(\xi) - z(\xi) \right)^2 d\Gamma \right)^{1/2}}{\int z(\xi) d\Gamma} \tag{52}$$

where the numerical integrations are calculated using Gaussian quadrature.

4.1.2 A smooth function: helix modelling

As the first numerical example, we consider approximating a smooth height function as

$$z(\xi) = b\varphi = b\left(\frac{\pi}{2}\xi \right) \tag{53}$$

over the parametric curve in (35). In the above equation, φ is the center angle of the circular arc in *x*-*y* plane and *b* is a constant. Eq. (53) together with (35) represent a segment of a helical curve, shown in Fig. 8 for $b=1$, and is a classic problem in geometric modelling. We here demonstrate how the proposed variations of NURBS can be useful for improved modelling of such type of problems.

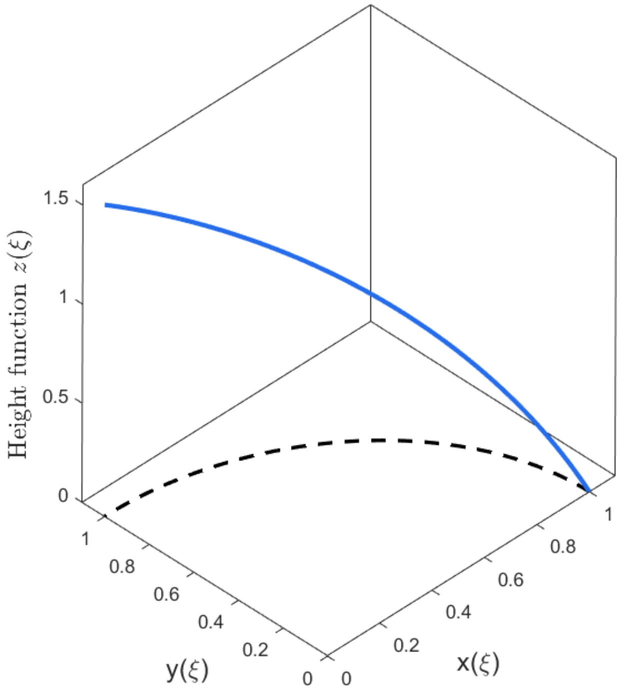


Fig. 8. A smooth helical curve.

Helical curves and surfaces do not have an exact representation in terms of polynomials or rational polynomials [40]. A high accuracy of approximation by NURBS using the minimal number of control points is of interest, and will make the helix more convenient to use in current CAD/CAM systems [34]. There is a large number of studies in the literature addressing this problem using R-Bézier, NURBS or other parametric representations; see e.g. [33–35, 41] for a review of these studies. Having examined these studies, it can be found that there are several considerations for a suitable approximation of helix such as the accuracy of normal angle, curvature, torsion and height, besides meeting certain geometric conditions at the end points of each segment [34]. However, we only focus here on approximating the height function with maximum accuracy, for simplicity. Further, it is desirable that the fitting curve precisely lies on the cylinder surface of the helix [35].

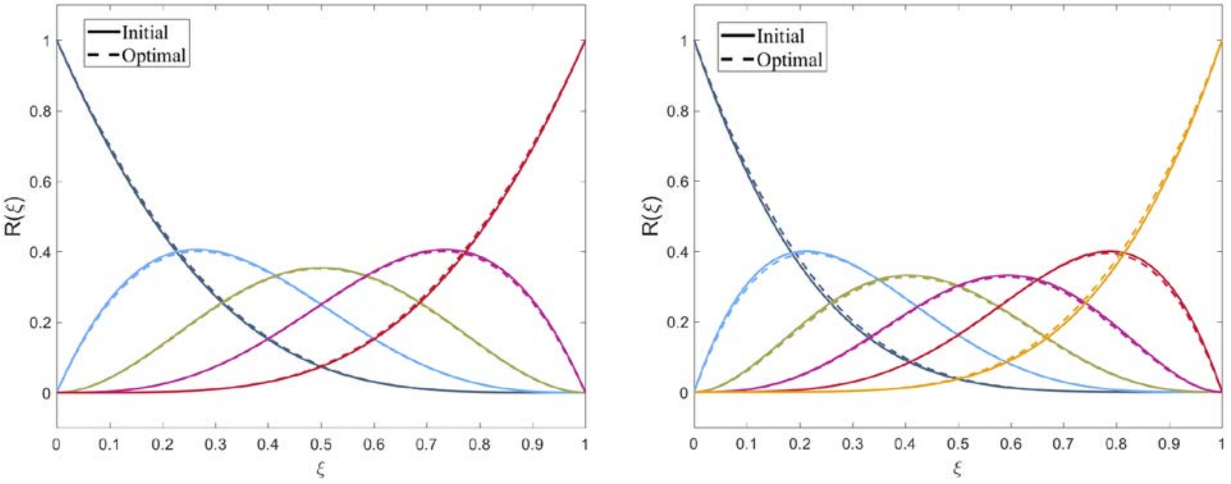
Since this is a geometric modelling problem, the properties of NURBS are important to be preserved for this particular application. Therefore, it is an ideal candidate for employing the second variation, i.e. isoparametric GR-Bézier, as the obtained optimal design is directly in the NURBS space. The obtained results using the above-discussed algorithm for different degrees of basis functions are presented in Table 1 for comparison.

Table 1. Error of approximating the helix height function using R-Bézier versus GR-Bézier in relative *L*$_2$-norm.

Curve type	Degree $(n = p + q)$	No. of control variables	No. of control weights	Error	Error ratio
R-Bézier	2	3	0	2.41E-2	1.0
2nd GR-Bézier			0	2.41E-2	
R-Bézier	3	4	0	1.50E-4	1.0
2nd GR-Bézier			2	1.50E-4	
R-Bézier	4	5	0	1.50E-4	121.9
2nd GR-Bézier			3	1.23E-6	
R-Bézier	5	6	0	2.30E-6	209.1
2nd GR-Bézier			4	1.10E-8	

As the table shows, the accuracy of approximation by GR-Bézier over R-Bézier increasingly improves by elevating the degree, as a larger number of control weights are added to the design space. In case of $p = 3$, however, no improvement in the accuracy is gained. This implies that the optimal values of the control weights for this case are equal to 1; that is, cubic R-Bézier obtained via order elevation is coincidentally optimal for the approximation of this height function.

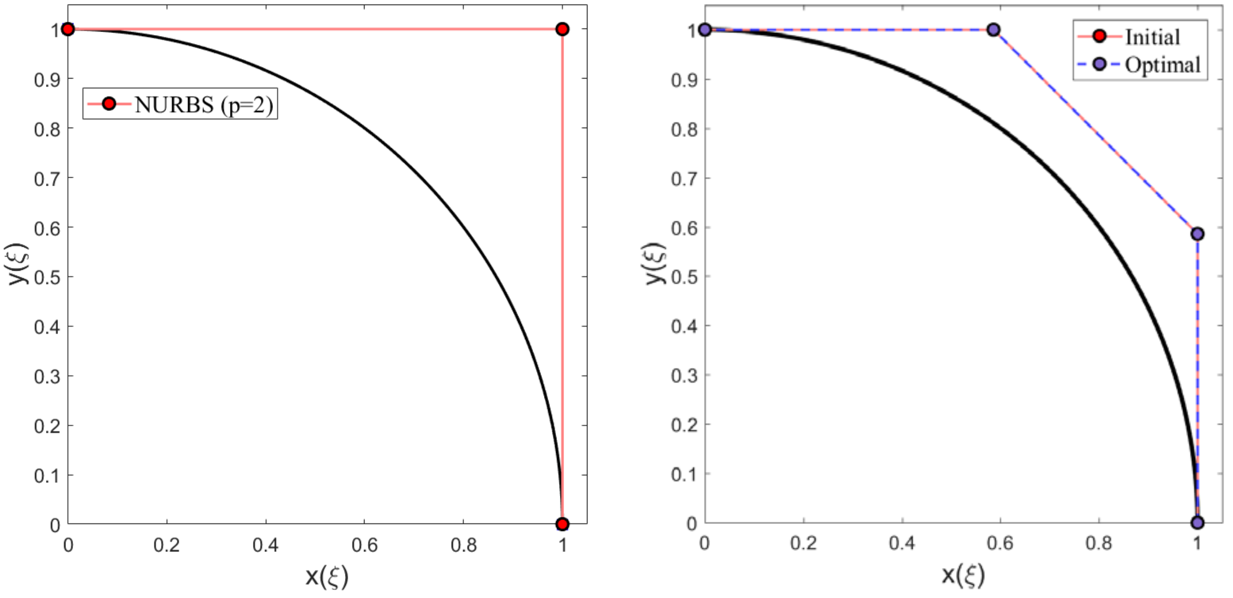
The initial and optimal sets of basis functions for approximation with different degrees are represented in Fig. 9. As can be observed in this figure, in both cases, the optimal sets of basis functions are only slightly different than the initial ones, however, this small deviation results in dramatic improvement of the accuracy of approximation as reported in Table 1.



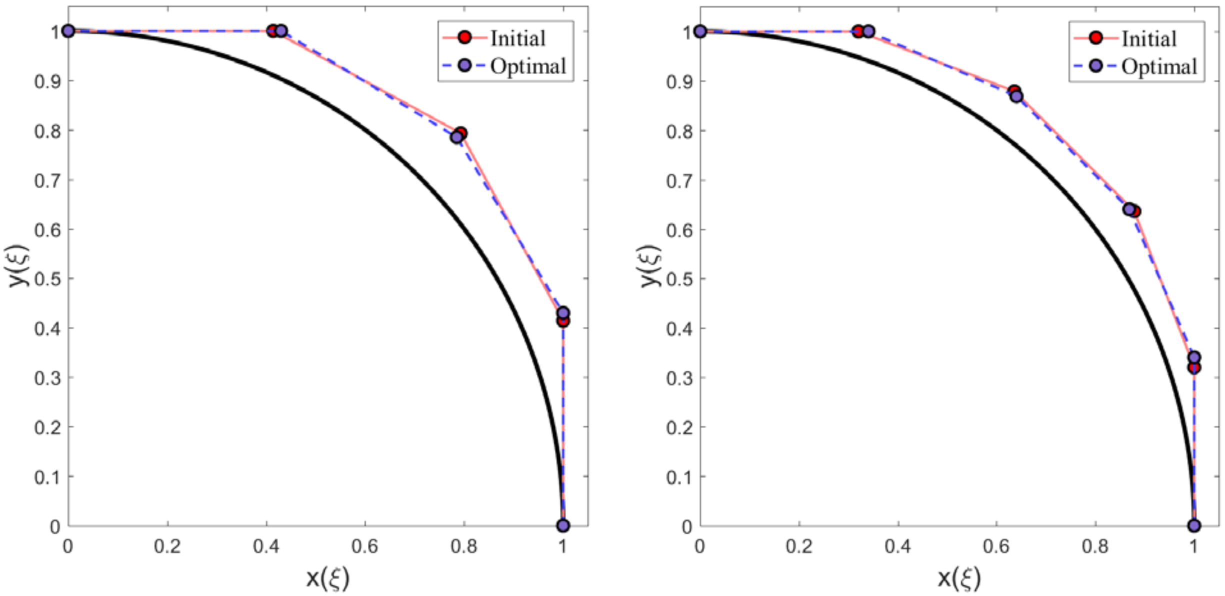
(a) (b)

Fig. 9. Initial and optimal basis functions for approximating the helix height function using 2nd GR-Bézier with degree (a) $n = 4$ and (b) $n = 5$.

We remind that in the case of isoparametric generalization (2nd GR-Bézier), the basis functions are identical along all physical coordinates. As previously explained, this leads to automatic re-arrangement of the in-plane coordinates of control points, depicted in Fig. 10, in such a manner that the in-plane geometry and its parameterization remain unchanged.



(a) (b)



(c) (d)

Fig. 10. Initial and optimal control nets for approximating the helix height function with (a) R-Bézier of degree $n = 2$, and 2nd GR-Bézier of degree (b) $n = 3$ (c) $n = 4$ and (d) $n = 5$.

We also investigate the performance of GNURBS compared to NURBS with respect to refining the knot sequence. For this experiment, we use the first variation (non-isoparametric), for simplicity and as it provides better flexibility. The obtained results for $p = 2$ are represented in Fig. 11.

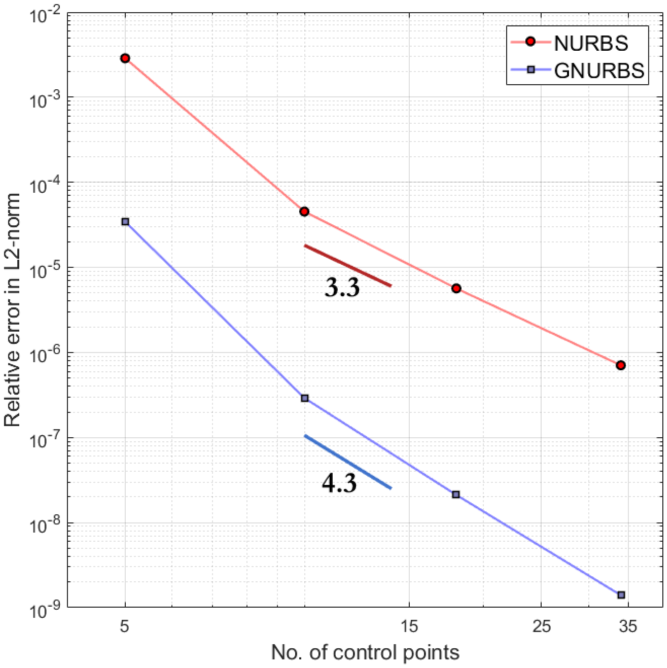


Fig. 11. Convergence rate of 1st GNURBS versus NURBS for approximating the helix height function.

As the figure shows, by including the control weights to the design space, the convergence rate is improved from 3.3 to 4.3, resulting in dramatic improvement in the accuracy especially when larger numbers of control points are employed. However, as previously mentioned, in the case of GNURBS there is an extra computational cost for obtaining the optimal weights via solving an additional homogenous system of equations.

4.1.3 A rapidly varying function

As the second example, we investigate the performance of the proposed variations of NURBS in capturing rapidly varying functions. We consider the problem of approximating a rapidly varying function as in (54) over the same circular arc

$$z(\xi) = \varphi\left(1 + e^{-\alpha(\varphi-0.5)^2} + e^{-\alpha(\varphi-0.8)^2}\right),\ \varphi = (\frac{\pi}{2})\xi \qquad (54)$$

which is plotted in Fig. 12 for $\alpha = 20$.

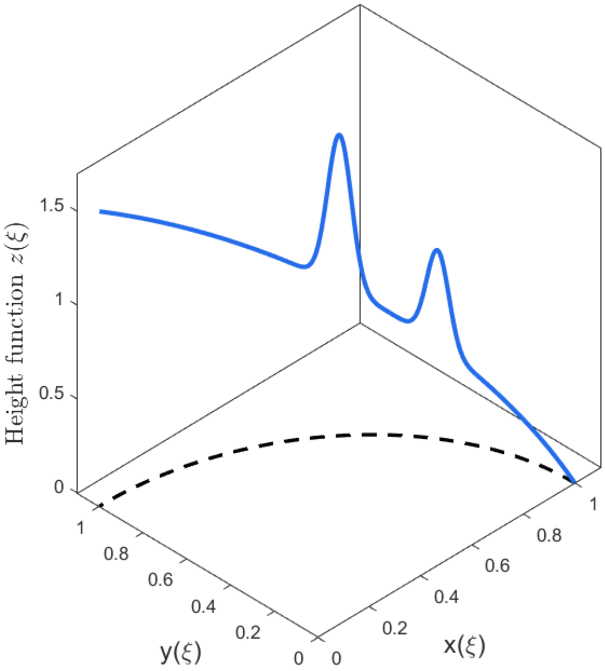


Fig. 12. A rapidly varying function over a circular arc.

Employing the first proposed variation of NURBS, we approximate the height function using different degrees of basis functions. The obtained results are presented in Table 2. All these models are obtained by performing uniform knot insertion over an initial R-Bézier arc and therefore possess maximal continuity.

Table 2. Error of approximating the rapidly varying function in (54) using NURBS versus 1st GNURBS in relative L_2-norm.

Curve type	Degree (p)	No. of control variables	No. of control weights	Error	Error ratio
NURBS	2	18	0	6.86E-2	9.23
1st GNURBS			18	7.43E-3	
NURBS	3	19	0	5.35E-2	9.80
1st GNURBS			19	5.46E-3	
NURBS	4	20	0	6.27E-2	14.31
1st GNURBS			20	4.38E-3	
NURBS	5	21	0	5.48E-2	40.60
1st GNURBS			21	1.35E-3	

According to the table, the accuracy of approximation using NURBS does not change noticeably by elevating the degree. On the other hand, the obtained results with GNURBS persistently improve by elevating the degree, which reveals the superiority of approximation of GNURBS over NURBS in capturing rapidly varying fields.

The approximation results for $p = 5$ are plotted in Fig. 13. The figure clearly shows the improvement of approximation in the case of GNURBS especially in the vicinity of existing sharp transitions in the field.

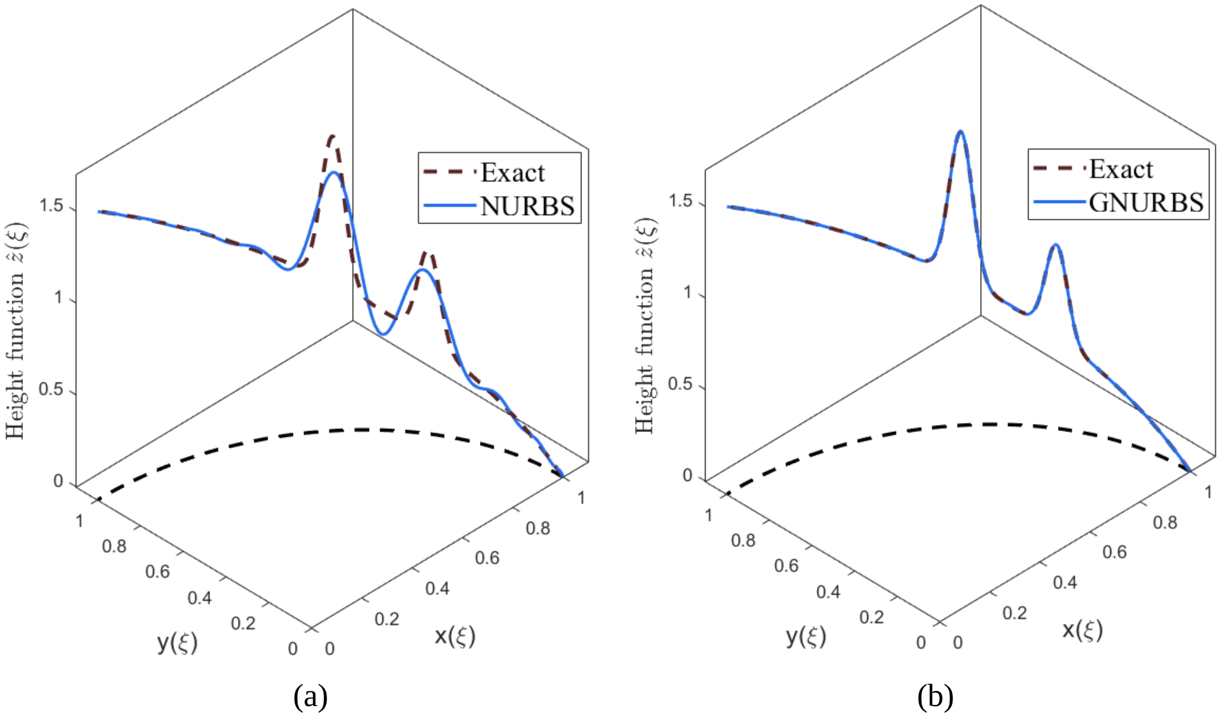


Fig. 13. Approximation of the rapidly varying function with quintic (a) NURBS and (b) 1st GNURBS.

Further, the corresponding basis functions are represented in Fig. 14. It is interesting to note that, unlike the previous case of approximating a smooth function, there is a significant change between the initial and optimal basis functions. As can be seen, this difference is more substantial for the basis functions effecting the behavior of the curve around the existing sharp local gradients, implying that the corresponding weights tend to take the extreme values in these regions.

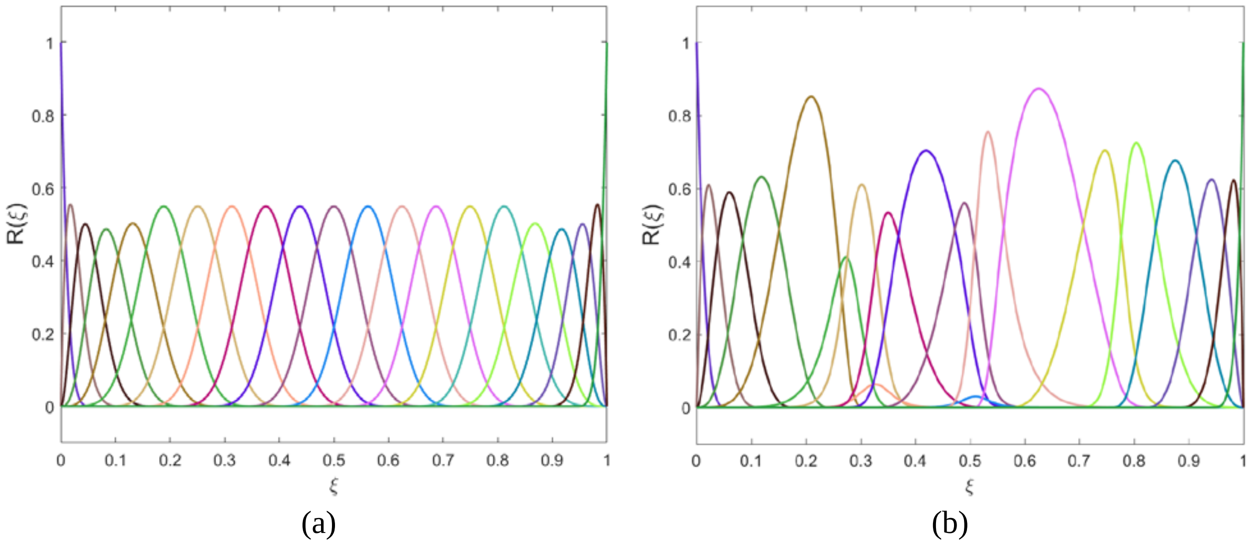


Fig. 14. (a) Initial, and (b) optimal sets of quintic basis functions associated with Fig. 13.

4.2 Extensions and further applications

While, in this paper, we limited our study to applying the proposed generalizations to NURBS curves, they can be similarly applied to surfaces and volumes which is the subject of our future research. Moreover, due to fundamental similarities between different variations of splines, these generalizations seem plausible to other rational forms of splines such as T-splines, Tri-angular Béziers, etc.

In addition to the discussed applications in CAD, there are other areas of applications of NURBS where employing the weights as additional design variables for better flexibility can be problematic or sometimes impossible. For instance, while we limited our numerical experiments to approximation over curved domains, GNURBS may also help circumventing the difficulties of considering the weights as degrees of freedom in general curve/surface fitting problems. As previously studied in [22, 23], employing the weights as additional degrees of freedom in data approximation can deteriorate the surface parameterization, and lead to undesirable results. In this regard, existing studies suggest imposing bounding constraints on the variation of the weights explicitly or via regularization [11, 20, 21], to avoid this issue. However, this limits the obtained improvement in the accuracy of approximation, especially in the case of problems containing rapid variation in data or field where the weights tend to take extreme values.

On the other hand, employing the suggested variations of NURBS, one can create a good parameterization and preserve it while including the *control weights* as design variables for fitting the curve/surface to 3D data points, without imposing any limitations on the values of the weights. Further potential applications in CAD where GNURBS can be exploited with improved flexibility include NURBS-based metamodeling [42], which is of significant interest in engineering design.

Furthermore, owing to the inherent properties of NURBS, they have been extensively used in computational mechanics for the optimization of different fields of interest over a computational domain. For instance, Qian [43] employs B-spline basis for the representation of density field in FEM-based topology optimization as an intrinsic filtering technique. Within the framework of IGA, numerous studies have been performed where the same NURBS based parameterization of computational domain has also been used for the representation of different fields which need to be optimized over the domain in various applications such as size optimization of curved beams [44–46], topology optimization [8, 47–49], optimization of material distribution in functionally graded materials (FGMs) [50, 51] etc.

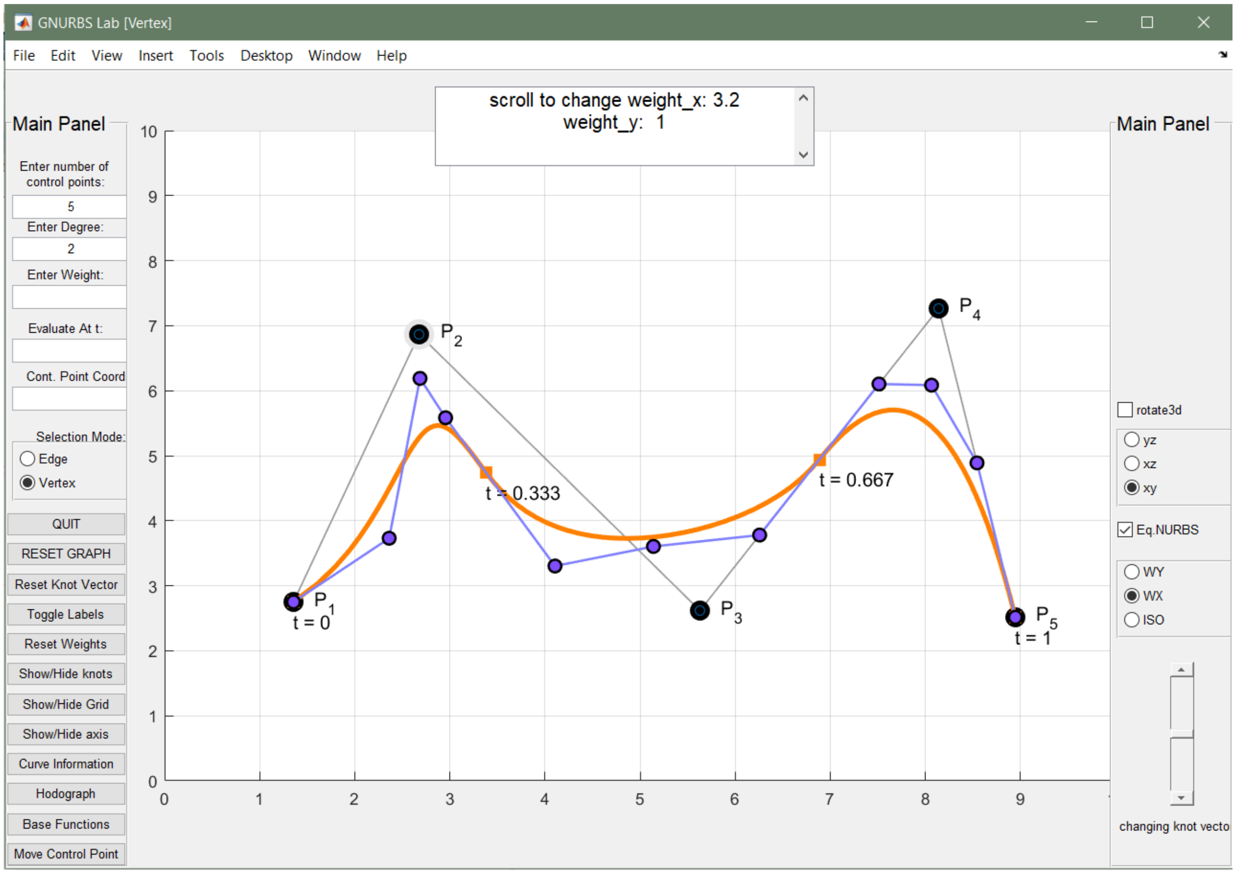
Having examined these studies, it can be noticed that in this class of applications, the parameterization of the design domain must remain fixed throughout the optimization process. Moreover, many of them require linear parameterization of the design domain and achieve this by placing the control points at their Greville abscissae, see e.g. [43, 50]. Hence, they are only able to treat the out-of-plane coordinates of control points as design variables, as the variation of weights alters the underlying parameterization which is disallowed.

Owing to the proposed GNURBS representations with decoupled weights, one can now treat the out of plane weights as additional design variables while setting up the optimization problem and still preserve the underlying geometry as well as its parameterization unchanged. As the presented numerical results suggest, this idea can lead to significant improvement in the flexibility in both cases of smooth as well as rapidly varying fields. Exploring these applications is the subject of future studies.

5. **MATLAB Toolbox:** GNURBS Lab

In order to facilitate understanding the behavior of GNURBS and further abilities they provide, a comprehensive interactive MATLAB toolbox, *GNURBS Lab*, has been developed. This toolbox is developed via the extension of an existing NURBS toolbox in MATLAB, *Bspline Lab,* available as an opensource package under GNU license at github.com.

A snapshot of the *GNURBS Lab* environment is depicted in Fig. 15, which demonstrates some of the available features in this software. The figure shows an example of designing a quadratic GNURBS curve with 5 control points constructed over a uniform knot-vector. Employing the provided tools, one can easily manipulate any defining parameter of the curve, including the locations of control points, knots or weight components, and observe the changes interactively in both the original GNURBS and its equivalent higher order counterpart, simultaneously.



scroll to change weight_x: 3.2
weight_y: 1

Enter number of control points: 5

Enter Degree: 2

Enter Weight:

Selection Mode:
○ Edge
● Vertex

□ rotate3d
○ yz
○ xz
● xy
☑ Eq.NURBS
○ WY
● WX
○ ISO

Fig. 15. A snapshot of *GNURBS lab*.

The open-source toolbox is available at http://www.ersl.wisc.edu/software/GNURBS-Lab.zip Detailed instructions for using this toolbox is also available as an additional document *Manual.pdf* via the same link.

6. Conclusion

We presented two generalizations of NURBS, referred to as GNURBS, by decoupling of the weights associated with the control points along different physical coordinates. These generalizations, which can be obtained using either a non-isoparametric or an isoparametric concept, improve the flexibility of NURBS and circumvent its deficiencies by providing the possibility of treating the weights as additional design variables in special applications. It was proved that these representations are only variations of classic NURBS and do not constitute a new superset of NURBS. The superior approximation abilities of these variations for both smooth and rapidly varying functions were shown via simple examples. However, as pointed out in Section

4.2, there are many other areas of applications which can potentially benefit from GNURBS. A comprehensive MATLAB toolbox, *GNURBS Lab*, was developed to demonstrate the behavior of GNURBS in a fully interactive manner. Further, although we limited our study to NURBS curves, similar extensions are applicable to surfaces and volumes, as well as perhaps any other rational form of splines. Overall, GNURBS provides a new powerful technology with superior flexibility while including NURBS as a special case.

Acknowledgements

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

References

1. Versprille KJ (1975) Computer-aided Design Applications of the Rational B-spline Approximation Form. Syracuse University

2. J. Austin Cottrell, Thomas J. R. Hughes YB (2009) Isogeometric Analysis: Toward Integration of CAD and FEA. John Wiley & Sons

3. Hughes TJR, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput Methods Appl Mech Eng 194:4135–4195. https://doi.org/10.1016/j.cma.2004.10.008

4. Mishra BP, Barik M (2018) NURBS-augmented finite element method for stability analysis of arbitrary thin plates. Eng Comput 0:1–12. https://doi.org/10.1007/s00366-018-0603-9

5. Qian X (2010) Full analytical sensitivities in NURBS based isogeometric shape optimization. Comput Methods Appl Mech Eng 199:2059–2071. https://doi.org/10.1016/j.cma.2010.03.005

6. Takahashi T, Yamamoto T, Shimba Y, et al (2018) A framework of shape optimisation based on the isogeometric boundary element method toward designing thin-silicon photovoltaic devices. Eng Comput 0:1–27. https://doi.org/10.1007/s00366-018-0606-6

7. Lieu QX, Lee J (2017) A multi-resolution approach for multi-material topology optimization based on isogeometric analysis. Comput Methods Appl Mech Eng 323:272–302. https://doi.org/10.1016/j.cma.2017.05.009

8. Taheri AH, Suresh K (2017) An isogeometric approach to topology optimization of multi-material and functionally graded structures. Int J Numer Methods Eng 109:668–696. https://doi.org/10.1002/nme.5303

9. Coelho M, Roehl D, Bletzinger K-U (2016) Material Model Based on Response Surfaces of NURBS Applied to Isotropic and Orthotropic Materials. In: Muñoz-Rojas PA (ed) Computational Modeling, Optimization and Manufacturing Simulation of Advanced Engineering Materials. Springer International Publishing, Cham, pp 353–373

10. Coelho M, Roehl D, Bletzinger KU (2017) Material model based on NURBS response surfaces.

Appl Math Model 51:574–586. https://doi.org/10.1016/j.apm.2017.06.038

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

12. Kanna SA, Tovar A, Wou JS, El-Mounayri H (2014) Optimized NURBS Based G-Code Part Program for High-Speed CNC Machining. In: ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers

13. Sederberg, T.W., Cardon, D.L., Finnigan, G.T., North, N.S., Zheng, J., Lyche T (2004) T-spline simplification and local refinement. ACM Trans Graph 23:276–283

14. Sederberg, T.N., Zhengs, J.M., Bakenov, A., Nasri A (2003) T-splines and T-NURCCSs. ACM Trans Graph 22:477–484

15. Chen W (2008) Generalized Hierarchical NURBS for Interactive Shape Modification. In: VRCAI ’08 Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry. pp 1–4

16. Hu G, Wu J, Qin X (2018) A novel extension of the Bézier model and its applications to surface modeling. Adv Eng Softw 125:27–54. https://doi.org/10.1016/j.advengsoft.2018.09.002

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

18. Wang Q, Hua W, Li G, Bao H (2004) Generalized NURBS curves and surfaces. Proc - Geom Model Process 2004 365–368. https://doi.org/10.1023/B:JMSC.0000008091.55395.ee

19. Piegl L, Tiller W (1995) The NURBS Book, 1st ed. Springer-Verlag Berlin Heidelberg

20. Carlson N (2009) NURBS Surface Fitting with Gauss-Newton. Lulea University of Technology

21. Ma W (1994) NURBS-based computer aided design modelling from measured points of physical models. Catholic University of Leuven

22. Dimas E, Briassoulis D (1999) 3D geometric modelling based on NURBS: a review. Adv Eng Softw 30:741–751

23. Piegl L (1991) On NURBS: A Survey. IEEE Comput Graph Appl 55–71

24. Mehaute A Le, Schumaker LL, Rabut C (1997) Curves and Surfaces with Applications in CAGD, 1st ed. Vanderbilt University Press

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

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

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

28. Lee ETY (1994) Computing a chain of blossoms, with application to products of splines. Comput Aided Geom Des 11:597–620

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

30. Farin G (2001) Curves and Surfaces for CAGD A Practical Guide, 5th ed. Morgan Kaufmann

31. Alfeld P, Neamtu M, Schumaker LL (1996) Fitting scattered data on sphere-like surfaces using spherical splines. J Comput Appl Math 73:5–43. https://doi.org/10.1016/0377-0427(96)00034-9

32. Fasshauer GE, Schumaker LL (1998) Scattered Data Fitting on the sphere. Vanderbilt University Press, Nashville, TN

33. Pu X, Liu W (2009) A subdivision scheme for approximating circular helix with NURBS curve. Proceeding 2009 IEEE 10th Int Conf Comput Ind Des Concept Des E-Business, Creat Des Manuf - CAID CD’2009 620–624. https://doi.org/10.1109/CAIDCD.2009.5374879

34. Yang X (2003) High accuracy approximation of helices by quintic curves. Comput Aided Geom Des 20:303–317. https://doi.org/10.1016/S0167-8396(03)00074-8

35. Juhasz I (1995) Approximating the helix with rational cubic Bezier curves. Comput Des 27:587–593

36. Shojaee S, Izadpenah E, Haeri A (2012) Imposition of Essential Boundary Conditions in Isogeometric Analysis Using the Lagrange Multiplier Method. Int J Optim Civ Eng 2:247–271

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

38. Anand Embar, John Dolbow IH (2010) Imposing Dirichlet boundary conditions with Nitsche’s method and spline-based finite elements. Int J Numer Methods Eng 83:877–898. https://doi.org/10.1002/nme.2863

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

40. Pottmann H, Leopoldseder S, Hofer M (2002) Approximation with Active B-spline Curves and Surfaces. In: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications (PG’02). IEEE, pp 8–25

41. Erdönmez C (2013) N-Tuple Complex Helical Geometry Modeling Using Parametric Equations. Eng Comput 30:715–726. https://doi.org/10.1007/s00366-013-0319-9

42. Turner CJ, Crawford RH, Campbell MI (2007) Multidimensional sequential sampling for NURBs-based metamodel development. Eng Comput 23:155–174. https://doi.org/10.1007/s00366-006-0051-9

43. Qian X (2013) Topology optimization in B-spline space. Comput Methods Appl Mech Eng 265:15–35. https://doi.org/10.1016/j.cma.2013.06.001

44. Nagy AP, Abdalla MM, Gürdal Z (2010) Isogeometric sizing and shape optimisation of beam structures. Comput Methods Appl Mech Eng 199:1216–1230. https://doi.org/10.1016/j.cma.2009.12.010

45. Nagy AP, Abdalla MM, Gürdal Z (2010) Isogeometric sizing and shape optimisation of beam structures. Comput Methods Appl Mech Eng 199:1216–1230. https://doi.org/10.1016/j.cma.2009.12.010

46. Liu H, Yang D, Wang X, et al (2018) Smooth size design for the natural frequencies of curved Timoshenko beams using isogeometric analysis. Struct Multidiscip Optim.

https://doi.org/10.1007/s00158-018-2119-8

47. Hassani B, Khanzadi M, Tavakkoli SM (2012) An isogeometrical approach to structural topology optimization by optimality criteria. Struct Multidiscip Optim 45:223–233. https://doi.org/10.1007/s00158-011-0680-5

48. Wang Y, Benson DJ (2016) Isogeometric analysis for parameterized LSM-based structural topology optimization. Comput Mech 57:19–35. https://doi.org/10.1007/s00466-015-1219-1

49. Dedè L, Borden MMJ, Hughes TJRT (2012) Isogeometric analysis for topology optimization with a phase field model. Arch Comput Methods Eng 19:427–465. https://doi.org/10.1007/s11831-012-9075-z

50. Lieu QX, Lee J (2017) Modeling and optimization of functionally graded plates under thermo-mechanical load using isogeometric analysis and adaptive hybrid evolutionary firefly algorithm. Compos Struct 179:89–106. https://doi.org/10.1016/j.compstruct.2017.07.016

51. Taheri AH, Hassani B, Moghaddam NZ (2014) Thermo-elastic optimization of material distribution of functionally graded structures by an isogeometrical approach. Int J Solids Struct 51:416–429. https://doi.org/10.1016/j.ijsolstr.2013.10.014