

Surface approximations using Generalized NURBS

Alireza H. Taheri¹, Krishnan Suresh^{1*}

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

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

Abstract:

We extend the concept of generalized NURBS (GNURBS), recently introduced by the authors for parametric curves, to bivariate parametric surfaces. These generalizations are obtained via either explicit or implicit decoupling of the weights along different physical coordinates. This decoupling allows for treating the weights as additional degrees of freedom in a wider range of applications compared to classic NURBS surfaces, providing additional flexibility and increased control. This proposed concept effectively improves the capability of NURBS and alleviates its deficiencies in certain applications. In particular, we will demonstrate that GNURBS can be effectively used for improved approximation of certain class of surfaces such as helicoids, revolved surfaces and minimal surfaces. It will also be established that these proposed generalizations can be exactly transformed to equivalent, but higher order, classic NURBS surfaces, ensuring a strong theoretical foundation. Finally, a comprehensive MATLAB toolbox, GNURBS3D-Lab, has been developed and introduced in order to better demonstrate the behavior and properties of GNURBS surfaces compared to classic NURBS.

Keywords: GNURBS, bivariate surfaces, directional weights, non-isoparametric.

1. Introduction

Non-Uniform Rational B-Splines (NURBS) were first introduced in 1975 by Versprille [1] via rational extension of B-splines. The primary motivation for introducing NURBS was to represent conical shapes precisely. This is the critical advantage of NURBS over other polynomial-based classes of splines, and one of the main reasons for its prevalence. Due to this crucial ability, NURBS are still the prevalent technology for curve and surface modelling in Computer-Aided Design/Computer-Aided Manufacturing (CAD/CAM), and an integral part of most existing CAD/CAM commercial software.

The applications of this rational form, however, is not limited to precise representation of conics. Other applications of NURBS can also be found in CAD where the weights have been employed as additional degrees of freedom for improved flexibility. A thorough review of these applications has been reported by the authors in [2]. Moreover, in addition to CAD/CAM, NURBS have also been extensively used in many other areas of applications such as 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.

Despite being a powerful tool in engineering design, NURBS have multiple shortcomings which restricts its capability in certain applications [13]. A thorough review of the advantages and limitations of NURBS is provided in [13]. A major shortcoming of NURBS which has received significant attention is their inability to allow for local refinement. Due to the rigid tensor-product structure of NURBS, knot-insertion is a global operation and cannot be performed locally. This was soon known as a fundamental limitation of NURBS, since local knot-insertion is critical in many applications and is considered a common and efficient way for achieving desirable accuracy in approximating sharp features [14] or scattered data of highly varying density [15]. For instance, Leal et al. [14] mention that “*Despite the advantages of fitting with NURBS, it is still necessary to improve the representation of sharp features like high curvatures, edges and corners with this fitting method.*”

In order to remove this fundamental limitation, various generalizations of NURBS have been proposed so far. The concept of hierarchical B-spline constructions which considers multilevel B-spline extensions where the tensor-product structure is preserved at any level was originally proposed by Forsey and Bartels in 1988 [16]. The application of hierarchical splines for adaptive

scattered data fitting has recently been investigated by Bracco et al. [15]. To efficiently deal with non-trivial data configurations, they describe the local solutions in terms of (variable-degree) polynomial approximations according not only to the number of data points locally available, but also to the smallest singular value of the local collocation matrices. These local approximations are subsequently combined without the need of additional computations with the construction of hierarchical quasi-interpolants described in terms of truncated hierarchical B-splines.

Generalized Hierarchical NURBS (H-NURBS) were introduced in 2008 by Chen et al. [17] by extending the idea of hierarchical B-splines to NURBS. Another popular technology are T-splines [18, 19] which constitute a superset of NURBS, and provide the local refinement properties by allowing for unstructured-ness. Most recent class of splines which removes the limitations of T-splines are Unstructured-splines (U-splines) that have been developed by Thomas [20].

In addition to the above technologies, an alternative strategy for addressing the same issue has also been adopted by some researchers. The basic idea of these studies is to preserve the tensor-product structure of NURBS, and instead include the weights of control points as additional degrees of freedom. This idea has also shown promising results for the approximation of scattered data of highly varying local density [21] as well as for the representation of sharp geometric features [14]. For instance, Leal et al. [14] present a new method for improving NURBS surface sharp feature representation that first subdivides the fitting data in clusters, by using SOM; then, in each cluster they use an evolutionary strategy to obtain the optimal weights of the NURBS such that the fitting error is minimized and the representation of sharp features is improved. While including the weights as additional degrees of freedom in data approximation with NURBS usually results in non-linear algorithms, Ma [11, 22] proposes a two-step linear algorithm which yields the optimal coordinates of control points as well as their optimal weights by solving two separate linear systems of equations.

As discussed in [2], in spite of 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 [13] point out that a bad choice of weights in approximation can lead to poor curve/surface parameterization. Piegl [23] states that “*improper application of the weights can result in a very bad parameterization, which can destroy subsequent surface constructions*”. Further, there are many applications where

treating the weights as additional design variables is essentially impossible. These limitations inspired introducing the concept of Generalized NURBS (GNURBS) which is thoroughly discussed for parametric curves in [2] by the authors. Further, the extension of this mathematical model was introduced in [24] as a means for improved solution of boundary value problems using the isogeometric analysis method.

The focus of this paper is to comprehensively study various types of GNURBS surfaces, provide mathematical proofs of their theoretical properties, and explore their applications in the context of CAGD. In particular, we will investigate a common application of these generalizations for improved surface approximation. It will be shown that, despite simply being disguised forms of classic NURBS, these generalizations provide significantly better approximation abilities compared to classic NURBS.

The remainder of this paper is organized as follows: in Sections 2, 3 and 4 we introduce different generalizations of NURBS, and develop their theoretical properties. We explore the application of GNURBS for improved approximation of surfaces in Section 5 where least-square approximation algorithms are developed. A series of numerical examples are presented in Section 6 where the performance of GNURBS compared to NURBS for the approximation of different class of surfaces is studied. Further potential areas of applications and extensions of GNURBS are discussed in Section 7. An interactive MATLAB toolbox for GNURBS surfaces is introduced in Section 8, and finally conclusions are drawn in Section 9.

2. Generalized NURBS surfaces: non-isoparametric form via explicit decoupling of the weights

We recall that the equation of a NURBS surface is defined in the following parametric form [23]

$$\mathbf{S}(\xi, \eta) = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} R_{ij}^{p,q}(\xi, \eta) \mathbf{P}_{ij} \quad \begin{array}{l} a \leq \xi \leq b \\ c \leq \eta \leq d \end{array} \quad (1)$$

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

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

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 $\Xi = \{\xi_0, \xi_1, \dots, \xi_{n_1+p}\}$ and $\mathbf{H} = \{\eta_0, \eta_1, \dots, \eta_{n_2+q}\}$, 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 [2].

We extend here the concept of Generalized Non-Uniform Rational B-Splines (GNURBS) [2] to surfaces by modifying Eq. (1) as follows

$$\mathbf{S}(\xi, \eta) = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} \mathbf{R}_{ij}^{p,q}(\xi, \eta) \odot \mathbf{P}_{ij} \quad \begin{array}{l} a \leq \xi \leq b \\ c \leq \eta \leq d \end{array} \quad (3)$$

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

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

In the above equation, $(w_{ij}^x, w_{ij}^y, w_{ij}^z)$ represent 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 which are discussed in detail in the following section.

2.1. Theory and properties

It can be shown that due to coordinate-dependence of basis functions, a GNURBS surface (in its original form) need not satisfy properties such as strong convex hull and affine invariance. We demonstrate here that most of the theoretical properties which were discussed for GNURBS curves in [2] can be extended for GNURBS surfaces.

1. Local modification effect¹

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 d , while the surface parameterization in the other directions will be preserved. This is, in fact, the key difference between GNURBS and NURBS which provides additional flexibility. 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. (3) 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 $\Xi = \mathbf{H} = \{0, 0, 0, 0.2, 0.4, 0.6, 0.8, 1, 1, 1\}$. The employed net of control points is illustrated in Figure 1. As the figure shows, the heights of all control points are set to zero except for z_{44} which is raised to 1.

¹ This property has already been studied in [24]

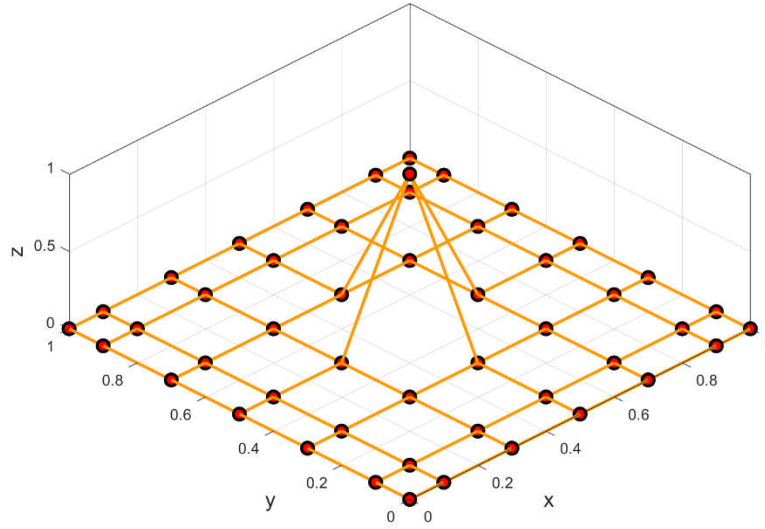


Figure 1. Employed control net for construction of different NURBS surfaces.

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

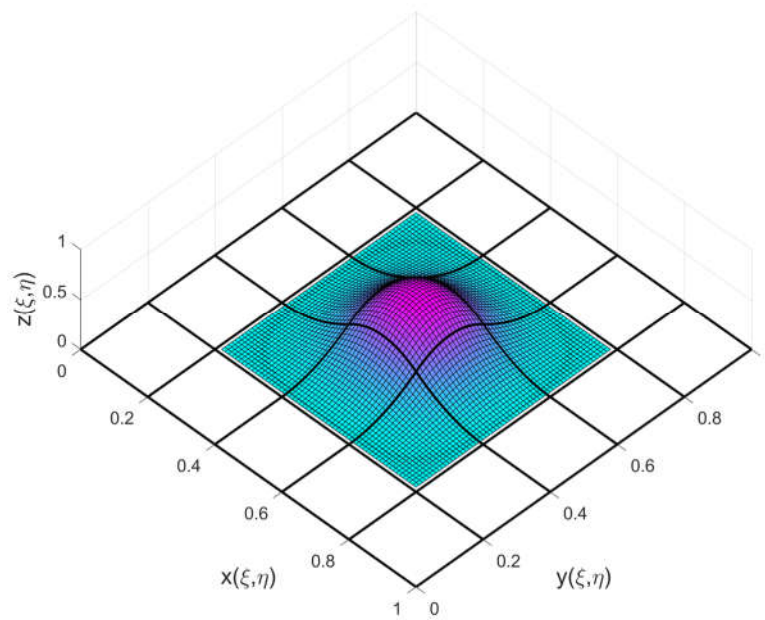


Figure 2. The B-spline surface in physical space.

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

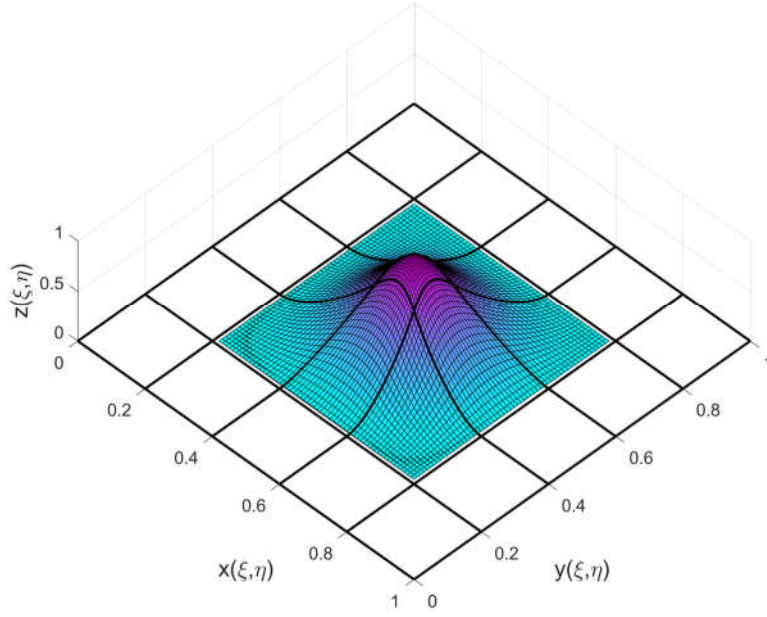


Figure 3. The NURBS surface with $w_{44} = 4$ in physical space.

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

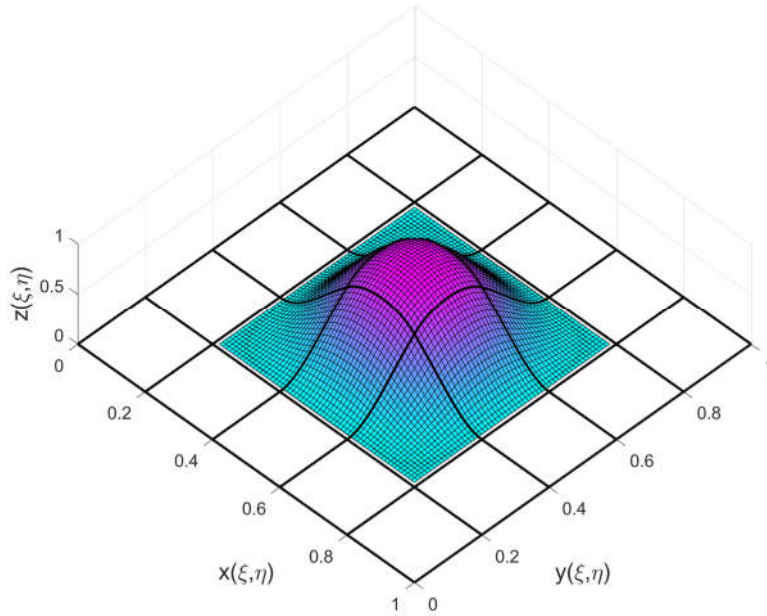


Figure 4. The GNURBS surface with $w_{44}^r = 4$ in physical space.

Note that the depicted GNURBS surface in Figure 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 Figure 3 and Figure 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 Figure 4). It will be seen later that this property is critical for treating the weights as additional degrees of freedom in certain applications.

2. Axis-aligned bounding box (AABB):

Every GNURBS knot-element lies within the *axis-aligned bounding box* of its corresponding control points. That is, if $(\xi, \eta) \in [\xi_i, \xi_{i+1}) \times [\eta_j, \eta_{j+1})$, then $\mathbf{S}(\xi, \eta)$ lies within the bounding box of the control points \mathbf{P}_{kl} , $i - p \leq k \leq i$ and $j - q \leq l \leq j$.

Note that Eq. (3) can be easily written in the following form:

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

Accordingly, Eq. (5) could be written as

$$\mathbf{S}(\xi, \eta) = \mathbf{S}_x(\xi, \eta) + \mathbf{S}_y(\xi, \eta) + \mathbf{S}_z(\xi, \eta), \quad \begin{cases} a \leq \xi \leq b \\ c \leq \eta \leq d \end{cases} \quad (6)$$

where $\mathbf{S}_x(\xi, \eta)$, $\mathbf{S}_y(\xi, \eta)$ and $\mathbf{S}_z(\xi, \eta)$ are simply classic NURBS surfaces. From a geometric standpoint, each of these surfaces is the projection of the original non-isoparametric surface onto the corresponding physical axes.

The following figure shows a graphical representation of the above equations for a quadratic \times cubic GNURBS surface constructed over the knot vectors $\Xi = \{0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1\}$ and $H = \{0, 0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1\}$. Random weights in z -direction have been assigned to the control points and the control points are plotted proportional to these weights in size for better insight.

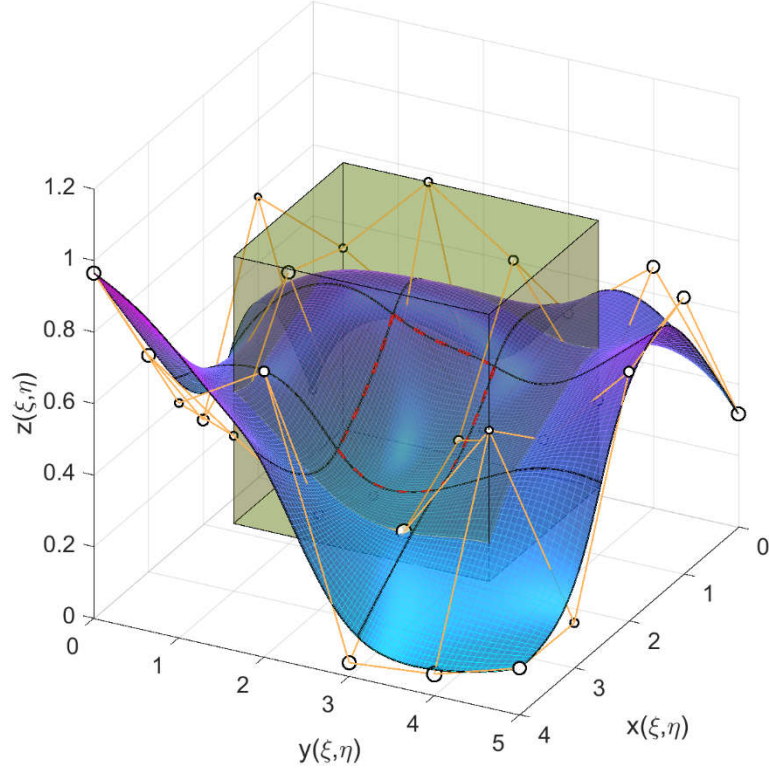


Figure 5. Geometric representation of the bounding box property for a GNURBS surface.

Since each of these projected surfaces is a classic NURBS surface, they satisfy the convex hull property. Therefore, the middle knot-element of the surface which is marked in Figure 5, must lie within the convex hulls of its corresponding control points on all three projected surfaces. That is, if $(\xi, \eta) \in \left[\frac{1}{3}, \frac{2}{3}\right] \times \left[\frac{1}{3}, \frac{2}{3}\right]$, then $\mathbf{S}_x(\xi, \eta)$ lies within the convex hull of the control points $(x_{kl}, 0, 0)$, $1 \leq k \leq 3$ and $1 \leq l \leq 4$ which is the space between the two planes parallel to yz -plane. Similarly, $\mathbf{S}_y(\xi, \eta)$ lies within the convex hull of the control points $(0, y_{kl}, 0)$, $1 \leq k \leq 3$ and $1 \leq l \leq 4$ which is the area between the two planes parallel to xz -plane, and $\mathbf{S}_z(\xi, \eta)$ lies within the convex hull of the control points $(0, 0, z_{kl})$, $1 \leq k \leq 3$ and $1 \leq l \leq 4$ which is the area between the two planes parallel to xy -plane. Consequently, $\mathbf{S}(\xi, \eta)$ is contained in the intersection of these six planes, which is the highlighted box area shown in Figure 5, referred to as the axis-aligned bounding box of \mathbf{P}_{kl} , $1 \leq k \leq 3$ and $1 \leq l \leq 4$. It is obvious that this property is less strict than the strong convex-hull property of classic NURBS surfaces.

3. Non-isoparametric 3D GNURBS surfaces with partial decoupling of the weights

A more practical variation of GNURBS, which will be the emphasis for the rest of this paper, 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

$$\begin{cases} x(\xi, \eta) \\ y(\xi, \eta) \\ z(\xi, \eta) \end{cases} = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} \begin{cases} R_{ij}^{xy}(\xi, \eta) x_{ij} \\ R_{ij}^{xy}(\xi, \eta) y_{ij} \\ R_{ij}^z(\xi, \eta) z_{ij} \end{cases} \quad \begin{matrix} a \leq \xi \leq b \\ c \leq \eta \leq d \end{matrix} \quad (7)$$

where

$$R_{ij}^{xy}(\xi, \eta) = \frac{N_{ij}^{p,q}(\xi, \eta) w_{ij}^{xy}}{\sum_{k=0}^{n_1} \sum_{l=0}^{n_2} N_{kl}^{p,q}(\xi, \eta) w_{kl}^{xy}} \quad (8)$$

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 planer surface in x - y plane.

3.1. Equivalence with NURBS

Despite losing some properties of NURBS which might be of interest in certain applications, we recall here a theorem [24] 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 $(2p, 2q)$ in the following form:

$$\begin{cases} x(\xi, \eta) \\ y(\xi, \eta) \\ z(\xi, \eta) \end{cases} = \sum_{i=0}^{\hat{n}_1} \sum_{j=0}^{\hat{n}_2} \hat{R}_{ij}^{2p, 2q}(\xi, \eta) \begin{cases} X_i \\ Y_i \\ Z_i \end{cases} \quad (9)$$

where

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

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. The proof of this theorem has been provided in [24].

As discussed in [24], in the special case of Rational Bézier (R-Bézier) surfaces, the following straightforward analytical expressions can be obtained for the coefficients of the equivalent higher order R-Bézier surface:

$$\begin{aligned} W_{ij} &= \sum_{k=\max(0, i-p)}^{\min(p, i)} \sum_{l=\max(0, j-q)}^{\min(q, j)} \lambda_{ik}^p \lambda_{jl}^q w_{i-k, j-l}^{xy} w_{kl}^z \\ X_{ij} &= \frac{1}{W_{ij}} \sum_{k=\max(0, i-p)}^{\min(p, i)} \sum_{l=\max(0, j-q)}^{\min(q, j)} \lambda_{ik}^p \lambda_{jl}^q w_{kl}^{xy} x_{kl} w_{i-k, j-l}^z \\ Y_{ij} &= \frac{1}{W_{ij}} \sum_{k=\max(0, i-p)}^{\min(p, i)} \sum_{l=\max(0, j-q)}^{\min(q, j)} \lambda_{ik}^p \lambda_{jl}^q w_{kl}^{xy} y_{kl} w_{i-k, j-l}^z \\ Z_{ij} &= \frac{1}{W_{ij}} \sum_{k=\max(0, i-p)}^{\min(p, i)} \sum_{l=\max(0, j-q)}^{\min(q, j)} \lambda_{ik}^p \lambda_{jl}^q w_{i-k, j-l}^{xy} z_{kl} w_{kl}^z \end{aligned} \quad (11)$$

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

Figure 6(a) shows an example of a degree (2,3)th GNURBS surface with random directional weights assigned in z-direction. Its equivalent higher order NURBS surface obtained using the above theorem is depicted in Figure 6(b). Note that the size of control points in these figures are plotted proportional to their weights for better insight.

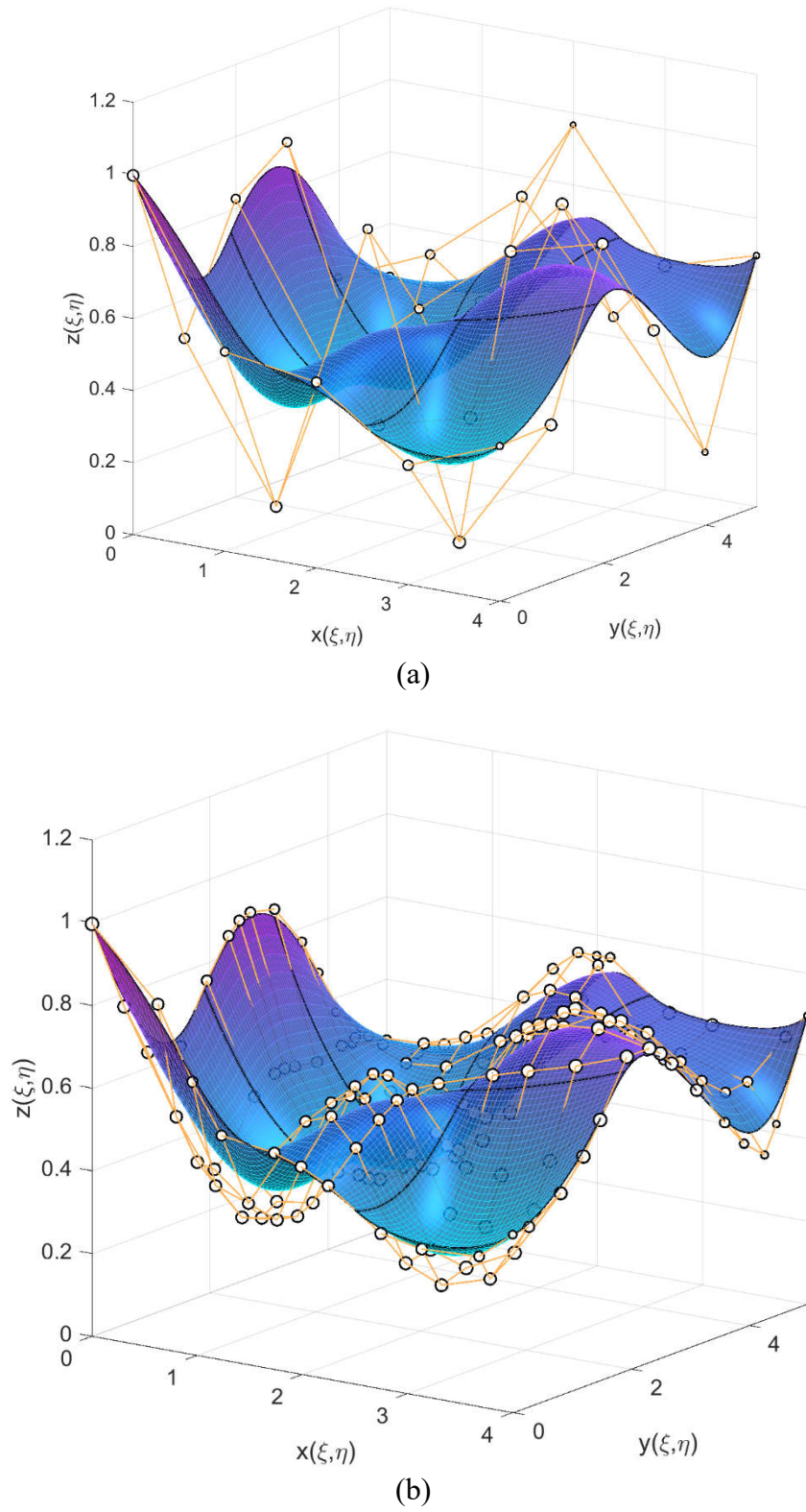


Figure 6. (a) A degree (2,3)th GNURBS surface with random weights assigned in z -direction, and (b) its equivalent (isoparametric) NURBS surface of degree (4,6).

4. Generalized NURBS surfaces: isoparametric form via implicit decoupling of the weights

It is interesting to note that the equivalent higher order NURBS representation in (9) itself provides another variation of NURBS which can be directly employed as another alternative to NURBS with better flexibility in many applications.

In order to clarify how this equation provides additional flexibility than classic NURBS, we first derive a more generic form of this equation via an alternative approach using an extension of order elevation technique. In this case, we limit our study to rational Bézier surfaces for simplicity.

Assume a 2D R-Bézier surface of degree (p, q) is given as follows

$$\begin{Bmatrix} x(\xi, \eta) \\ y(\xi, \eta) \end{Bmatrix} = \sum_{i=0}^p \sum_{j=0}^q \frac{B_{ij}^{p,q}(\xi, \eta) w_{ij}^{xy}}{w^{xy}(\xi, \eta)} \begin{Bmatrix} x_{ij} \\ y_{ij} \end{Bmatrix} \quad (12)$$

In order to elevate the degree of this surface by (r, s) , we can simply multiply both numerator and denominator of this equation by any arbitrary expression in the following form

$$w^z(\xi, \eta) = \sum_{i=0}^r \sum_{j=0}^s B_{ij}^{r,s}(\xi, \eta) w_{ij}^z \quad (13)$$

Recalling Theorem 1, we can obtain the higher order R-Bézier surface with (r, s) degree elevations as

$$\begin{Bmatrix} x(\xi, \eta) \\ y(\xi, \eta) \end{Bmatrix} = \sum_{i=0}^{\hat{p}} \sum_{j=0}^{\hat{q}} \hat{R}_{ij}^{\hat{p}, \hat{q}} \begin{Bmatrix} X_{ij} \\ Y_{ij} \end{Bmatrix} \quad (14)$$

where

$$\hat{R}_{ij}^{\hat{p}, \hat{q}}(\xi, \eta) = \frac{B_{ij}^{\hat{p}, \hat{q}}(\xi, \eta) W_{ij}}{\sum_{k=0}^{\hat{p}} \sum_{l=0}^{\hat{q}} B_{kl}^{\hat{p}, \hat{q}}(\xi, \eta) W_{kl}} \quad (15)$$

in which $\hat{p} = p + r, \hat{q} = q + s$ and (X_{ij}, Y_{ij}, W_{ij}) can be obtained using the following relations

$$\begin{aligned}
W_{ij} &= \sum_{k=\max(0,i-r)}^{\min(p,i)} \sum_{l=\max(0,j-s)}^{\min(q,j)} \lambda_{ik}^{p,r} \lambda_{jl}^{q,s} w_{kl}^{xy} w_{i-k,j-l}^z \\
X_{ij} &= \frac{1}{W_{ij}} \sum_{k=\max(0,i-r)}^{\min(p,i)} \sum_{l=\max(0,j-s)}^{\min(q,j)} \lambda_{ik}^{p,r} \lambda_{jl}^{q,s} w_{kl}^{xy} x_{kl} w_{i-k,j-l}^z \\
Y_{ij} &= \frac{1}{W_{ij}} \sum_{k=\max(0,i-r)}^{\min(p,i)} \sum_{l=\max(0,j-s)}^{\min(q,j)} \lambda_{ik}^{p,r} \lambda_{jl}^{q,s} w_{kl}^{xy} y_{kl} w_{i-k,j-l}^z
\end{aligned} \tag{16}$$

where $\lambda_{ij}^{\alpha,\beta} = \frac{\binom{\alpha}{j} \binom{\beta}{i-j}}{\binom{\alpha+\beta}{i}}.$

Observe that this procedure can be seen as a natural extension of the classic order elevation techniques in the literature [25, 26]. In fact, one can simply recover the common order elevation algorithm by assigning $w_{ij}^z = 1, \forall(i, j)$ in Eq. (13). We will refer to this procedure as *generalized order elevation* hereafter. Now assume we intend to add another dimension to the degree-elevated representation in Eq. (14) in an isoparametric manner. For this purpose, we extend this equation as

$$\begin{Bmatrix} x(\xi, \eta) \\ y(\xi, \eta) \\ z(\xi, \eta) \end{Bmatrix} = \sum_{i=0}^{\hat{p}} \sum_{j=0}^{\hat{q}} \hat{R}_{ij}^{\hat{p}, \hat{q}}(\xi, \eta) \begin{Bmatrix} X_{ij} \\ Y_{ij} \\ Z_{ij} \end{Bmatrix} \tag{17}$$

It is interesting to notice that, although Eq. (17) apparently seems to be a classic R-Bézier surface, it provides additional flexibility. Observe that in the above procedure, w_{ij}^z are arbitrary variables which can be freely chosen without perturbing the geometry or parameterization of the underlying surface in x - y plane. For better insight, we perform degree elevation on a circular annulus using the above procedure with different selections of w_{ij}^z and discuss how it differs from classic degree elevation technique.

For this purpose, we generate a 3D $(\hat{p}, \hat{q}) = (3, 2)$ isoparametric GR-Bézier surface by performing the above degree-elevation processes with $(r, s) = (1, 1)$ on an initial quarter annulus modelled by a $(p, q) = (2, 1)$ R-Bézier surface depicted in Fig. 6(a) and specifying the heights of control points of the degree-elevated surface as shown in Table 1.

Table 1. Assigned heights (z_{ij}) to the control points of the resulting degree-elevated isoparametric GR-Bézier surface.

	$i=0$	$i=1$	$i=2$	$i=3$
$j=0$	0	1	1	0
$j=1$	0	1	1	0
$j=2$	0	1	1	0

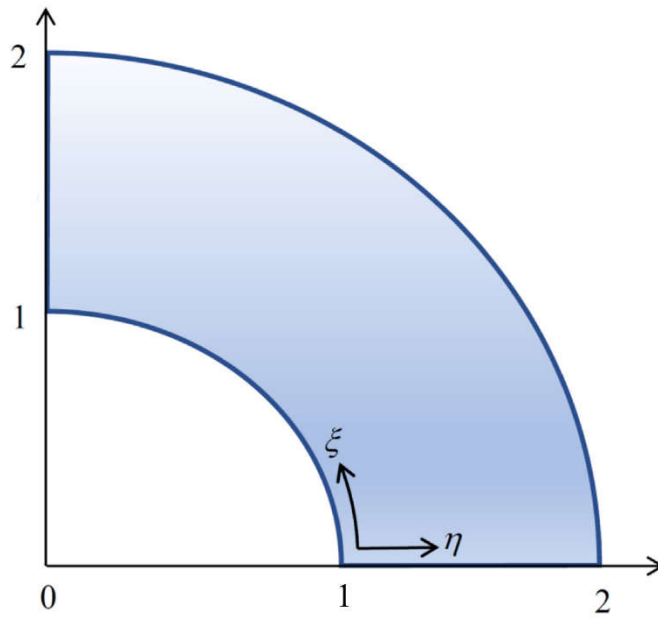


Figure 7. Configuration of the quarter annulus.

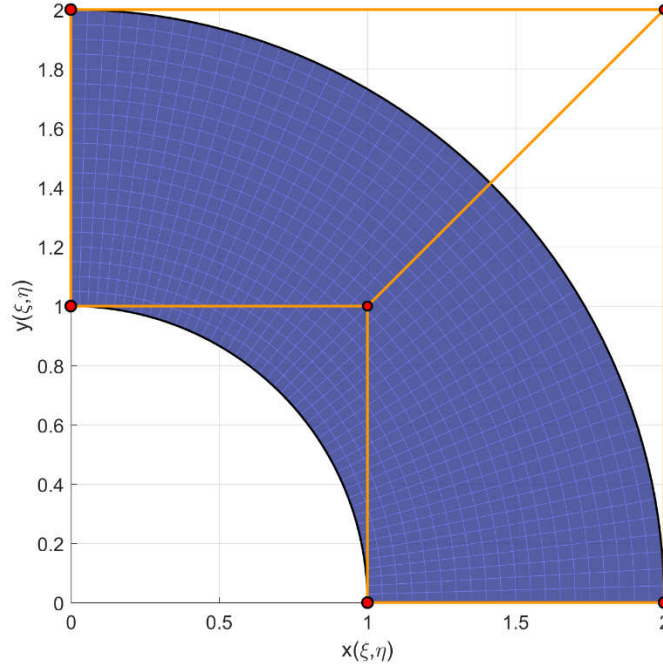


Figure 8. Exact representation of the quarter annulus with normal parameterization using a $(p, q)=(2,1)$ rational Bézier surface.

The obtained results for classic order elevation, that is, assuming unit values for all isoparametric control weights as in the following equation:

$$\begin{bmatrix} w_{11}^z & w_{12}^z \\ w_{21}^z & w_{22}^z \end{bmatrix} = \begin{bmatrix} 1.0 & 1.0 \\ 1.0 & 1.0 \end{bmatrix} \quad (18)$$

are shown in Figure 9.

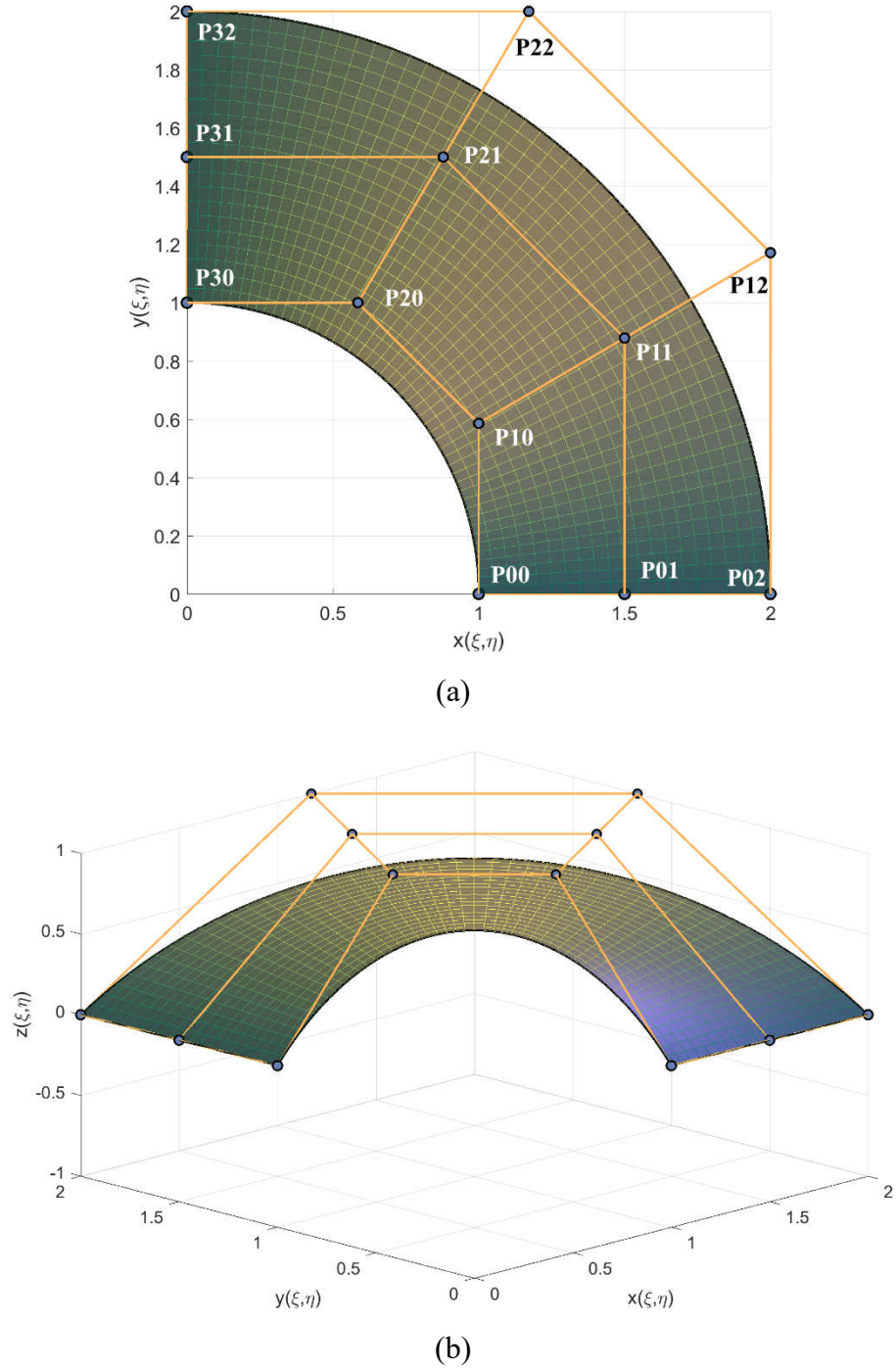


Figure 9. Classic degree-elevated R-Bézier representation of the quarter annulus with control variables of Table 3.1: (a) top view, (b) 3D view.

Moreover, the obtained results for generalized order elevation by assuming the following values for isoparametric control weights:

$$\begin{bmatrix} w_{11}^z & w_{12}^z \\ w_{21}^z & w_{22}^z \end{bmatrix} = \begin{bmatrix} 3.0 & 1.0 \\ 1.0 & 0.5 \end{bmatrix} \quad (19)$$

are depicted in Figure 10.

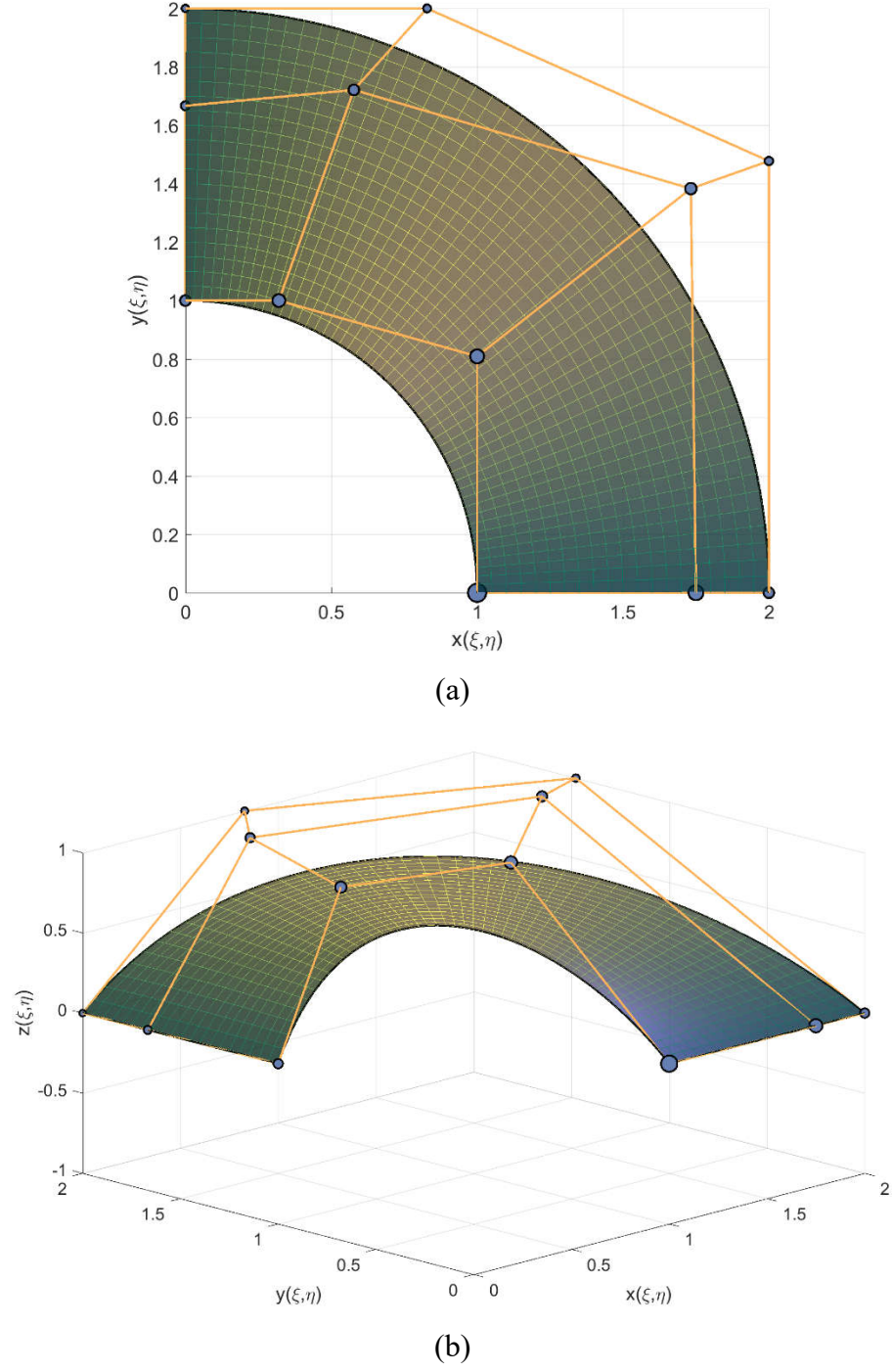


Figure 10. Generalized degree-elevated R-Bézier representation of the quarter annulus with control variables of Table 3.1: (a) top view, (b) 3D view.

As can be clearly seen in these figures, in both cases, the in-plane representation of the annular ring as well as its parameterization has remained unchanged. However, the out of plane deformation of the annular ring in the two cases are not identical.

While this variation of NURBS, which will be referred to as isoparametric GNURBS hereafter, similarly provides the same important possibility of treating the out of plane weights as additional degrees of freedom, it provides different advantages. In particular, unlike the first variation, it allows for introducing customized rationality for approximation, i.e. the number of coefficients to be considered as design variables in the denominator can be controlled here. Further, it directly lies in the NURBS space; hence, all the properties of NURBS are naturally satisfied.

The above algorithm can also be extended to NURBS in a straightforward manner using a similar three step algorithm elaborated above. That is, Eq. (17) also holds true for NURBS:

$$\begin{Bmatrix} x(\xi, \eta) \\ y(\xi, \eta) \\ z(\xi, \eta) \end{Bmatrix} = \sum_{i=0}^{\hat{n}_1} \sum_{j=0}^{\hat{n}_2} \hat{R}_{ij}^{\hat{p}, \hat{q}}(\xi, \eta) \begin{Bmatrix} X_{ij} \\ Y_{ij} \\ Z_{ij} \end{Bmatrix} \quad (20)$$

with the rational basis functions defined as

$$\hat{R}_{ij}^{\hat{p}, \hat{q}}(\xi, \eta) = \frac{N_{ij}^{\hat{p}, \hat{q}}(\xi, \eta) W_{ij}}{\sum_{k=0}^{\hat{n}_1} \sum_{l=0}^{\hat{n}_2} N_{kl}^{\hat{p}, \hat{q}}(\xi, \eta) W_{kl}} \quad (21)$$

The proposed generalizations of NURBS in Eqs. (7) and (20) can effectively improve the performance of NURBS in a wide area of applications. Exploring all these applications, however, is beyond the scope of this study. We limit our study here to a few classic examples in geometric modelling, that is the approximation of certain class of surfaces such as helical, revolved or minimal surfaces using GNURBS; and concisely point out some of their potential broader areas of applications. Finally, hereafter, we will persistently refer to Eq. (7) as the first generalization of NURBS or non-isoparametric GNURBS, while we will refer to Eq. (20) as the second generalization of NURBS or isoparametric GNURBS.

5. Least-square surface approximation using NURBS versus GNURBS

In this section, we demonstrate that the proposed generalizations of NURBS are able to provide superior approximation for certain class of surfaces compared to classic NURBS. We assume here

that a planar geometry with precise representation using NURBS, such as the annular ring in Figure 8, is given as:

$$\mathbf{S}_{xy}(\xi, \eta) = \begin{cases} x(\xi, \eta) \\ y(\xi, \eta) \end{cases}, \quad \begin{matrix} 0 \leq \xi \leq 1 \\ 0 \leq \eta \leq 1 \end{matrix} \quad (22)$$

Next, we assume that an analytical height function $z(\xi, \eta)$ is given and needs to be approximated with minimal error over the given planar surface. The problem can be posed as a least square approximation problem which leads to optimal accuracy in L^2 -norm. Considering $\{(\xi_s, \eta_s) \rightarrow (x_s, y_s, z_s) : s \in \mathcal{S}\}$ as a set of n_c chosen collocation points, the error function f to be minimized is defined as

$$f = \frac{1}{2} \sum_{s \in \mathcal{S}} \|\hat{z}(\xi_s, \eta_s) - \bar{z}_s\|^2 = \frac{1}{2} \sum_{s \in \mathcal{S}} \left\| \sum_{L \in \mathcal{L}^s} R_L(\xi_s, \eta_s) z_L - \bar{z}_s \right\|^2 \quad (23)$$

where $\hat{z}(\xi, \eta)$ is the approximated NURBS function, \mathcal{L}^s is the set of indices of non-zero basis functions at (ξ_s, η_s) , $\bar{z}_s = z(\xi_s, \eta_s)$, and z_L are the unknown *control variables*. For simplicity, the global index L is used for numbering which is defined as $L = j(n_1 + 1) + i + 1$ for the basis (i, j) . In the following, we provide the detailed formulation of this problem using NURBS as well as its different proposed generalizations.

5.1. Linear least-square approximation using NURBS

In the case of NURBS, the only unknowns to consider are *control variables* z_L . Taking the partial derivatives of f with respect to the unknowns z_L , and setting them to zero yields

$$\frac{\partial f}{\partial z_k} = 0, \quad k = 1, \dots, n_T \quad (24)$$

$$\sum_{s \in \mathcal{S}} \sum_{L \in \mathcal{L}^s} R_k(\xi_s, \eta_s) R_L(\xi_s, \eta_s) z_L = \sum_{s \in \mathcal{S}} \bar{z}_s R_k(\xi_s, \eta_s), \quad k = 1, \dots, n_T \quad (25)$$

where $n_T = (n_1 + 1) \times (n_2 + 1)$ denotes the total number of control points. Eq. (25) could be written in the matrix form

$$\sum_{s \in \mathcal{S}} \begin{pmatrix} R_1(\xi_s, \eta_s) R_1(\xi_s, \eta_s) & \cdots & R_1(\xi_s, \eta_s) R_{n_T}(\xi_s, \eta_s) \\ \vdots & \ddots & \vdots \\ R_{n_T}(\xi_s, \eta_s) R_1(\xi_s, \eta_s) & \cdots & R_{n_T}(\xi_s, \eta_s) R_{n_T}(\xi_s, \eta_s) \end{pmatrix} \begin{pmatrix} z_1 \\ \vdots \\ z_{n_T} \end{pmatrix} = \sum_{s \in \mathcal{S}} \bar{z}_s \begin{pmatrix} R_1(\xi_s, \eta_s) \\ \vdots \\ R_{n_T}(\xi_s, \eta_s) \end{pmatrix} \quad (26)$$

which represents a classic linear least square problem and can be easily solved for the n_T unknowns $\boldsymbol{\lambda} = \{z_1, \dots, z_{n_T}\}$ by proper choice of collocation points.

5.2. Non-linear least-square approximation using non-isoparametric GNURBS

In order to improve the accuracy of the above discussed NURBS-based approximation, we develop a non-linear least-square minimization algorithm using 1st GNURBS. Invoking the non-isoparametric GNURBS surface with partial decoupling of the weights in Section 3, we can treat the out of plane weights w_L^z as extra design variables without perturbing the geometry or parameterization of the underlying precise planar surface $\mathbf{S}_{xy}(\xi, \eta)$. We may refer to these variables as *control weights* hereafter.

The objective function to be minimized could still be written as (23). However, the vector of design variables now changes to $\boldsymbol{\lambda}' = \{z_1, \dots, z_{n_T}, w_1^z, \dots, w_{n_T}^z\}$. Moreover, the following bounding constraints on control weights are often desired to be satisfied for numerical stability.

$$w_k^z > 0, \quad k = 1, \dots, n_T \quad (27)$$

Eq. (23) with the new vector of design variables $\boldsymbol{\lambda}'$ establishes a non-linear least-square optimization problem which could be solved using different existing algorithms. Some of these algorithms, such as Levenberg-Marquardt, do not allow for the imposition of bounding constraints on design variables. In this case, one can easily apply an exponential transformation to control weights to ensure their positivity without the imposition of bounding constraints as in [21]. We will use here the trust-region-reflective algorithm which is available in MATLAB and allows for the imposition of bounding constraints on design variables.

In order to solve the established problem, the Jacobian matrix is required. The Jacobian matrix \mathbf{J} is composed of two parts

$$\mathbf{J} = [\mathbf{J}_z \quad \mathbf{J}_w] \quad (28)$$

where \mathbf{J}_z contains the partial derivatives of f with respect to z_k , while \mathbf{J}_w includes the partial derivatives of f with respect to w_k^z . Differentiating with respect to z_k , \mathbf{J}_z will be easily derived as

$$\mathbf{J}_z = \begin{bmatrix} R_1(\xi_1, \eta_1) & R_2(\xi_1, \eta_1) & \cdots & R_{n_T}(\xi_1, \eta_1) \\ R_1(\xi_2, \eta_2) & R_2(\xi_2, \eta_2) & \cdots & R_{n_T}(\xi_2, \eta_2) \\ \vdots & \vdots & \ddots & \vdots \\ R_1(\xi_{n_c}, \eta_{n_c}) & R_2(\xi_{n_c}, \eta_{n_c}) & \cdots & R_{n_T}(\xi_{n_c}, \eta_{n_c}) \end{bmatrix} \quad (29)$$

The other component of the Jacobian matrix can be obtained as

$$\mathbf{J}_w = \begin{bmatrix} \frac{\partial \hat{z}(\xi_1, \eta_1)}{\partial w_1^z} & \cdots & \frac{\partial \hat{z}(\xi_1, \eta_1)}{\partial w_{n_T}^z} \\ \vdots & \ddots & \vdots \\ \frac{\partial \hat{z}(\xi_{n_c}, \eta_{n_c})}{\partial w_1^z} & \cdots & \frac{\partial \hat{z}(\xi_{n_c}, \eta_{n_c})}{\partial w_{n_T}^z} \end{bmatrix} \quad (30)$$

In order to evaluate the partial derivatives with respect to weight design variables, we rewrite $\hat{z}(\xi, \eta)$ as

$$\hat{z}(\xi, \eta) = \frac{\mathcal{Z}(\xi, \eta)}{\mathcal{W}(\xi, \eta)} \quad (31)$$

where $\mathcal{Z}(\xi, \eta)$ and $\mathcal{W}(\xi, \eta)$ are

$$\mathcal{Z}(\xi, \eta) = \sum_{L=1}^{n_T} N_L(\xi, \eta) w_L^z z_L \quad (32)$$

$$\mathcal{W}(\xi, \eta) = \sum_{L=1}^{n_T} N_L(\xi, \eta) w_L^z \quad (33)$$

Using these definitions, we can obtain

$$\frac{\partial \hat{z}(\xi, \eta)}{\partial w_k^z} = \frac{N_k(\xi, \eta) w_k^z}{\mathcal{W}(\xi, \eta)} [z_k - \hat{z}(\xi, \eta)], \quad k = 1, \dots, n_T \quad (34)$$

Having the analytical Jacobian matrix components in Eqs. (29) and (30), we can now solve the established non-linear least-square optimization problem efficiently. We impose the initial conditions by setting all the control variables to zero and all the control weights to 1, i.e.

$$\lambda'_0 = \left\{ \underbrace{0, 0, \dots, 0}_{n_T}, \underbrace{1, 1, \dots, 1}_{n_T} \right\} \quad (35)$$

5.3. Non-linear least-square approximation using isoparametric GNURBS

Since the derivation of analytical Jacobian matrix with this generalization becomes complicated in case of having internal knots, we limit our derivation here to GR-Bézier. Invoking the isoparametric GR-Bézier representation in Eq. (17), we can again establish the approximation problem as a non-linear least square problem with the objective function defined in Eq. (23) but with the new set of design variables $\lambda'' = \{Z_1, \dots, Z_{n_T}, w_1^z, \dots, w_{n_d}^z\}$ where $n_d = (r+1) \times (s+1)$ is the total number of isoparametric control weights, and $n_T = (\hat{p}+1) \times (\hat{q}+1)$ is the total number of control points. The Jacobian matrix \mathbf{J} can again be divided into two components where \mathbf{J}_z contains the partial derivatives of f with respect to Z_k , while \mathbf{J}_w includes the partial derivatives of f with respect to w_l^z . Differentiating with respect to Z_k , \mathbf{J}_z will be easily derived as

$$\mathbf{J}_z = \begin{bmatrix} \hat{R}_1(\xi_1, \eta_1) & \hat{R}_2(\xi_1, \eta_1) & \cdots & \hat{R}_{n_T}(\xi_1, \eta_1) \\ \hat{R}_1(\xi_2, \eta_2) & \hat{R}_2(\xi_2, \eta_2) & \cdots & \hat{R}_{n_T}(\xi_2, \eta_2) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{R}_1(\xi_{n_c}, \eta_{n_c}) & \hat{R}_2(\xi_{n_c}, \eta_{n_c}) & \cdots & \hat{R}_{n_T}(\xi_{n_c}, \eta_{n_c}) \end{bmatrix} \quad (36)$$

Also, \mathbf{J}_w can be obtained as

$$\mathbf{J}_w = \begin{bmatrix} \frac{\partial \hat{z}(\xi_1, \eta_1)}{\partial w_1^z} & \cdots & \frac{\partial \hat{z}(\xi_1, \eta_1)}{\partial w_{n_d}^z} \\ \vdots & \ddots & \vdots \\ \frac{\partial \hat{z}(\xi_{n_c}, \eta_{n_c})}{\partial w_1^z} & \cdots & \frac{\partial \hat{z}(\xi_{n_c}, \eta_{n_c})}{\partial w_{n_d}^z} \end{bmatrix} \quad (37)$$

In order to evaluate the partial derivatives of $\hat{z}(\xi)$ with respect to isoparametric control weights w_l^z , we rewrite $\hat{z}(\xi)$ as

$$\hat{z}(\xi) = \frac{\mathcal{Z}(\xi)}{\mathcal{W}(\xi)} \quad (38)$$

where $\mathcal{Z}(\xi)$ and $\mathcal{W}(\xi)$ are

$$\mathcal{Z}(\xi) = \sum_{L=1}^{n_r} B_L(\xi, \eta) W_L Z_L \quad (39)$$

$$\mathcal{W}(\xi) = \sum_{L=1}^{n_r} B_L(\xi, \eta) W_L \quad (40)$$

With these definitions, we can obtain the required derivatives as

$$\frac{\partial \hat{z}(\xi, \eta)}{\partial w_l^z} = \frac{1}{\mathcal{W}(\xi, \eta)} \left[\frac{\partial \mathcal{Z}(\xi, \eta)}{\partial w_l^z} - \hat{z}(\xi, \eta) \frac{\partial \mathcal{W}(\xi, \eta)}{\partial w_l^z} \right], \quad l = 1, \dots, n_d \quad (41)$$

The derivatives in above equation can be evaluated using the following expressions

$$\frac{\partial \mathcal{Z}(\xi, \eta)}{\partial w_l^z} = \sum_{L=1}^{n_r} B_L(\xi, \eta) \frac{\partial W_L}{\partial w_l^z} Z_L \quad (42)$$

$$\frac{\partial \mathcal{W}(\xi, \eta)}{\partial w_l^z} = \sum_{L=1}^{n_r} B_L(\xi, \eta) \frac{\partial W_L}{\partial w_l^z} \quad (43)$$

where

$$\frac{\partial W_L}{\partial w_l^z} = \frac{\partial W_{ij}}{\partial w_{mn}^z} = \begin{cases} \lambda_{i,i-m}^{p,r} \lambda_{j,j-n}^{q,s} w_{i-m,j-n}^{xy}, & \text{if } (i-p) \leq m \leq i \text{ \& } (j-q) \leq n \leq j \\ 0 & \text{otherwise} \end{cases} \quad (44)$$

in which

$$\lambda_{i,i-m}^{p,r} = \frac{\binom{p}{i-m} \binom{r}{m}}{\binom{p+r}{i}}, \quad \lambda_{j,j-n}^{q,s} = \frac{\binom{q}{j-n} \binom{s}{n}}{\binom{q+s}{j}}.$$

Similar to previous case, we specify the initial conditions as follows

$$\lambda_0'' = \left\{ \underbrace{0, 0, \dots, 0}_{n_r}, \underbrace{1, 1, \dots, 1}_{n_d} \right\} \quad (45)$$

As previously discussed, by changing w_l^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 result in the best approximation for a given in-plane parameterization. To our knowledge, no such investigation has been reported in the literature thus far.

6. Numerical examples

In this section, we present a few numerical examples of approximating various types of surfaces using the proposed generalizations of NURBS and compare the obtained results with those of classic NURBS. In all cases, the relative L^2 -norm of error is calculated using the following relation

$$error = \frac{\left(\int_{\Omega} (\hat{z}(\xi, \eta) - z(\xi, \eta))^2 d\Omega \right)^{1/2}}{\int_{\Omega} z(\xi, \eta) d\Omega} \quad (46)$$

where all integrations are calculated using Gaussian quadrature.

6.1. Test case 1: helicoid modelling

As the first numerical example, we consider the approximation of a partial helical surface with the following equation:

$$\mathbf{S}(\xi, \eta) = ((\eta + 1) \cos(\xi), (\eta + 1) \sin(\xi), \xi), \quad \begin{array}{l} 0 \leq \xi \leq \pi/2 \\ 0 \leq \eta \leq 1 \end{array} \quad (47)$$

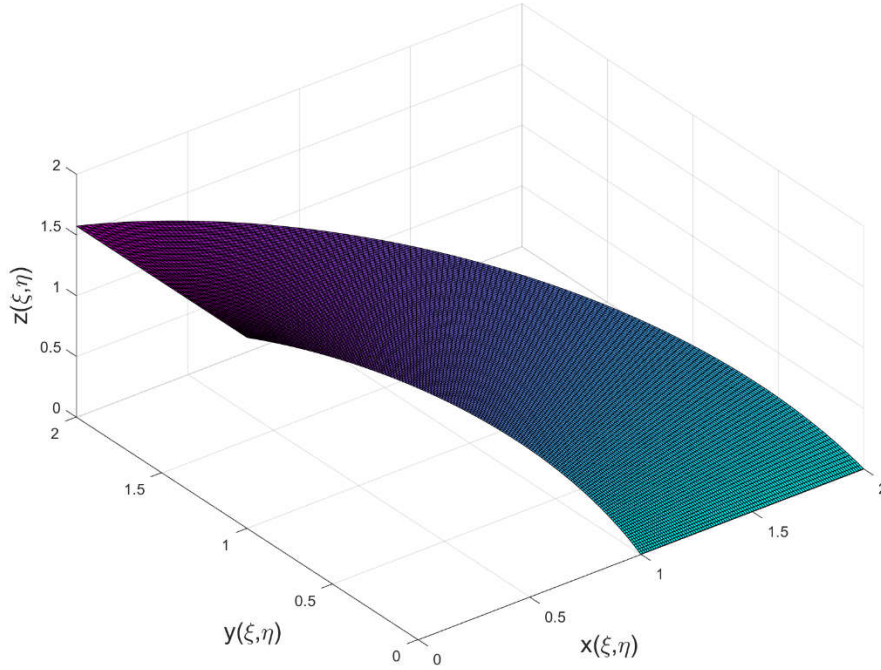


Figure 11. The helical surface in Eq. (47).

As observed, the in-plane parameterization of this surface is a quarter annulus with the configuration already shown in Figure 7. Since this is a geometric modelling problem where preserving the properties of NURBS are of interest, it is an ideal candidate for employing isoparametric GNURBS. Accordingly, following the procedure discussed above in Section 5.1, we try to approximate the given height function in Eq. (47) and compare the obtained results with classic NURBS. The obtained results for different degrees of basis functions are presented in Table 2.

Table 2. Error of approximating the height function of helical surface in Eq. (47) using R-Bézier versus isoparametric GR-Bézier in relative L^2 -norm.

Surface type	Degree (\hat{p}, \hat{q}) $= (p + r, q + s)$	No. of control variables	No. of control weights	Error	Error ratio
R-Bézier	(2,1)	6	0	2.68E-2	1.0
2 nd GR-Bézier			0	2.68E-2	
R-Bézier	(3,2)	12	0	1.28E-4	1.0
2 nd GR-Bézier			4	1.28E-4	
R-Bézier	(4,3)	20	0	1.28E-4	109.4

2 nd GR-Bézier			9	1.17E-6	
R-Bézier	(5,4)	30	0	2.22E-6	180.5
2 nd GR-Bézier			16	1.23E-8	

According to this table, by including larger numbers of control weights, better improvement of accuracy is achieved. This reveals superior approximation of rational functions especially when higher degrees of basis functions are employed. The results, however, show no improvement in accuracy for the first level of degree elevation, i.e. $(r,s) = (1,1)$. This implies that the optimal values of control weights for this particular level of degree elevation are unity. In other words, classic order elevation results in optimal accuracy for the approximation of helical height function using this particular degree of basis functions.

6.2. Test case 2: Scherk minimal surface

As the second numerical experiment, we consider the construction of a minimal surface model referred to as Scherk minimal surface over a square domain. This example has been addressed by Pan et al. [27] using isogeometric analysis of minimal surfaces based on extended loop subdivision scheme. The equation of this minimal surface is given as

$$\mathbf{S}(\xi, \eta) = \left(\xi, \eta, \ln \left(\frac{\cos(\eta)}{\cos(\xi)} \right) \right), \quad \begin{matrix} -1.5 \leq \xi \leq 1.5 \\ -1.5 \leq \eta \leq 1.5 \end{matrix} \quad (48)$$

which is depicted in Figure 12.

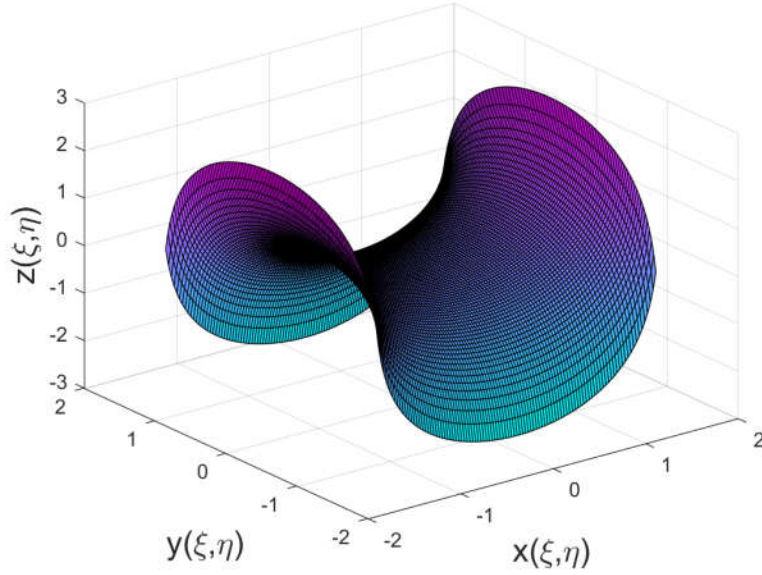


Figure 12. Scherk minimal surface.

As the figure shows, the surface features steep gradients near the boundaries. In this example, for simplicity, we use non-isoparametric GNURBS and compare its approximation properties with classic NURBS. The obtained results using various employed degrees of basis functions are shown in Table 3. As observed, the accuracy of approximation using 1st GNURBS in all cases is better than that of classic NURBS. Further, the increase in accuracy substantially improves when larger degrees of basis functions are used.

Table 3. Error of approximating the Scherk minimal surface in Eq. (48) using NURBS versus 1st GNURBS in relative L^2 -norm.

Surface type	Degree (p, q)	No. of control variables	No. of control weights	Error	Error ratio
NURBS	(2, 2)	25	0	1.52E-1	18.76
1 st GNURBS			25	8.11E-3	
NURBS	(3, 3)	36	0	9.40E-2	18.42
1 st GNURBS			36	5.10E-3	
NURBS	(4, 4)	49	0	5.02E-2	142.21
1 st GNURBS			49	3.53E-4	
NURBS	(5, 5)	64	0	3.59E-2	262.04
1 st GNURBS			64	1.37E-4	

Moreover, for the case of quadratic basis functions, we also perform a convergence study where we persistently refine the knot sequence and compare the obtained accuracy of NURBS versus GNURBS. The obtained results are plotted in Figure 13. As the figure shows, the convergence rate of GNURBS is more than order faster than classic NURBS, resulting in substantial improvement of accuracy especially when larger numbers of control points are used.

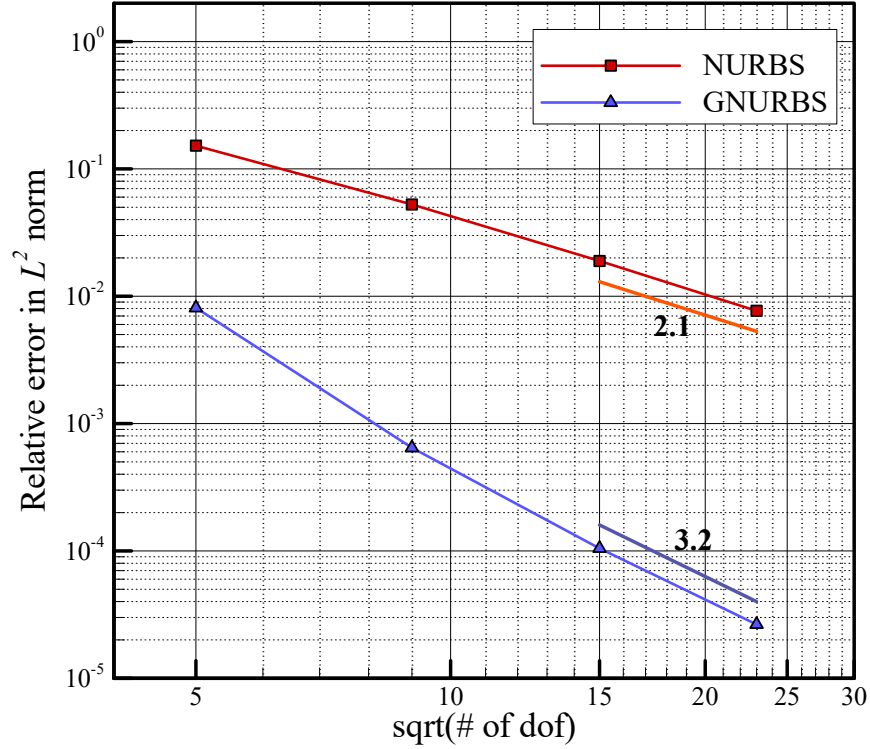


Figure 13. Convergence rate of quadratic NURBS versus GNURBS for the approximation of Scherk minimal surface.

6.3. Test Case 3: Surface of revolution

As the final numerical study, we consider the problem of the approximation of a surface of revolution defined using Eq. (49), which is depicted in Figure 14.

$$\mathbf{S}(\xi, \eta) = \left((\eta + 1) \cos(\xi), (\eta + 1) \sin(\xi), e^{\eta^2} \right), \quad \begin{array}{l} 0 \leq \xi \leq 2\pi \\ 0 \leq \eta \leq 1 \end{array} \quad (49)$$

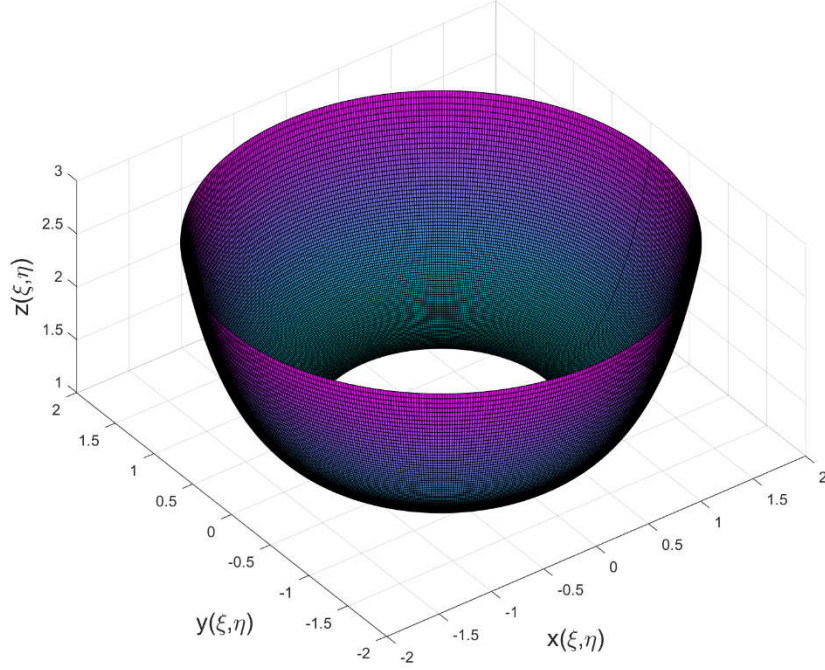


Figure 14. The surface of revolution in Eq. (49).

As observed, the surface has an exponential behavior along the radial direction. In this example, we demonstrate how employing the second proposed variation of NURBS could be useful for improved approximation of these type of surfaces using the same number of control points. For simplicity, we only consider modelling a quarter of the surface, i.e. $(0 \leq \xi \leq \pi/4)$. Similar to the first numerical example, we start with the initial model of degree $(p, q) = (2, 1)$ in Figure 8. Since the height function here only varies along the radial direction, we only elevate the degree along this direction (η) and compare the obtained approximation results using Bézier (classic order elevation) with those of isoparametric GR-Bézier (optimal order elevation). The obtained results for $(r, s) = (0, 0)$ up to $(r, s) = (0, 3)$ are presented in Table 4.

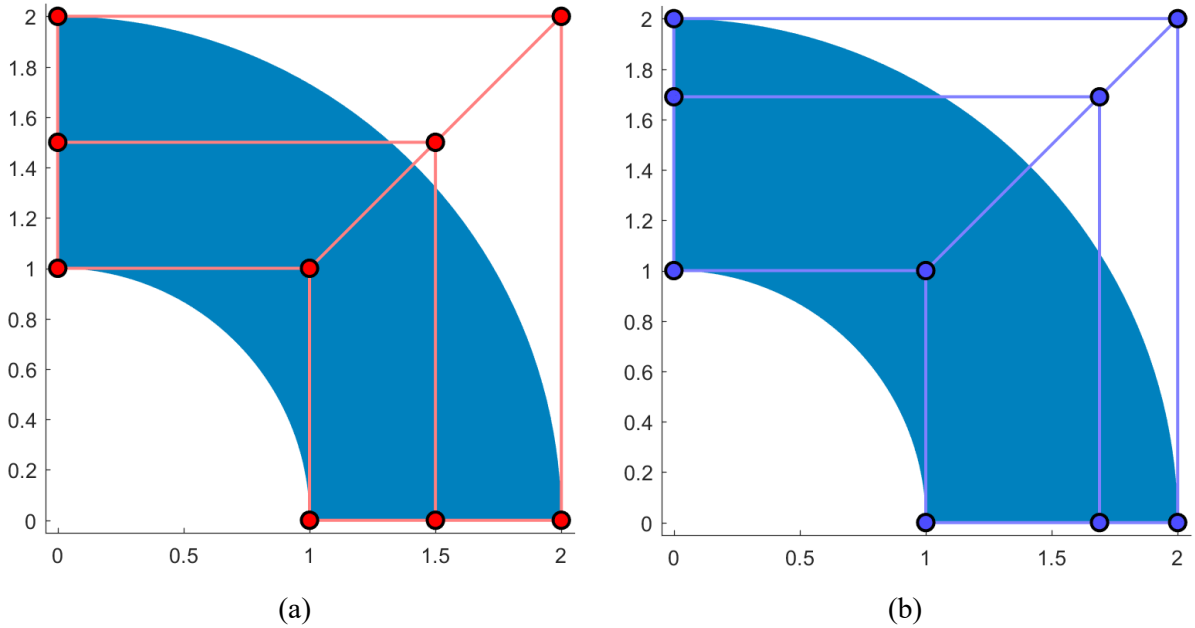
Table 4. Error of approximating the height function of the surface of revolution in Eq. (49) using R-Bézier versus isoparametric GR-Bézier in relative L^2 -norm.

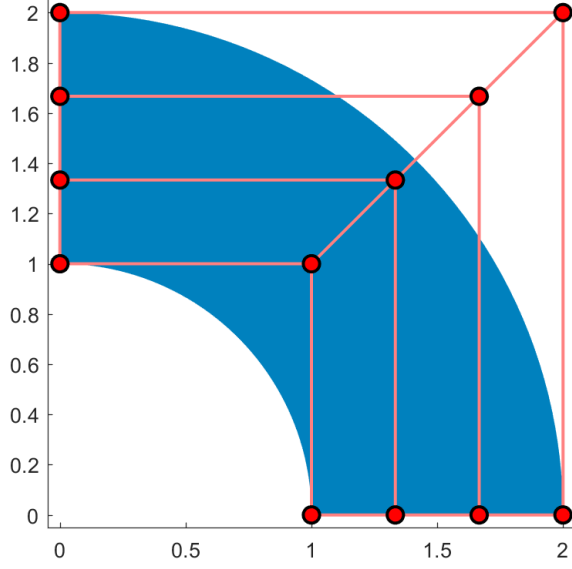
Case No.	Surface type	Degree (\hat{p}, \hat{q}) $= (p + r, q + s)$	No. of control variables	No. of control weights	Error	Error ratio
1	R-Bézier	(2,1)	6	0	0.20E0	1.0
2	2 nd GR-Bézier			0	0.20E0	

3	R-Bézier	(2,2)	12	0	3.42E-2	45.25
4	2 nd GR-Bézier			4	7.55E-4	
5	R-Bézier	(2,3)	20	0	7.10E-3	43.58
6	2 nd GR-Bézier			9	1.63E-4	
7	R-Bézier	(2,4)	30	0	1.12E-3	1.26E4
8	2 nd GR-Bézier			16	8.90E-8	

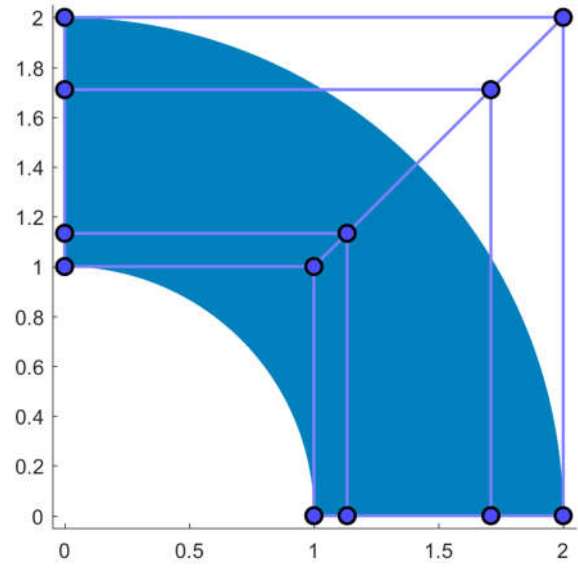
According to this table, the accuracy of approximation by using isoparametric GR-Bézier is significantly higher than that of classic Bézier, especially when higher order elevations are applied. These results clearly show the superiority of rational functions for the approximation of this class of surfaces.

Finally, the corresponding arrangements of control points for cases 3 to 8 are represented in Figure 15. As observed, the arrangements of control points in all cases only differ along the radial direction. This was expected to be the case, since in this example, order elevation has only been performed along the radial direction.

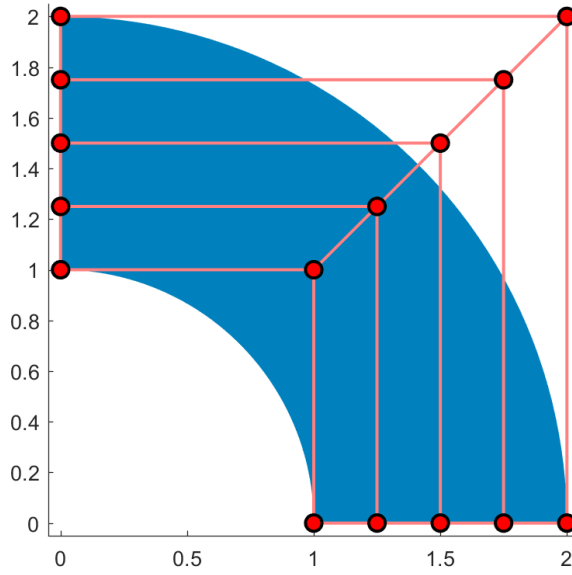




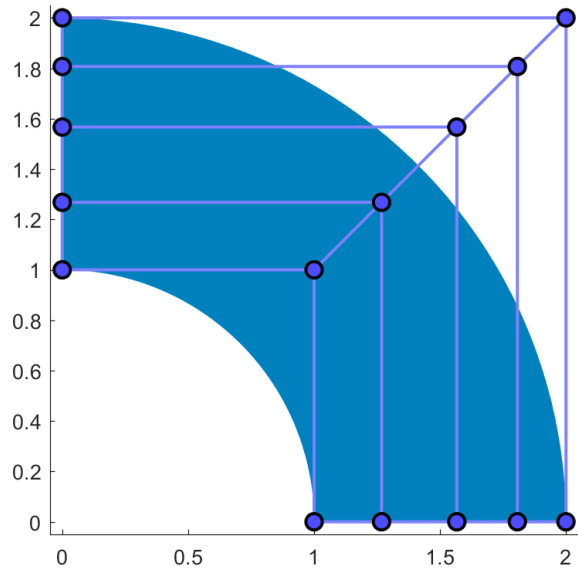
(c)



(d)



(e)



(f)

Figure 15. The resulting control net for the approximation of the surface of revolution in Eq. (49): (a) Case 3, (b) Case 4, (c) Case 5, (d) Case 6, (e) Case 7, and (f) Case 8.

7. Extensions and further applications

While, in this paper, we limited our study to applying the proposed generalizations to NURBS, due to fundamental similarities between different variations of splines, similar generalizations

seem plausible to other rational forms of splines such as T-spline surfaces, Tri-angular Bézier surfaces etc.

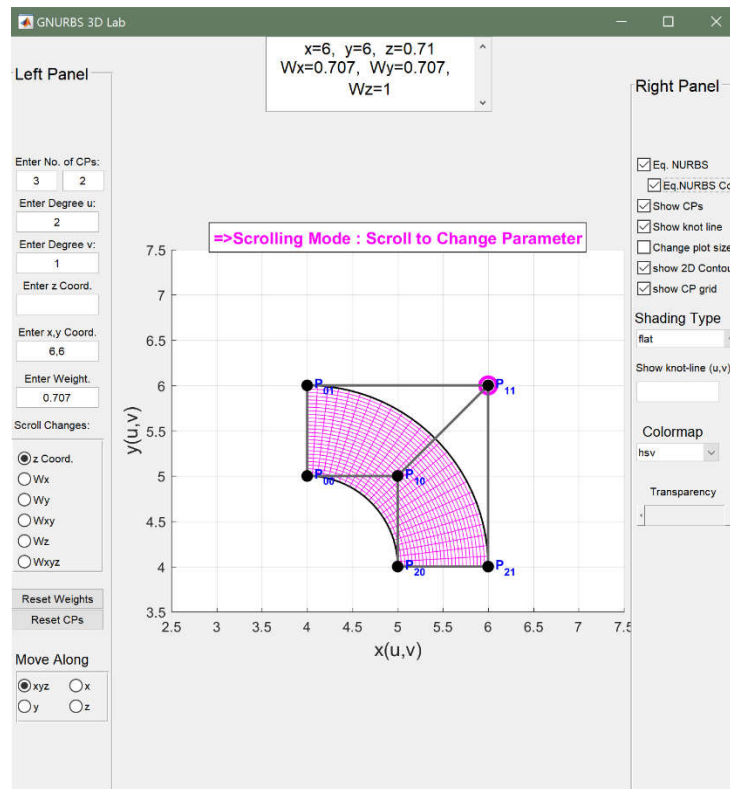
In addition to the discussed applications of GNURBS in CAGD, other applications of NURBS in this area can be found where employing the weights as additional design variables for better flexibility can be problematic or sometimes impossible. For instance, GNURBS may also help circumventing the difficulties of considering the weights as degrees of freedom in general surface fitting problems with arbitrary parameterization. As previously studied in [13, 23], employing the weights as additional degrees of freedom in data approximation can deteriorate the surface parameterization, and lead to undesirable results, especially when approximating rapidly varying data. On the other hand, employing GNURBS, by including the *control weights* as design variables, one can create a good surface parameterization and preserve it during fitting without imposing any restrictions on the magnitude of variations of the weights.

Furthermore, NURBS have been extensively used in other disciplines such as computational mechanics for the optimization of different fields of interest over a given computational domain. Considering these studies, we can find out that in this class of applications, the parameterization of the design domain needs to remain fixed throughout the optimization process; see [8, 28–36], for instance. 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. However, owing to the proposed GNURBS representations with decoupled weights, one can now treat the control weights as additional design variables while setting up the optimization problem and still preserve the underlying geometry as well as its parameterization. As elaborated in this research, this can lead to significant improvement in the obtained accuracy in both cases of smooth as well as rapidly varying fields. Exploring some of these applications is the subject of our future studies.

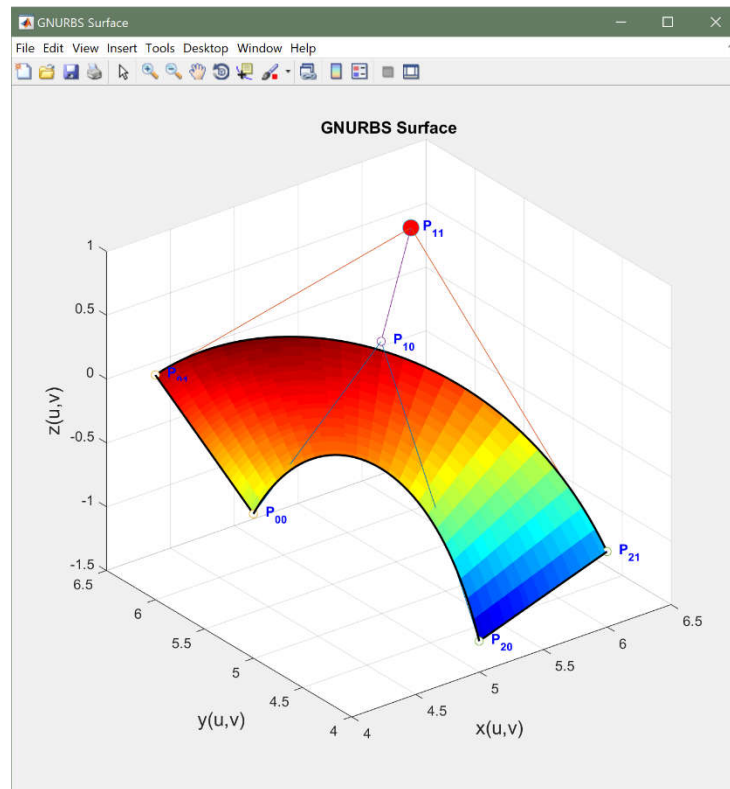
8. MATLAB toolbox: GNURBS3D-Lab

In order to facilitate understanding the behavior of GNURBS surfaces and the additional abilities they serve, a comprehensive and fully interactive MATLAB toolbox, named *GNURBS3D-Lab*, has been developed. This toolbox is developed via the extension of *GNURBS Lab*, a similar interactive MATLAB toolbox already developed for GNURBS curves [2]. Snapshots of different

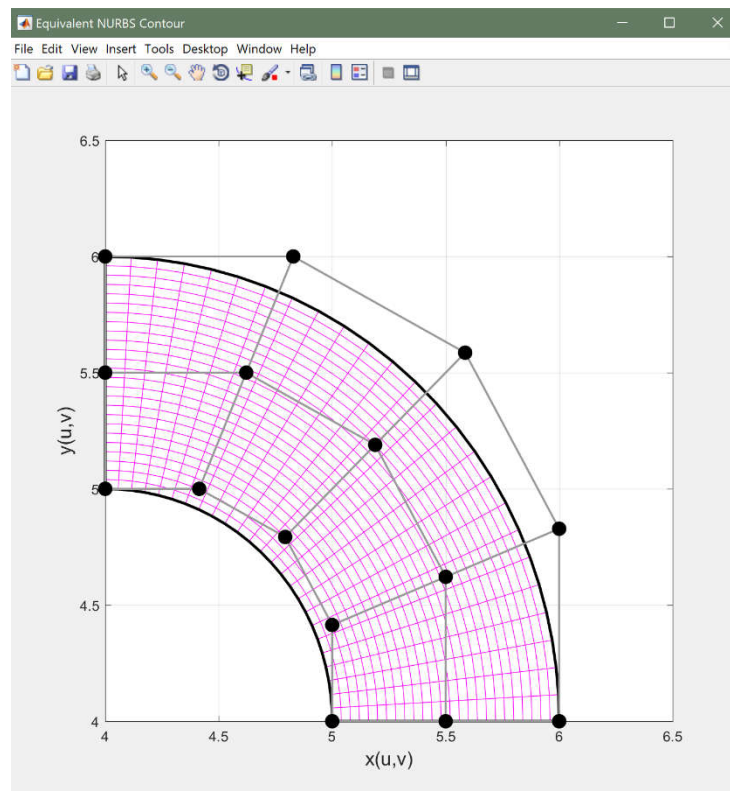
available windows in *GNURBS3D-Lab* are shown in Figure 16, which demonstrate the environment of the toolbox and numerous features that the software provides.



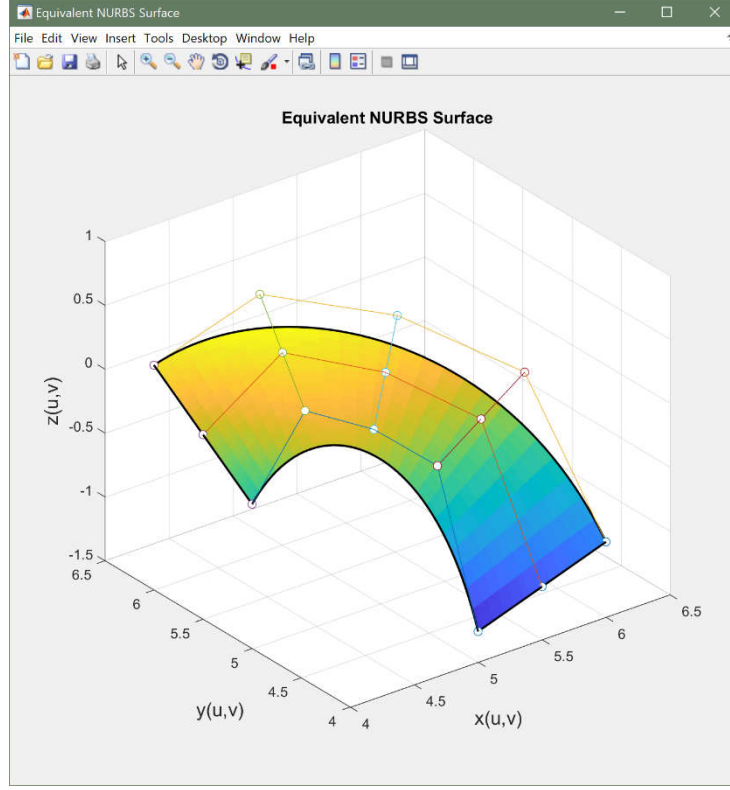
(a)



(b)



(c)



(d)

Figure 16. Snapshots of different windows of GNURBS3D-Lab: (a) Main window, (b) 3D surface plot window, (c) in-plane equivalent NURBS window, and (d) 3D equivalent NURBS window.

The figure shows an example of designing a 3D surface with an in-plane shape of a quarter annulus and a free-form out of plane shape using GR-Bézier. As demonstrated in Figure 16, the toolbox is enabled to evaluate the equivalent higher-order rational Bézier representations with the designed surface in 2D and 3D interactively. Employing the provided wide range of tools shown in Figure 16(a), one can easily manipulate any defining parameter of the surface, including the locations of control points, or a variety of weight components, and observe the changes interactively in all four windows shown in Figure 16, simultaneously.

The open-source toolbox is available at <http://www.ersl.wisc.edu/software/GNURBS3D-Lab.zip>. Detailed instructions for using this toolbox are also provided in an additional document *GNURBS3D_Manual.pdf* accessible via the same link.

9. Conclusion

We introduced two generalizations of NURBS surfaces, referred to as GNURBS, by decoupling of the weights associated with the control points along different physical coordinates. These

generalizations were obtained via either explicit or implicit decoupling of the weights leading to non-isoparametric and isoparametric representations, respectively. As demonstrated, both these variations 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. Superior approximation abilities of these variations for both smooth and rapidly varying functions were shown via simple examples in surface modelling. It was shown that GNURBS can be effectively used for improved construction of various types of surfaces such as helicoids, minimal surfaces as well as surfaces of revolution using the same number of control points. A comprehensive MATLAB toolbox, named *GNURBS3D-Lab*, was developed and introduced to better demonstrate the behavior of different types of GNURBS surfaces in a fully interactive manner. In summary, GNURBS were shown to serve as a new effective technology in surface modelling with superior accuracy while merely being disguised forms of classic NURBS.

Acknowledgements

The authors would like to thank the support of National Science Foundation through grant CMMI-1661597. Prof. Suresh also serves as a consulting Chief Scientific Officer for SciArt, Corp.

References

1. Versprille KJ (1975) Computer-aided Design Applications of the Rational B-spline Approximation Form
2. Taheri AH, Abolghasemi S, Suresh K (2019) Generalizations of non-uniform rational B-splines via decoupling of the weights: theory, software and applications. *Eng Comput* 36:1831–1848. <https://doi.org/10.1007/s00366-019-00799-w>
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 (2019) NURBS-augmented finite element method for stability analysis of arbitrary thin plates. *Eng Comput* 35:351–362. <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 (2019) A framework of shape optimisation based on the isogeometric boundary element method toward designing thin-silicon photovoltaic devices. *Eng Comput* 35:423–449. <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. Dimas E, Briassoulis D (1999) 3D geometric modelling based on NURBS: a review. *Adv Eng Softw* 30:741–751
14. Leal NE, Ortega Lobo O, Branch JW (2007) Improving NURBS Surface Sharp Feature Representation. *Int J Comput Intell Res* 3:131–138. <https://doi.org/10.5019/j.ijcir.2007.97>
15. Bracco C, Giannelli C, Sestini A (2017) Adaptive scattered data fitting by extension of local approximations to hierarchical splines. *Comput Aided Geom Des* 52–53:90–105. <https://doi.org/10.1016/j.cagd.2017.03.008>
16. Forsey DR, Barrels RH (1988) Hierarchical B-spline refinement. *Proc 15th Annu Conf Comput Graph Interact Tech SIGGRAPH 1988* 22:205–212. <https://doi.org/10.1145/54852.378512>

17. 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
18. 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
19. Sederberg, T.N., Zheng, J.M., Bakenov, A., Nasri A (2003) T-splines and T-NURCCSs. *ACM Trans Graph* 22:477–484
20. Thomas D (2019) U-splines: splines over unstructured meshes
21. Carlson N (2009) NURBS Surface Fitting with Gauss-Newton. Lulea University of Technology
22. Ma W (1994) NURBS-based computer aided design modelling from measured points of physical models. Catholic University of Leuven
23. Piegl L (1991) On NURBS: A Survey. *IEEE Comput Graph Appl* 55–71
24. Taheri AH, Suresh K (2020) Adaptive w-refinement: A new paradigm in isogeometric analysis. *Comput Methods Appl Mech Eng* 368:. <https://doi.org/10.1016/j.cma.2020.113180>
25. Farin G (2001) *Curves and Surfaces for CAGD A Practical Guide*, 5th ed. Morgan Kaufmann
26. Piegl L, Tiller W (1995) *The NURBS Book*, 1st ed. Springer-Verlag Berlin Heidelberg
27. Pan Q, Chen C, Xu G (2017) Isogeometric finite element approximation of minimal surfaces based on extended loop subdivision. *J Comput Phys* 343:324–339. <https://doi.org/10.1016/j.jcp.2017.04.030>
28. 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>
29. 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>
30. 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>
31. 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->

32. 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>
33. 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>
34. Liu H, Yang D, Wang X, et al (2019) Smooth size design for the natural frequencies of curved Timoshenko beams using isogeometric analysis. *Struct Multidiscip Optim* 59:1143–1162. <https://doi.org/10.1007/s00158-018-2119-8>
35. 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>
36. 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>