Proceedings of the ASME 2024
International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
IDETC/CIE2024
August 25–28, 2024, Washington, DC

DETC2024-142195

ENHANCING ISOGEOMETRIC ANALYSIS WITH NURBS-BASED SYNTHESIS

Shaoliang Yang[1,†], Kang Wang[2,†], Jun Wang[3,*]

[1]Santa Clara University, Santa Clara, CA

ABSTRACT

Isogeometric analysis (IGA) is a computational technique that integrates computer-aided design (CAD) with finite element analysis (FEA) by employing the same basis functions for both geometry representation and solution approximation. While IGA offers numerous advantages, such as improved accuracy and efficiency, it also presents several challenges related to geometric modeling. Some of these challenges include accurately representing complex geometries with NURBS (Non-Uniform Rational B-Splines) or other basis functions used in IGA and generating high-quality meshes that conform to the complex geometry represented by NURBS curves/surfaces. This paper introduces an analytical framework to provide a more efficient and theoretically grounded method for generating curvilinear configurations and its analytical solution in IGA, bridging the gap between generated data and its physical representations. This innovative approach is distinguished by integrating the NURBS parameterization in curve generation and providing a corresponding framework to achieve a broader and more accurate explanation of meshes and properties, especially constructing new coordinates and calculating the physical displacements under these conditions. Our model enables the analytical understanding of complex curves from the UIUC airfoil and superformula datasets, demonstrating a deeper dive into simulations. This study signifies a pivotal juncture wherein machine-learning-based complex geometrical formulations are synergistically combined with actual isogeometric analysis.

Keywords: Shape Synthesis, Non-Uniform Rational B-Splines, Isogeometric Analysis, Generative Model

1. INTRODUCTION

In the field of mechanical engineering and physics-based design generation [1], the integration and application of NURBS (Non-Uniform Rational B-Splines) [2] have emerged as a pivotal enhancement over conventional geometric modeling techniques such as B-Splines [3] and Bezier Curves [4]. Unlike B-Splines and Bezier Curves, which are predominantly linear and offer limited flexibility, NURBS provide a more advanced framework that allows for the representation of both standard geometric shapes and more complex figures with a higher degree of accuracy and control [5, 6]. This inherent versatility makes NURBS particularly suitable for intricate design tasks in mechanical engineering, where precision and adaptability are paramount [7]. The transition from B-Splines and Bezier to NURBS marks a significant evolution in design capabilities, enabling engineers to craft more sophisticated and nuanced models [8].

Isogeometric Analysis (IGA) [9] has marked a significant advancement in mechanical engineering [10], particularly in the realms of analysis and simulation [11]. By employing NURBS-based geometric models directly from the design phase into various analytical studies, including structural, fluid dynamics, and thermal assessments, IGA fosters a harmonious integration of design and analysis [12]. This methodology circumvents the traditional need for geometry simplification and extensive meshing typical in Finite Element Analysis (FEA) [13], thereby maintaining the integrity of the original design and elevating simulation precision [14].

Despite these strides, bridging the gap between deep-learning-based models and the IGA framework presents a notable challenge [15]. Machine learning integration is transforming industrial and mechanical engineering [16, 17], leading to smarter system designs and improved predictive analytics in manufacturing and maintenance [18–21]. Existing tools like BSplineGAN [4] and BezierGAN [3], despite their innovation in employing adversarial generative networks (GAN) [22] for design generation, primarily yield data at the point level, which falls short of the comprehensive, simulation-ready models required for detailed IGA analysis [9]. Furthermore, their focus on Bezier and B-Splines generation needs to satisfactorily cater to the intricate requirements of NURBS configurations essential for effective IGA implementation [23]. Addressing these issues head-on, our framework endeavors to connect AI-driven NURBS modeling with IGA, promising advancement in the analytical capabilities

[†]Joint first authors
*Corresponding author: jwang22@scu.edu

within mechanical engineering.

However, a significant gap exists in bridging the deep-learning-based NURBS models with the IGA framework efficiently. While advancements such as BSplineGAN [3] and BezierGAN [4] have facilitated automated design generation with adversarial generative network (GAN), which is commonly applied in physics-based modeling [18, 24]. these are primarily focused on data-point level outputs, limiting their direct applicability in systematic simulations required for comprehensive analysis. Additionally, the specific generation of curves for Bezier and B-Splines does not fully address the complex needs of NURBS configurations, which are essential for IGA applications.

Our contribution can be summarized as follows:

- We propose a framework to bridge the gap between generated data with a deep learning model and the isogeometric analysis.
- We introduce an interpolation strategy to automatically map the data points into IGA-suitable coordinates.
- We create an improved isogeometric analysis framework to facilitate the force / location / mesh / displacements analysis.

2. METHODOLOGY

The general pipeline of our proposed framework is shown in Fig. 1. In this section, we will discuss 1) Non-Uniform Rational B-Splines with Deep Learning, 2) Preprocessing of NURBS Parameters for Isogeometric Analysis, 3) NURBS Surface Generation, 4) Refinement Strategy for Knot Vectors in NURBS-based Plane Stress Simulations, 5) Material Constitutive Relations.

2.1 Non-Uniform Rational B-Splines with Deep Learning

To cooperate with the isogeometric analysis approach framework, we apply Non-Uniform Rational B-Splines parameterization with the same generator as BezierGAN [4]. Specifically, we deploy the generator to generate weights and control points, as well as features extracted through the MLP layer [25].

1. *Initialization*: The initialization sets up the parameters necessary for NURBS computations: Number of input features: $[Latent : 3, \ Noise : 10]$, Number of control points: 32, Number of data points (output dimensionality): 192, Degree of the NURBS curve: $degree = 3$, Small constant to prevent division by zero: $\epsilon = 1e-7$, Knot vector: $knots$, computed as follows:

$$knots = [\underbrace{0, \ldots, 0}_{\text{degree + 1}}, \underbrace{\frac{1}{n_{cp} - \text{degree}}, \ldots, \frac{n_{cp} - \text{degree} - 1}{n_{cp} - \text{degree}}}_{n_{controlpoint} - \text{degree}}, \underbrace{1, \ldots, 1}_{\text{degree + 1}}]. \tag{1}$$

2. *Intervals Generation*: The intervals are generated by transforming the input features through a neural network sequence:
   1) Linear Transformation (L): first layer: $\mathbb{R}^{in\_features} \rightarrow \mathbb{R}^{n\_data\_points-1}$ and rest layers: $\mathbb{R}^{n\_data\_points-1} \rightarrow \mathbb{R}^{n\_data\_points-1}$ to bring in more accurate projection.
   2) Activation (S:Softmax): Normalizes the outputs across the dimension.
   3) This process is repeated, ending with a padding operation to ensure the correct dimensionality:

$$\begin{aligned} intervals = \ & (S(L(S(L(S(L(input))))))), \\ & ConstantPad1d([1,0], 0). \end{aligned} \tag{2}$$

3. *Basis Function Computation*: The basis functions, $N_{i,p}(t)$, are defined recursively. For a knot vector $\{knots\}$ and a degree p:
   1) For $p = 0$ (base case):

$$N_{i,0}(t) = \begin{cases} 1 & \text{if } knots_i \leq t < knots_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

   2) For $p > 0$ (recursive case):

$$\begin{aligned} N_{i,p}(t) = & \frac{t - knots_i}{knots_{i+p} - knots_i + \epsilon} N_{i,p-1}(t) + \\ & \frac{knots_{i+p+1} - t}{knots_{i+p+1} - knots_{i+1} + \epsilon} N_{i+1,p-1}(t). \end{aligned} \tag{4}$$

4. *Forward Pass*: During the forward pass, the NURBS curve is computed from the input tensor and control points:
   1) Compute the parameter intervals from the input features.
   2) Calculate the upper bounds (ub) by cumulatively summing the intervals and clamping the result between 0 and 1.
   3) Compute the NURBS basis functions for each control point across all parameter values:

$$\begin{aligned} N = [& N_{0,p}(ub), N_{1,p}(ub), \ldots, \\ & N_{n\_control\_points-1,p}(ub)]. \end{aligned} \tag{5}$$

   4) Combine the control points **CP** with their corresponding weights **W** and apply the NURBS basis functions to compute the curve:

$$\mathbf{DP} = \frac{\sum_{j=0}^{n\_control\_points-1} N_{j,p}(ub) \times (\mathbf{CP}_j \times \mathbf{W}_j)}{\sum_{j=0}^{n\_control\_points-1} N_{j,p}(ub) \times \mathbf{W}_j + \epsilon}. \tag{6}$$

   where **DP** represents the data points of the NURBS curve, which are the output of this layer. The weights **W** are applied to control points **CP**, and the resulting weighted points are combined according to the NURBS basis functions $N_{j,p}(ub)$ to form the curve. The generated samples are shown in Fig. 2.

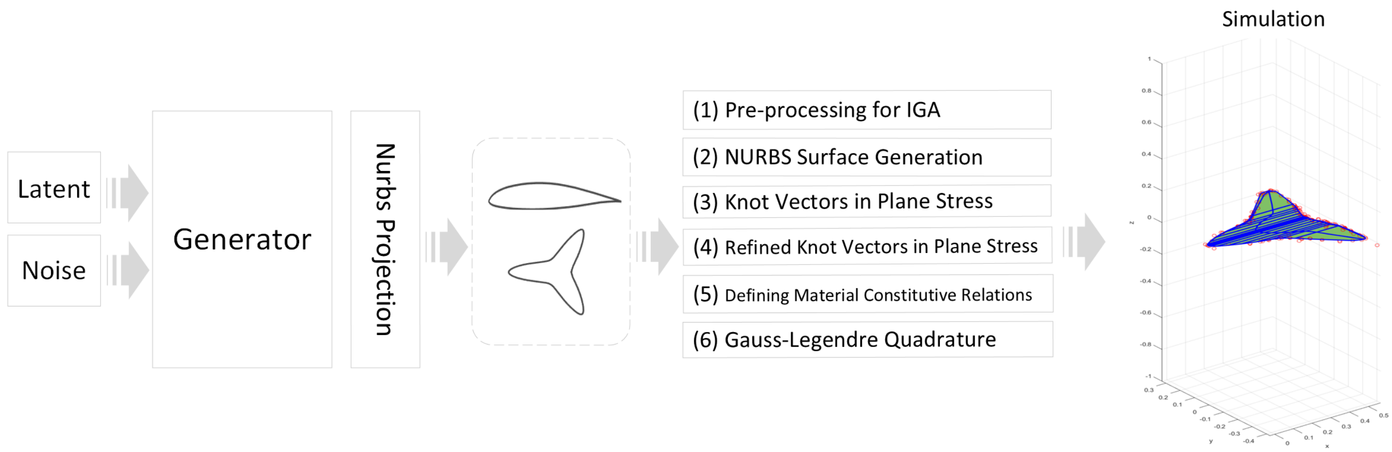

FIGURE 1: Pipeline of overall framework

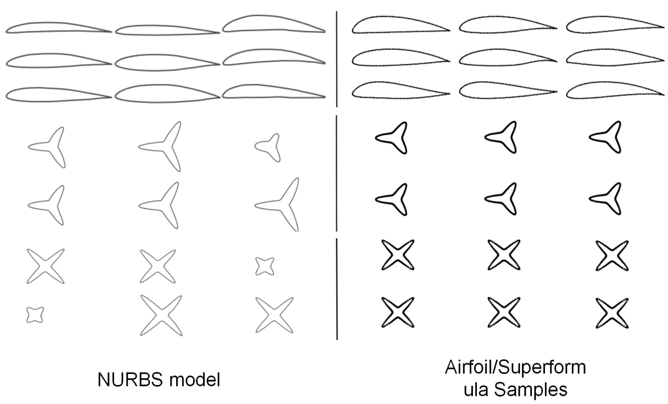

FIGURE 2: Generated samples of Airfoil/Superformula.

2.2 Preprocessing of NURBS Parameters for Isogeometric Analysis

The preprocessing phase in the isogeometric analysis is instrumental for transitioning raw NURBS parameters, including control points and associated weights, into a sophisticated matrix configuration, vital for further analytical evaluations [9, 12]. Below, we delineate this procedure through a more quantitative and mathematical lens.

1. *Initial Configuration and Sorting*: Consider the preliminary set of control points and their respective weights denoted by:

$$\{(x_i, y_i) | i = 1, 2, ..., n\}, \tag{7}$$

$$\{w_i | i = 1, 2, ..., n\}. \tag{8}$$

where n symbolizes the total number of points. The preliminary step involves organizing these points based on ascending $y-$coordinates, which is mathematically depicted as sorting the set $\{(x_i, y_i)\}$ based on the values of y_i.

2. *Enhanced Interpolation Mechanism*: The interpolation strategy is elaborated through the following mathematical formulations:

   1) Evaluation of Required Points: For adjacent control points P_i and P_{i+1}, the necessity for intermediate points is assessed based on disparities in their coordinates and weights.
   2) Computational Differences: These differences are mathematically quantified as:

$$\text{Y-coordinate disparity:} \quad \Delta y = y_{i+1} - y_i. \tag{9}$$

$$\text{X-coordinate disparity:} \quad \Delta x = x_{i+1} - x_i. \tag{10}$$

$$\text{Mean weight formulation:} \quad w_{avg} = \frac{w_i + w_{i+1}}{2}. \tag{11}$$

   3) Actual Interpolation Sequence: The program will first identify the gaps in y coordinates. For each gap, the program will iterate through its length, adding y values and interpolating x and w values. For example, if there is a gap of 5 between the second and third y coordinates, then the interpolation for the third variable is computed as follows:

$$y_{new} = y1[3 + i], \tag{12}$$

$$x_{\text{new}} = x_i + (\Delta x) \times, \frac{3}{5} \tag{13}$$

$$w_{new} = w_{avg}. \tag{14}$$

3. *Detailed Construction of NURBS Matrix (B)*: The assembly of matrix B is formulated as follows:

   1) Initial Matrix Setup:

$$B_{ij} = [x, y, z, w] \text{ for } i, j = 1, 2, ..., n, \tag{15}$$

   where typically z = 0 for planar surfaces.

   2) Boundary Determination: At the lower boundary (first row of B):

$$B[0, j] = [x_{min}, y_{min}, 0, w_{min}]. \tag{16}$$

   At the upper boundary (last row of B):

$$B[n-1, j] = [x_{max}, y_{max}, 0, w_{max}]. \tag{17}$$

3) Matrix Population: Intermediate entries of B for each distinct y-coordinate y_i:

$$B[i, j] = [x_i, y_i, 0, 1]. \tag{18}$$

ensuring linear variation in x from x_{min} to x_{max} across each row.

4) Normalization and Adaptation: Final adjustments to weights and coordinates based on defined criteria to ensure uniformity and adherence to NURBS standards.

Post-processing adjustments ensure the refined matrix B mirrors the NURBS surface precisely, suitable for isogeometric applications. This entails confirming the orderly alignment of control points and standardizing weights to uphold geometric authenticity.

2.3 Detailed Mathematical and Algorithmic Approach for NURBS Surface Generation

The generation of NURBS surfaces forms the foundation of our computational framework, the original model was outlined in [26], which was initially intended for analyzing object deformations, this base model has been adapted to develop an enhanced Isogeometric Analysis (IGA) framework. This section elaborates on each step involved in this process, utilizing equations to elucidate the methodologies employed. The framework is shown in Fig. 3.

The NURBS surface, $S(u, v)$, is defined by the equation:

$$S(u, v) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(u) M_{j,q}(v) w_{ij} P_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(u) M_{j,q}(v) w_{ij}}, \tag{19}$$

where $N_{i,p}(u)$ and $M_{j,q}(v)$ are the B-spline basis functions of degrees p and q, respectively. w_{ij} are the weights associated with each control point. P_{ij} are the control points. n and m are the numbers of control points in the u and v directions, respectively.

2.3.1 B-Spline Basis Functions. The zero-degree B-spline basis functions [27] are defined as:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } U_i \le u < U_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

The higher-degree basis functions are constructed using the recurrence relation:

$$N_{i,k}(u) = \frac{u - U_i}{U_{i+k} - U_i} N_{i,k-1}(u) + \frac{U_{i+k+1} - u}{U_{i+k+1} - U_{i+1}} N_{i+1,k-1}(u), \tag{21}$$

where U is the knot vector.

2.3.2 FindSpan Algorithm. This algorithm locates the span i within the knot vector U that contains the parameter value u, essential for evaluating the basis functions:

$$FindSpan(n, p, u, U) = \begin{cases} n - 1 & if\, u = U_{n+1} \\ binarysearchresult & otherwise \end{cases} \tag{22}$$

The FindSpan algorithm employs a binary search to identify the correct knot span index for a given parameter value u within the knot vector U. The search iteratively narrows down the range by comparing u against the midpoint values of the knot vector. Initially set with boundaries low and $high$, the algorithm recalculates the midpoint, $mid = \left\lfloor \frac{high+low}{2} \right\rfloor$, in each iteration. If u is less than the midpoint value $U[mid]$, the search range narrows to the lower half by setting $high = mid$; otherwise, it narrows to the upper half by setting $low = mid$. This process repeats until the high and low indices converge, at which point mid represents the span index that encloses u, which is then returned as the result.

2.3.3 BasisFun Algorithm. The BasisFun algorithm is designed to compute the non-zero B-spline basis functions corresponding to a given parameter value u and a specific knot span i. This is integral for the evaluation of NURBS surfaces at any given point in their domain. The procedure is detailed in Algorithm 1.

Algorithm 1 Calculate the B-spline basis functions

Input: i, u, p, U
Output: N (array of size $p + 1$)
Initialize $N[0] \leftarrow 1$
Initialize $left$ and $right$ arrays of size $p + 1$ with zeros
for $j = 1$ to p **do**
  $left[j + 1] \leftarrow u - U[i + 1 - j]$
  $right[j + 1] \leftarrow U[i + j] - u$
  $saved \leftarrow 0$
  for $r = 0$ to $j - 1$ **do**
    $temp \leftarrow \frac{N[r+1]}{right[r+2]+left[j-r+1]}$
    $N[r + 1] \leftarrow saved + right[r + 2] \times temp$
    $saved \leftarrow left[j - r + 1] \times temp$
  end for
  $N[j + 1] \leftarrow saved$
end for

2.3.4 Final Evaluation of the NURBS Surface Point. With the FindSpan and BasisFun algorithms, we compute the NURBS surface point $S(u, v)$ for each parameter pair (u, v) as follows:

$$S(u, v) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(u) \cdot M_{j,q}(v) \cdot w_{ij} \cdot P_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(u) \cdot M_{j,q}(v) \cdot w_{ij}}. \tag{23}$$

This comprehensive mathematical framework and the accompanying algorithmic procedures are the cornerstone for generating and evaluating NURBS surfaces, integral to numerous engineering and graphics applications. The NURBS generation is shown in Fig. 4.

2.4 Refinement Strategy for Knot Vectors in NURBS-based Plane Stress Simulations

Refining the knot vector (ξ) is crucial for enhancing the resolution of a NURBS model, particularly relevant in plane stress simulations, as detailed by the recent research [28]. This process, depicted in Fig. 5 and Fig. 6, involves calculating and incorporating new knot values to improve the model's granularity. Specifically, we enrich the NURBS model by computing intermediate

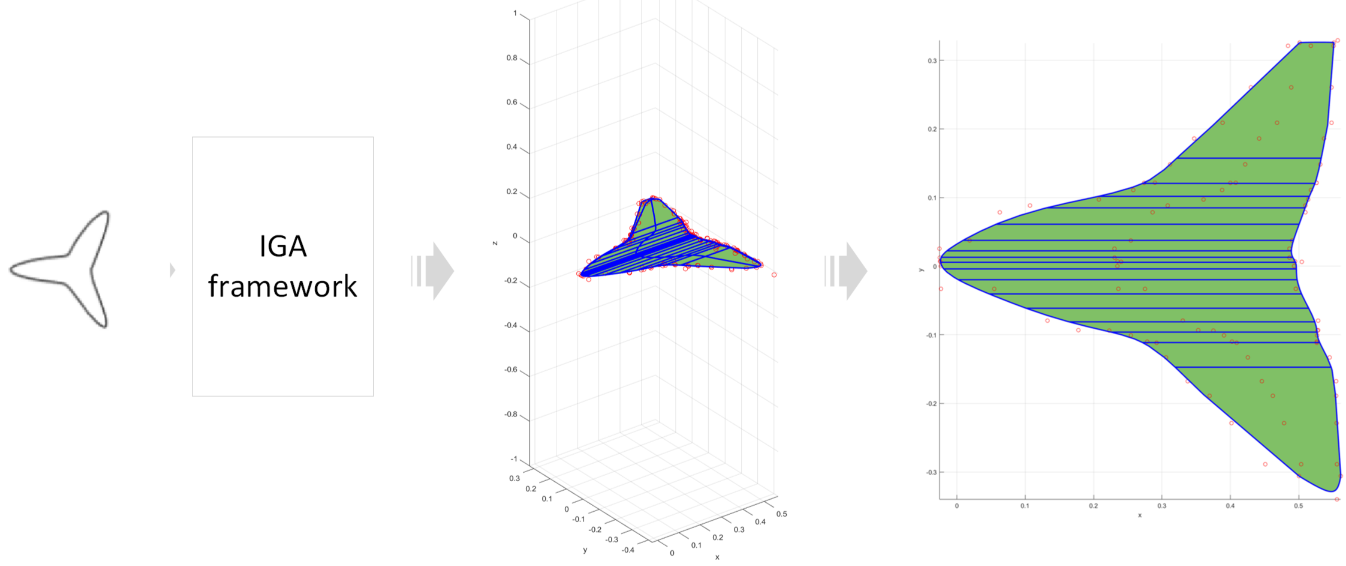

FIGURE 3: Pipeline of IGA framework

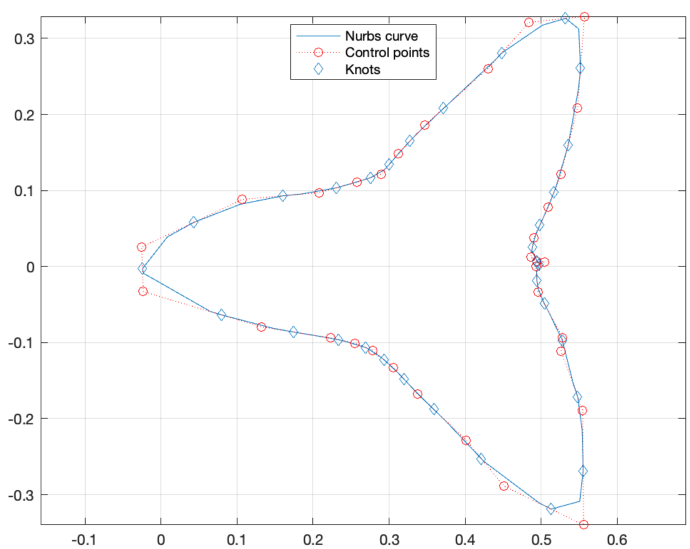

FIGURE 4: NURBS curve generated

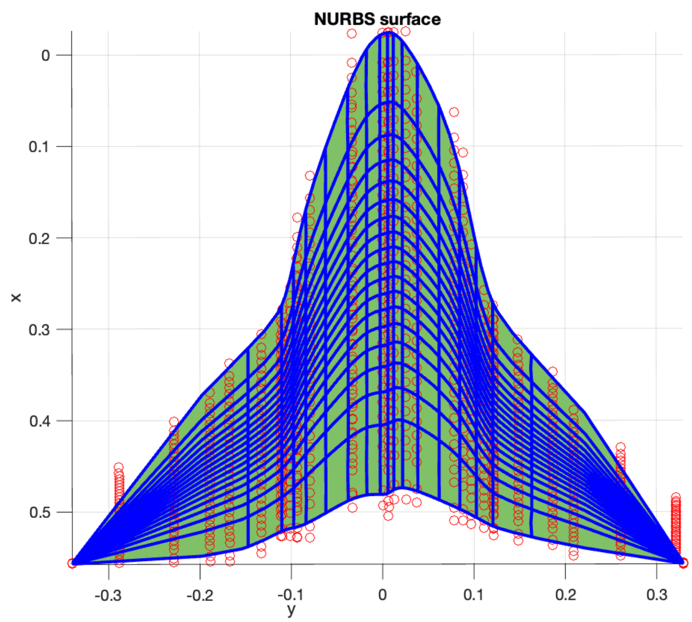

FIGURE 5: IGA before knot refinement and with 32×32 interpolation

values $\frac{\xi(i)+\xi(i+1)}{2}$ for each consecutive pair of knots within the vector, as defined by the equation:

$$F_{\text{inter}} = \left\{ \frac{\xi(i) + \xi(i+1)}{2} \mid i = 1, \ldots, \text{length}(\xi) - 1 \right\} - \{\xi\}, \quad (24)$$

where we ensure that each new value added is unique by excluding those already existing in the knot vector, thereby enhancing the detail and analytical capabilities of the NURBS model without introducing redundant information.

2.4.1 Algorithmic Steps for Refinement.

1. *Initialization*: Start by calculating midpoints between consecutive knots to divide segments:

$$X_{\text{div}} = F_{\text{inter}}(\xi), \quad (25)$$

where F_{inter} calculates midpoints, omitting existing knots to ensure enhancement.

2. *Subdivision Values*: If multiple subdivisions are needed

($sDivs > 1$), refine iteratively:

$$\text{while } sDivs > 1: \begin{cases} X_{temp} = F_{\text{inter}}(\text{sort}([\xi, X_{div}])), \\ X_{div} = [X_{div}, X_{temp}], \\ sDivs = sDivs - 1. \end{cases} \tag{26}$$

3. *Knot Refinement*: Knot refinement involves integrating the subdivision values into the existing knot vector. Following this, the control points and knot vector are updated while carefully maintaining the curve's continuity and smoothness, despite the addition of new knots. The procedure for knot insertion for a NURBS curve is outlined below:

   Consider a NURBS curve defined by control points P with associated weights w and a knot vector U of degree p. We aim to discuss the process of knot insertion, a fundamental operation in NURBS curve manipulation.

   Knot insertion involves introducing a new knot u into the existing knot vector U, thereby affecting the curve's shape. Let U' represent the updated knot vector after insertion, which can be mathematically defined as:

$$U' = (U_1, U_2, \ldots, U_i, u, U_{i+1}, \ldots, U_n). \tag{27}$$

   Here, i denotes the index satisfying $U_i < u < U_{i+1}$. Following knot insertion, adjustments to the control points and weights are necessary to preserve the curve's integrity. The updated control points P' and weights w' can be expressed as:

$$P' = (P'_1, P'_2, \ldots, P'_m), \tag{28}$$

$$w' = (w'_1, w'_2, \ldots, w'_m). \tag{29}$$

   The formulas for P'_j and w'_j ensure a smooth transition and continuity in the curve:

$$P'_j = \frac{U_{i+p+1} - u}{U_{i+p+1} - U_j} P_j + \frac{u - U_j}{U_{i+p+1} - U_j} P_{j-1}, \tag{30}$$

$$w'_j = \frac{U_{i+p+1} - u}{U_{i+p+1} - U_j} w_j + \frac{u - U_j}{U_{i+p+1} - U_j} w_{j-1}. \tag{31}$$

   Here, the index j will iterate p times for each knot inserted, where p denotes the degree of the curve. This multiplicity of control point insertion will ensure the continuity of the curve. These calculations adhere to de Boor's algorithm for knot insertion, ensuring smooth curve transitions while accommodating the new knot u. The IGA result after knot refinement is shown in Fig. 6.

2.4.2 Connectivity Matrix Construction. Connectivity matrices map local element coordinates to the global domain:

1. Establish the number of elements (nel), total basis functions (nnp), and local basis functions (nen):

$$\begin{aligned} nel &= (n - \text{deg}.p) \times (m - \text{deg}.q), \\ nnp &= n \times m, \\ nen &= (\text{deg}.p + 1) \times (\text{deg}.q + 1). \end{aligned} \tag{32}$$

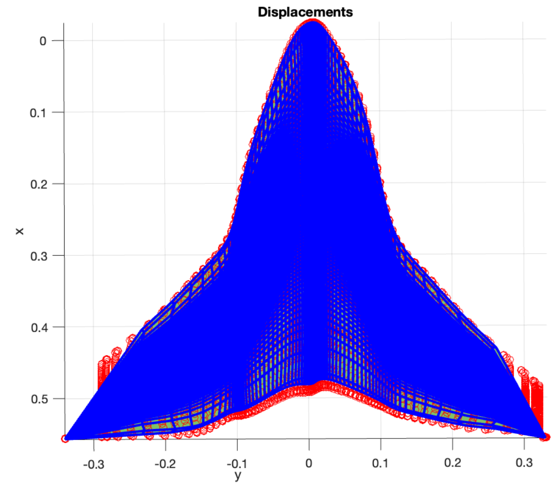

FIGURE 6: IGA after knot refinement

2. Construct INN (Index of Node Numbers) and IEN (Index of Element Numbers) for the element-to-global mapping. The Index of Node Numbers (INN) and the Index of Element Numbers (IEN) play significant roles in finite element analysis. They help in mapping between local and global structures. In the context of NURBS (Non-Uniform Rational B-Splines), we typically use the Index NURBS Coordinates (INC) instead of the traditional INN. INC is specifically designed to map a global basis function number to its corresponding coordinates in the NURBS parameter space. For a given global basis function number A, INC maps A to a pair of coordinates (ξ, η). These coordinates define the position of the basis function in the NURBS parameter space. The mathematical representation of this mapping is:

$$\text{INC}(A, :) = (\xi_{\text{coord}}(A), \eta_{\text{coord}}(A)), \tag{33}$$

   where $\xi_{\text{coord}}(A)$ represents the ξ-coordinate of the global basis function A, and $\eta_{\text{coord}}(A)$ represents the η-coordinate of the global basis function A. The Index of Element Numbers (IEN), on the other hand, is used to relate global basis function numbers to their local orderings within each finite element. This mapping is crucial for assembling the global matrix from local elements, as it dictates how each local basis function (or shape function) contributes to the global solution.

2.5 Defining Material Constitutive Relations for NURBS Simulations

The constitutive relationship of a material under mechanical stress is crucial for accurate simulations. In NURBS-based plane stress analysis, this relationship is encapsulated within the material matrix D, which is derived based on the material's elasticity properties.

2.5.1 Material Matrix Construction. The function computes the material matrix D for a linear elastic and isotropic material based on the specified problem type, Young's modulus E, and Poisson's ratio ν:

1. *Plane Stress (ptype = 1)*: For simulations assuming that out-of-plane stresses are negligible:

$$D = \frac{E}{1-\nu^2}\begin{bmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & \frac{1-\nu}{2}\end{bmatrix}. \tag{34}$$

2. *Plane Strain and Axisymmetry (ptype = 2 or 3)*: For cases with deformations restricted in one dimension or symmetric about an axis:

$$D = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix}1-\nu & \nu & \nu & 0\\ \nu & 1-\nu & \nu & 0\\ \nu & \nu & 1-\nu & 0\\ 0 & 0 & 0 & \frac{1-2\nu}{2}\end{bmatrix}. \tag{35}$$

3. *Three-Dimensional Stress (ptype = 4)*: For comprehensive 3D simulations:

$$D = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix}1-\nu & \nu & \nu & 0 & 0 & 0\\ \nu & 1-\nu & \nu & 0 & 0 & 0\\ \nu & \nu & 1-\nu & 0 & 0 & 0\\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0\\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0\\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2}\end{bmatrix}. \tag{36}$$

   Within the NURBS-based simulation framework, the stiffness matrix D is assigned by invoking E_Y, ν_P, where E_Y is Young's modulus and ν_P is Poisson's ratio.

In NURBS-based simulations for plane stress analysis, the precision of numerical integration is crucial, achieved effectively through Gauss-Legendre quadrature tailored to the NURBS basis functions' polynomial order. We compute the necessary Gauss points (gp) and weights (w), crucial for integration, based on the order p. This mathematical strategy ensures that the integration accuracy is commensurate with the degree of polynomial complexity inherent in the NURBS model, which is vital for capturing the nuanced physical behaviors of the system.

The implementation within the NURBS framework is methodically outlined as follows:

$$[gp_x, w_x] = \text{Gaussian Points}(p), \tag{37}$$

$$[gp_y, w_y] = \text{Gaussian Points}(q), \tag{38}$$

where p and q are the degrees in two directions. Inside the function, the input degree is increased by 1 to obtain the order of the NURBS curve. Then, the function simply returns the corresponding Gauss points (gp_x, gp_y) and weights (w_x, w_y), enabling precise integration over the NURBS geometry.

In the framework of NURBS-based simulations for mechanical engineering analysis, the computation of the stiffness matrix (K) and load vector (F) is integral to understanding material responses and structural behaviors. We apply the following function to calculate the local stiffness matrix contribution (Ke) from individual NURBS elements, utilizing the following mathematical formulation:

$$Ke = B^T D B J_{\text{mod}}, \tag{39}$$

where B is the strain-displacement matrix formed from NURBS basis function derivatives (dR_{dx}), D denotes the material's constitutive matrix embodying its physical properties, and J_{mod} signifies the modified Jacobian determinant that integrates the weights from Gauss-Legendre quadrature.

The assembly process in NURBS-based simulations is pivotal for merging local element contributions into a global context, crucial for forming the global stiffness matrix (K) and load vector (F). This ensures that individual element properties are cohesively integrated into the entire system per NURBS mesh connectivity.

Initially, K and F are set to zero, sized $[ndof \times ndof]$ and $[ndof \times 1]$ respectively, with $ndof$ representing the system's total degrees of freedom. Each element contributes through a loop where 1) the local stiffness matrix Ke and load vector Fe are computed based on material properties, geometric traits, and NURBS basis functions, often utilizing Gauss-Legendre quadrature for integration; 2) contributions are mapped from local to global systems via connectivity arrays, with local degrees of freedom linked to global indices through an ID mapping. The assembly transitions local matrices to global using:

$$K_{global}(ID(i), ID(j)) += Ke_{local}(i, j), \tag{40}$$

$$F_{global}(ID(i)) += Fe_{local}(i), \tag{41}$$

Following this, boundary conditions are applied, modifying K and F for any fixed displacements or specific loads. This streamlined process results in a comprehensive global matrix representation that is foundational for displacement solutions and subsequent stress and strain evaluations, encapsulating the mechanical properties within the unified NURBS framework.

2.5.2 Boundary Conditions and Solution.

1. The application of boundary conditions is a critical step in the simulation process. Specifically, we identify and constrain control points where displacement must be restricted. In the context of our model, control points with an x-coordinate less than or equal to 0.1 (denoted as $loc_x = 0.1$) are fixed to prevent movement. This operation is crucial for mimicking real-world constraints and ensuring the physical accuracy of the simulation.

2. Following the imposition of boundary conditions, the system of equations defined by the stiffness matrix K and force vector F is solved to find the displacement vector **a**. The solution of this equation provides the deformation at each control point, allowing us to analyze the structural response under applied loads. This step is fundamental to understanding the mechanical behavior of the model under investigation.

3. EXPERIMENTS

In this section, we discuss the analytical process of the framework, including 1) the implementation details, 2) the NURBS-based model and its generation results, and 3) the deformation/displacement calculated through our isogeometric analysis framework given certain physical constraints, such as material properties (Young's modulus, Poisson's ratio), force, and its direction.

Our overall framework can be divided into two aspects. A deep generative model for NURBS-based design synthesis is first trained on the superformula I/II [4] and airfoil datasets (*i.e.*, UIUC airfoil datasets[1]) with PyTorch 2.0 on Nvidia RTX4090. The generated samples are then transformed into IGA-friendly samples via the interpolation strategy proposed, and those interpolated samples are fed into MATLAB-based IGA framework for further analysis [26].

As shown in Fig. 2, we first generate samples for preprocessing of IGA, and then these sample coordinates and weights are transformed into new coordinates via the interpolation strategy introduced in Section 2.2.

Taking a Superformula sample as an illustration, as shown in Fig. 3, the coordinates and meshes are re-arranged into IGA geometry. Given specific material properties and force location and magnitude, we can analyze the corresponding deformation and displacements. In this study, we specifically define two groups of material properties and loading conditions:

1. We apply boundary conditions for all control-points that are less or equal to 0.1 fixed in x-axis.

2. We define the material to be structural steel (Young's Modulus (E): 210 GPa, Poisson's Ratio (v): 0.30), given a negative force load (3e5) for all knot vectors that is greater than 0.01 in x direction, the results are shown in Fig. 7.

3. We define the force load to be consistent, and the material is changed to aluminum (E: 68.9 GPa, V: 0.33) with a positive load of 3e5. The deformation is then shown in Fig. 8.

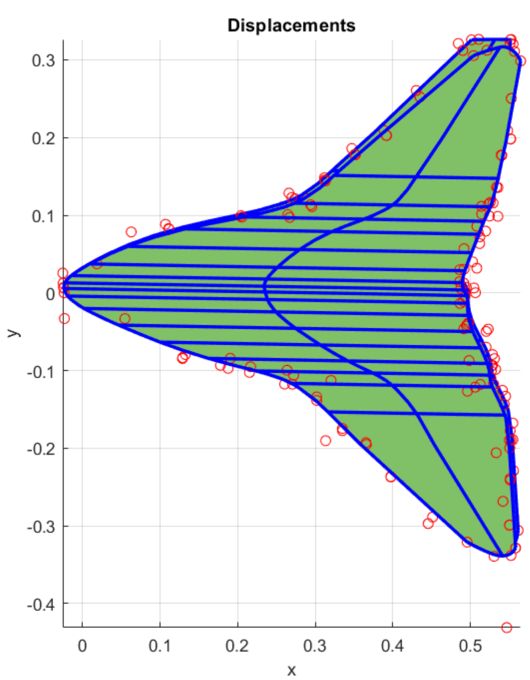

FIGURE 7: Displacement analysis given load -3e5, structural steel.

[1]https://m-selig.ae.illinois.edu/ads/coord_database.html

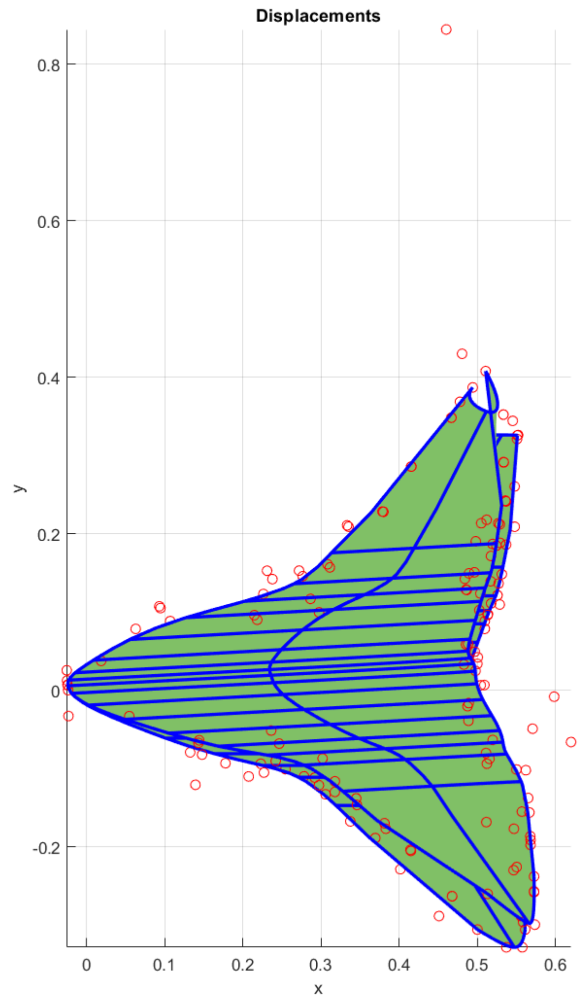

FIGURE 8: Displacement analysis given load +3e5, aluminum.

4. CONCLUSION

The innovative framework represents a stride forward in the realm of isogeometric analysis (IGA), boasting improvements in both efficiency and accuracy regarding curvilinear configuration generation and subsequent analysis. By incorporating sophisticated spline parameterization techniques, the framework not only elevates mesh quality but also deepens our comprehension of underlying physical properties. Its validation against complex datasets underscores its efficacy, heralding a pivotal moment in the fusion of machine learning and IGA. This amalgamation sets the stage for forthcoming computational breakthroughs aimed at surmounting the geometric modeling hurdles inherent in IGA.

Limitations and Future Work: Integrating NURBS preprocessing within the deep generative model enhances its compatibility with IGA, streamlining the process and enabling direct utilization of generated samples for IGA tasks, eliminating the need for additional data manipulation. Furthermore, the extension of the existing NURBS curve-based model to a NURBS surface-based model promises a more accurate and direct learning approach for NURBS surfaces, optimizing their suitability for instantaneous IGA applications. Ultimately, this endeavor aims to revolutionize the automation of intricate 3D geometric modeling for IGA through the powerful assistance of deep learning techniques.

ACKNOWLEDGMENTS

This research was partially supported by funding from the National Science Foundation (NSF) PD 22-8084 Computational and Data-Enabled Science and Engineering (CDS&E) funding opportunity through award #2245299.

REFERENCES

[1] Zhuang, Xinwei, Ju, Yi, Yang, Allen and Caldas, Luisa. "Synthesis and generation for 3D architecture volume with generative modeling." *International Journal of Architectural Computing* (2023): p. 14780771231168233.

[2] Xie, Wei-Cheng, Zou, Xiu-Fen, Yang, Jian-Dong and Yang, Jie-Bin. "Iteration and optimization scheme for the reconstruction of 3D surfaces based on non-uniform rational B-splines." *Computer-Aided Design* Vol. 44 No. 11 (2012): pp. 1127–1140.

[3] Du, Xiaosong, He, Ping and Martins, Joaquim RRA. "A B-spline-based generative adversarial network model for fast interactive airfoil aerodynamic optimization." *AIAA Scitech 2020 Forum*: p. 2128. 2020.

[4] Chen, Wei, Chiu, Kevin and Fuge, Mark D. "Airfoil design parameterization and optimization using bézier generative adversarial networks." *AIAA journal* Vol. 58 No. 11 (2020): pp. 4723–4735.

[5] De Lazzari, Elena, van den Boom, Sanne J, Zhang, Jian, van Keulen, Fred and Aragón, Alejandro M. "A critical view on the use of Non-Uniform Rational B-Splines to improve geometry representation in enriched finite element methods." *International Journal for Numerical Methods in Engineering* Vol. 122 No. 5 (2021): pp. 1195–1216.

[6] Munira, Ali, Jaafar, NUR NAJMIYAH, Fazilah, Abdul Aziz and Nooraizedfiza, Z. "Review on non uniform rational B-spline (NURBS): Concept and Optimization." *Advanced Materials Research* Vol. 903 (2014): pp. 338–343.

[7] Fisher, TR and Wales, RQ. "Three dimensional solid modeling of geo-objects using Non-Uniform Rational B-Splines (NURBS)." *Three-dimensional modeling with geoscientific information systems*. Springer (1992): pp. 85–105.

[8] Audoux, Yohann, Montemurro, Marco and Pailhes, Jérôme. "A surrogate model based on Non-Uniform Rational B-Splines hypersurfaces." *Procedia CIRP* Vol. 70 (2018): pp. 463–468.

[9] Nguyen, Vinh Phu, Anitescu, Cosmin, Bordas, Stéphane PA and Rabczuk, Timon. "Isogeometric analysis: an overview and computer implementation aspects." *Mathematics and Computers in Simulation* Vol. 117 (2015): pp. 89–116.

[10] Cottrell, J Austin, Hughes, Thomas JR and Bazilevs, Yuri. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons (2009).

[11] Cottrell, J Austin, Reali, Alessandro, Bazilevs, Yuri and Hughes, Thomas JR. "Isogeometric analysis of structural vibrations." *Computer methods in applied mechanics and engineering* Vol. 195 No. 41-43 (2006): pp. 5257–5296.

[12] Hughes, Thomas JR, Cottrell, John A and Bazilevs, Yuri. "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement." *Computer methods in applied mechanics and engineering* Vol. 194 No. 39-41 (2005): pp. 4135–4195.

[13] Bhavikatti, SS. *Finite element analysis*. New Age International (2005).

[14] Hughes, Thomas JR and Evans, John A. "Isogeometric analysis." *Proceedings of the International Congress of Mathematicians 2010 (ICM 2010) (In 4 Volumes) Vol. I: Plenary Lectures and Ceremonies Vols. II–IV: Invited Lectures*: pp. 299–325. 2010. World Scientific.

[15] Wang, Dandan, Xu, Jinlan, Gao, Fei, Wang, Charlie CL, Gu, Renshu, Lin, Fei, Rabczuk, Timon and Xu, Gang. "IGA-Reuse-NET: A deep-learning-based isogeometric analysis-reuse approach with topology-consistent parameterization." *Computer Aided Geometric Design* Vol. 95 (2022): p. 102087.

[16] Panchal, Jitesh H, Fuge, Mark, Liu, Ying, Missoum, Samy and Tucker, Conrad. "Machine learning for engineering design." *Journal of Mechanical Design* Vol. 141 No. 11 (2019): p. 110301.

[17] Amini, Mahyar, Sharifani, Koosha and Rahmani, Ali. "Machine Learning Model Towards Evaluating Data gathering methods in Manufacturing and Mechanical Engineering." *International Journal of Applied Science and Engineering Research* Vol. 15 No. 2023 (2023): pp. 349–362.

[18] Hsieh, Yung-Ting, Anjum, Khizar, Huang, Songjun, Kulkarni, Indraneel and Pompili, Dario. "Neural Network Design via Voltage-based Resistive Processing Unit and Diode Activation Function - A New Architecture." *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*: pp. 59–62. 2021. DOI 10.1109/MWSCAS47672.2021.9531917.

[19] Hsieh, Yung-Ting, Qi, Zhuoran and Pompili, Dario. "ML-based Joint Doppler Estimation and Compensation in Underwater Acoustic Communications." *Proceedings of the 16th International Conference on Underwater Networks & Systems*: pp. 1–8. 2022.

[20] Pan, Zhenyu, Sharma, Anshujit, Hu, Jerry Yao-Chieh, Liu, Zhuo, Li, Ang, Liu, Han, Huang, Michael and Geng, Tony. "Ising-traffic: Using ising machine learning to predict traffic congestion under uncertainty." *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 8: pp. 9354–9363. 2023.

[21] He, Chunguang, Wang, Dianhai, Yu, Yi, Cai, Zhengyi et al. "A Hybrid Deep Learning Model for Link Dynamic Vehicle Count Forecasting with Bayesian Optimization." *Journal of Advanced Transportation* Vol. 2023 (2023).

[22] Creswell, Antonia, White, Tom, Dumoulin, Vincent, Arulkumaran, Kai, Sengupta, Biswa and Bharath, Anil A. "Generative adversarial networks: An overview." *IEEE signal processing magazine* Vol. 35 No. 1 (2018): pp. 53–65.

[23] Gupta, Vibhushit, Jameel, Azher, Verma, Shubham K, Anand, Sanjeev and Anand, Yatheshth. "An Insight on NURBS Based Isogeometric Analysis, Its Current Status

and Involvement in Mechanical Applications." *Archives of Computational Methods in Engineering* Vol. 30 No. 2 (2023): pp. 1187–1230.

[24] Zhuang, Xinwei, Design, CA, Phase, ED, Generative, C and Network, AN. "Rendering Sketches." *eCAADe 2022* Vol. 1 (1973): pp. 517–521.

[25] Riedmiller, Martin and Lernen, A. "Multi layer perceptron." *Machine Learning Lab Special Lecture, University of Freiburg* Vol. 24 (2014).

[26] CALFEM. "CALFEM for MATLAB." https://github.com/CALFEM/calfem-matlab (2023). MATLAB toolbox for finite element analysis.

[27] Piegl, Les and Tiller, Wayne. *The NURBS book*. Springer Science & Business Media (2012).

[28] Xie, Hui and Qin, Hong. "A novel optimization approach to the effective computation of NURBS knots." *International Journal of Shape Modeling* Vol. 7 No. 02 (2001): pp. 199–227.