Gulf Journal of Mathematics
ISSN: 2309-4966
Vol 22, Issue 2 (2026) 1-12
https://doi.org/10.56947/gjom.v22i2.4199

SUPERVISED INITIALIZATION OF TENSION PARAMETERS IN NON-UNIFORM NON-STATIONARY SUBDIVISION SCHEMES

AMINE ARHANDOU[1], MOHAMED-YASSIR NOUR[2*] and ABDELLAH LAMNII[1]

ABSTRACT. In this work, we propose a supervised learning approach for initialising local tension parameters in non-uniform and non-stationary subdivision schemes. A neural network generates an initial tension vector from a control polygon, replacing manual parameter selection. These tensions are then propagated according to the analytical evolution rule of the ω-NSS scheme, preserving its theoretical properties. Numerical results show that this learned initialization strategy yields stable and consistent tension distributions, while improving reproducibility and reducing user intervention.

*Keywords.*tension parameters, Non-uniform non-stationary subdivision, Neural network, Supervised learning.
2020 Mathematics Subject Classification. Primary 46L55; Secondary 44B20

1. INTRODUCTION

Subdivision schemes are now essential tools in geometric modeling [1, 6, 7, 12, 13], computer-aided design [2, 3, 8, 10], and computer graphics [5, 9, 11, 14, 15]. They consist of refining an initial control polygon through repeated iterations to generate a smooth limit curve which interpolates or approximates the original shape. Their ease of implementation, local nature and numerical efficiency make them essential methods for constructing complex shapes, as well as for numerous applications in engineering and numerical simulation [16].

With the evolution of modeling needs, uniform stationary schemes are now proving insufficient to capture certain geometric subtleties. This has led to the development of non-stationary non-uniform schemes in which refinement rules can vary depending on the subdivision level and position within the control polygon [4]. Local parameters allow fine adjustment of the rigidity or flexibility of the generated curve, thus offering increased expressiveness and improved shape control [8]. These tension parameters play a crucial role in ωB-spline schemes as well as in other non-uniform variants. They determine whether the resulting boundary behaves in a polynomial, trigonometric or hyperbolic manner [8].

Date: Received: Feb 1, 2026; Revised: Feb 28, 2026; Accepted: Mar 14, 2026.
* Corresponding author

In theory, the ability to vary these tensions point by point is a major advantage for modeling heterogeneous shapes. In practice, however, manually determining these values remains one of the most challenging aspects. Each segment of the polygon may require specific adjustment, making the exploration of the parameter space difficult, time-consuming, and highly dependent on the user's experience [8]. When the number of control points increases, the dimension of the parameter space grows accordingly, making systematic manual tuning increasingly impractical.

To address this limitation, we propose a supervised learning framework for the initialization of local tension parameters. The objective is not to modify the analytical structure of the subdivision scheme, but to replace manual parameter tuning with a learned initialization strategy derived from representative examples. A fully connected neural network is used to generate an initial tension vector associated with a given control polygon. The model is trained on a structured dataset of polygons paired with reference tension configurations obtained from analytical settings and controlled synthetic examples.

Once initialized, the tensions are propagated through the subdivision levels according to the analytical evolution rule of the non-uniform non-stationary ω-NSS scheme. The neural component therefore intervenes only at the initialization stage, while the refinement process remains entirely governed by the original analytical formulation, preserving convergence and regularity properties. Numerical experiments illustrate that this learned initialization strategy provides stable and consistent parameter distributions, comparable to expert-driven manual tuning, while improving reproducibility and reducing user intervention.

The remainder of this article is organised as follows. Section 2 reviews the framework of non-uniform and non-stationary subdivision schemes with local tension parameters. Section 3 presents the proposed supervised neural model for parameter initialization. Section 4 details the integration of the learned tensions into the subdivision process. Numerical results and comparisons are presented in Section 5.

2. Non-Uniform Non-Stationary Subdivision with Local Tensions

In this section, we briefly review the framework of non-uniform subdivision schemes, as introduced by Dyn et al. [4]. We then present the order-3 subdivision scheme proposed by Lamnii et al. [8], highlighting its main practical limitation (the manual selection of tension parameters).

Definition 1. *A univariate subdivision scheme is said to be non-uniform and non-stationary if its refinement mask depends explicitly on the subdivision level k and on the spatial index i. Denoting the corresponding mask by $\mathbf{a}^{k,i} = \{\mathbf{a}^{k,i}_j : j \in \mathbb{Z}\}$, the refinement rule is defined as*

$$P_i^{k+1} = \sum_{j\in\mathbb{Z}} \mathbf{a}^{k,i}_{i-2j}\, P_j^k. \tag{1}$$

Where

$$P^k = \{P_i^k\}_{i=0}^{N_k},$$

denotes the control polygon at level k.

Definition 2. *For each level k and index i, the refinement mask $\mathbf{a}^{k,i}$ can be represented by the Laurent polynomial*

$$\mathcal{L}^{k,i}(z) = \sum_{j\in\mathbb{Z}} \mathbf{a}_j^{k,i}\, z^j, \tag{2}$$

assuming that the mask has finite support.

Definition 3. *Let $\mathbf{a}^{k,i}$ denote the masks of a non-stationary subdivision scheme and let $\mathbf{a}$ be the mask of a stationary scheme. The two schemes are said to be* asymptotically consistent *if*

$$\sum_{k=0}^{\infty} \sup_{i\in\mathbb{Z}} \left\|\mathbf{a}^{k,i} - \mathbf{a}\right\|_\infty < \infty. \tag{3}$$

Definition 4. *The refinement rule of the cubic ω non-uniform non-stationary subdivision scheme (see [8]) is defined by*

$$\begin{aligned} (P^{k+1})_{2i} &= a_i^k\, P_i^k + b_i^k\, P_{i+1}^k, \\ (P^{k+1})_{2i+1} &= b_i^k\, P_i^k + a_i^k\, P_{i+1}^k, \end{aligned} \tag{4}$$

where

$$a_i^k = \frac{4u_i^k - 1}{4u_i^k}, \qquad b_i^k = \frac{1}{4u_i^k}, \qquad a_i^k + b_i^k = 1, \tag{5}$$

and u_i^k denotes the local tension parameter.

The local tension parameters are updated according to

$$u_i^{k+1} = \sqrt{\frac{1+u_i^k}{2}}, \qquad i = 0, \ldots, N_k - 1. \tag{6}$$

Algorithm 1 in [8] relies on the manual selection of the initial parameters vector $U^0 := \{u_i^0, \quad i \in \mathbb{Z}\}$, which constitutes its main practical limitation, particularly for complex or heterogeneous geometries.

Although mathematically well established, the above scheme requires the initial parameter distribution U^0 to be specified by the user. This manual step is time-consuming and highly dependent on experience. When the number of control points increases, the dimension of the parameter space grows accordingly, making systematic exploration increasingly impractical. These issues are amplified in heterogeneous geometries, where different regions of the curve may require distinct tension behaviors. Moreover, manual parameter selection limits reproducibility.

These observations motivate the need for a systematic strategy for initializing the local tension parameters. In the next section, we introduce a supervised learning framework designed to provide such an initialization, while preserving the analytical structure of the subdivision scheme.

3. Supervised Neural Model for Tension Initialization

In this section, we introduce a supervised neural model designed to provide an initialization of local tension parameters for a given control polygon. The objective is not to modify the analytical formulation of the subdivision scheme, but to replace manual parameter tuning with a learned initialization strategy derived from representative examples. The model acts as a regression mechanism that associates each vertex of the control polygon with a tension value consistent with the supervised training data.

Initialization is performed on a point-by-point basis, meaning that each vertex of the control polygon is processed independently. This design choice reflects the role of the model, which is to learn a supervised parameter initialization rather than to estimate intrinsic geometric quantities. The neural network serves as a parametric regressor, while the geometric coupling between control points is introduced by the analytical subdivision operator itself.

3.1. **Problem Formulation.** Given a control polygon

$$P = \{P_i\}_{i=0}^{N-1}, \qquad P_i = (x_i, y_i)^\top,$$

we seek to compute a sequence of local tensions

$$U = \{u_i\}_{i=0}^{N-1}, \qquad u_i > 0,$$

that will serve as the initial parameter vector U^0 in the subdivision scheme.

In a supervised setting, we consider a training set

$$\mathcal{D} = \{(P^{(m)}, U^{(m)})\}_{m=1}^{M},$$

where each polygon is associated with a reference tension vector obtained from predefined analytical configurations or controlled examples.

The objective is to determine the network parameters θ that minimize the empirical regression error:

$$\theta^\star = \arg\min_\theta \sum_{m=1}^{M} \left\| f_\theta(P^{(m)}) - U^{(m)} \right\|_2^2 .$$

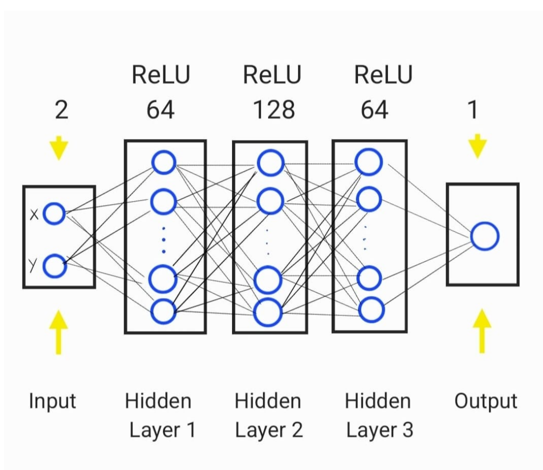

FIGURE 1. Architecture of the supervised neural network used for tension initialization. Each vertex (x_i, y_i) is processed independently through a fully connected network with three hidden layers and ReLU activations, producing a single scalar tension value.

Figure 1 illustrates the fully local structure of the proposed neural model, which assigns one tension value per vertex without introducing architectural coupling between control points.

The model consists of a fully connected neural network operating point by point. Although the training process flattens the batch and vertex dimensions for computational efficiency, the network itself processes each vertex independently.

Each vertex P_i is passed through three successive linear layers:

$$\hat{u}_i = W_3\,\sigma\big(W_2\,\sigma(W_1\,P_i + b_1) + b_2\big) + b_3,$$

where:

- $W_1 \in \mathbb{R}^{64\times 2}$, $W_2 \in \mathbb{R}^{64\times 64}$, $W_3 \in \mathbb{R}^{1\times 64}$ are the weight matrices,
- b_1, b_2, b_3 are bias vectors,
- $\sigma(\cdot)$ denotes the ReLU activation function.

The model is chosen for its simplicity and its ability to approximate nonlinear regression mappings in a stable manner. It serves exclusively to generate an initial parameter distribution consistent with the supervised dataset.

3.2. **Data Preparation.** The training data consists of polygons of varying sizes. In order to use mini-batches, the sequences are padded to obtain tensors of homogeneous dimensions:

$$X \in \mathbb{R}^{B\times N_{\max}\times 2}, \qquad Y \in \mathbb{R}^{B\times N_{\max}},$$

where B denotes the batch size and $N_{\max}$ the maximum polygon length in the dataset.

The inputs are normalized and converted into PyTorch tensors. The target tension values are treated as strictly positive real scalars. Since padding is introduced to handle variable-length polygons, padded entries are explicitly masked during the loss computation. This ensures that only valid vertices contribute to the optimization process and prevents bias due to zero-padding.

The loss function used is the Mean Squared Error (MSE):

$$\mathcal{L}(\theta) = \frac{1}{M}\sum_{m=1}^{M} \big\| f_\theta(P^{(m)}) - U^{(m)} \big\|_2^2 .$$

Optimization is performed using the Adam algorithm, known for its robustness in nonlinear regression tasks. The parameters are updated through stochastic gradient descent via backpropagation.

The learning procedure is summarized below.

Algorithm 1: Training procedure for the tension initialization network

Require: Dataset $\mathcal{D}$, number of epochs E, batch size B

1: Initialize network parameters θ
2: **for** $e = 1$ *to* E **do**
3:   **for** *batch* (X, Y) *of size* B **do**
4:     Flatten input polygons into tensors of shape $(B \cdot N_{\max}, 2)$
5:     Compute outputs $\hat{Y} = f_\theta(X)$
6:     Compute masked loss $\mathcal{L}$
7:     Update θ using Adam
8: **return** optimized parameters $\theta^\star$

Once trained, the model can be applied to a new control polygon:

$$U^0 = f_{\theta^\star}(P^0).$$

These initial tensions are then propagated according to the analytical evolution rule

$$u_i^{k+1} = \sqrt{\frac{1 + u_i^k}{2}},$$

and integrated into the non-uniform subdivision matrix at each refinement level.

The neural component therefore intervenes only at the initialization stage, while the subsequent subdivision process remains entirely governed by the analytical ω-NSS formulation.

4. Integration of the Learned Tensions into the Subdivision Scheme

In this section, we describe how the tensions obtained from the supervised neural model are incorporated into the non-uniform and non-stationary subdivision scheme. We first explain how the network is applied to a new control polygon, then detail the construction of the refinement masks from the resulting tensions, the evolution of these parameters across levels, and finally the complete subdivision pipeline with learned initialization.

The mathematical structure of the subdivision scheme remains unchanged. Only the initialization of the local tension parameters is replaced by the supervised neural model, while all refinement rules remain strictly analytical.

4.1. Application of the Neural Model to a New Control Polygon.

Consider a new control polygon

$$P^0 = \{P_i^0\}_{i=0}^{N-1}, \qquad P_i^0 \in \mathbb{R}^2,$$

which we aim to refine using the subdivision scheme with local tensions.

The trained neural network is applied to generate an initial tension vector:

$$U^0 = f_{\theta^\star}(P^0) = \{u_i^0\}_{i=0}^{N-1}.$$

The neural model is used only once, at the initial level, to produce the starting tension distribution. No learning-based intervention is performed during subsequent subdivision levels.

In practice, the polygon is first converted into a tensor (optionally normalized) and then passed through the trained model. The outputs are converted into real scalars, and if necessary, a lower bound is imposed to ensure positivity of the tensions (for example $u_i^0 \geq \varepsilon > 0$). This constraint is consistent with the analytical requirements of the subdivision scheme, as the refinement coefficients are well-defined only for strictly positive tension values.

The inference procedure for a given polygon can be summarized as:

$$P^0 \longmapsto \text{tensorization and normalization} \longmapsto f_{\theta^\star} \longmapsto U^0.$$

4.2. **Incorporation of the Learned Tensions into the Refinement Masks.** The initial tensions U^0 are used to construct the refinement coefficients of the non-uniform, non-stationary cubic scheme. For each segment associated with the tension u_i^k at level k, the combination coefficients are computed using the same analytical expressions as in the classical ω-NSS formulation:

$$a_i^k = \frac{4u_i^k - 1}{4u_i^k}, \qquad b_i^k = \frac{1}{4u_i^k}, \qquad a_i^k + b_i^k = 1.$$

These parameters are integrated into a non-uniform subdivision matrix $M^k(U^k)$ such that

$$P^{k+1} = M^k(U^k)\, P^k.$$

In practice, this matrix is constructed directly from the local formulas of the ω-NSS scheme by inserting, for each row, the corresponding coefficients (a_i^k, b_i^k). This implementation is a direct numerical realization of the analytical refinement operator $M^k(U^k)$.

4.3. **Parameter Evolution Across Subdivision Levels.** After the first subdivision step using the initialized tensions U^0, the parameters are updated across levels according to the analytical evolution rule:

$$u_i^{k+1} = \sqrt{\frac{1 + u_i^k}{2}}, \qquad i = 0, \ldots, N_k - 1.$$

This contracting relation keeps the tensions within a stable range and, under appropriate conditions, ensures convergence of the scheme and regularity of the limit curve.

The neural model therefore intervenes only at the initialization stage, while the evolution sequence $\{U^k\}_{k \geq 1}$ is entirely determined by the analytical scheme. Consequently, the learning component remains fully decoupled from the multilevel refinement process.

4.4. **Subdivision Pipeline with Neural Initialization.** The complete process consists of initializing the local tensions using the trained neural network, constructing the corresponding subdivision matrix, refining the control polygon, updating the tensions analytically, and iterating this procedure until the desired subdivision level is reached. The following algorithm summarizes this process.

Algorithm 2: Subdivision pipeline with neural tension initialization

Require: Initial polygon P^0, trained model $f_{\theta^\star}$, number of levels L
1: Generate initial tensions: $U^0 \leftarrow f_{\theta^\star}(P^0)$
2: $k \leftarrow 0$
3: **while** $k < L$ **do**
4: Build subdivision matrix $M^k(U^k)$
5: Compute refined polygon: $P^{k+1} \leftarrow M^k(U^k)\, P^k$
6: Update tensions using $u_i^{k+1} = \sqrt{\frac{1+u_i^k}{2}}$
7: $k \leftarrow k+1$
8: **return** P^L

This pipeline differs from the classical subdivision algorithm only at the initialization stage. All subsequent steps strictly follow the standard non-uniform non-stationary analytical refinement procedure.

5. Numerical Results

This section presents numerical experiments illustrating the behavior of the proposed subdivision scheme combined with the supervised neural initialization. The results are organized into three parts: the progressive evolution of the subdivision from a control polygon, a comparison with classical schemes using global tension parameters, and a comparison with classical non-uniform subdivision based on manual parameter selection.

5.1. **Subdivision Process Using Neural Initialization.** We first consider a control polygon representing a complex shape containing both smooth regions and areas with significant directional variation. The trained neural model is applied to this polygon to generate an initial vector of local tensions, which is then integrated into the non-uniform, non-stationary subdivision scheme.

It is important to emphasize that the neural model is used exclusively at the initialization stage, while the subdivision process itself strictly follows the analytical refinement rules described in Section 2.

Figure 2 illustrates the successive steps of the subdivision process: the initial control polygon, the first and second refinement levels, and the resulting limit curve.

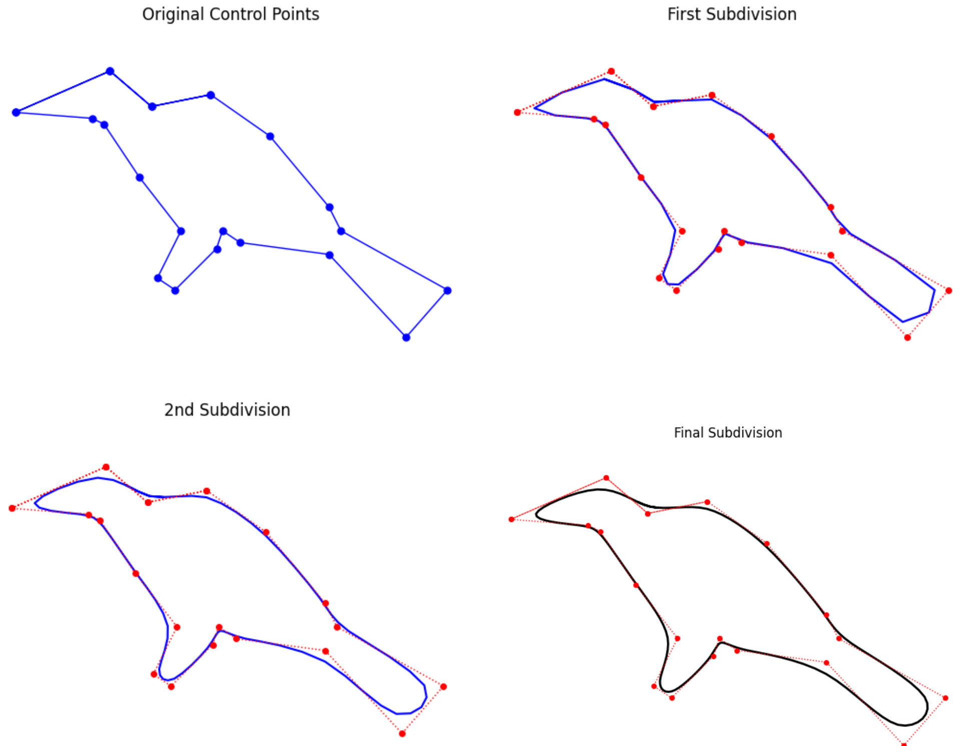

FIGURE 2. Subdivision process using learned local tensions: original control polygon, first subdivision, second subdivision, and final limit curve.

The results show that the scheme produces a smooth and stable limit curve. The spatially distributed tensions influence the local behavior of the curve without introducing oscillatory artifacts. The refinement process remains numerically stable at each level, confirming the compatibility between the supervised initialization and the analytical subdivision structure.

5.2. **Comparison with Trigonometric and Hyperbolic Subdivision Schemes.** We compare the proposed method with two classical subdivision schemes: a trigonometric scheme and a hyperbolic scheme, each governed by a single global tension parameter. The values of the global parameters are selected to produce stable and representative curves for these standard formulations. Figure 3 shows the final curves obtained using the three approaches.

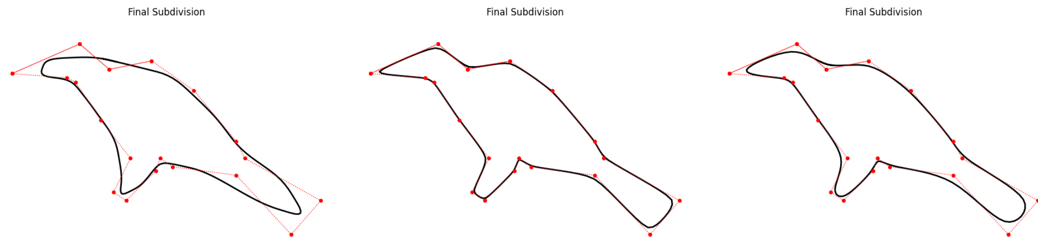

FIGURE 3. Final subdivision curves obtained with a trigonometric scheme (left), a hyperbolic scheme (center), and the proposed supervised-initialized non-uniform subdivision (right).

The trigonometric and hyperbolic schemes produce smooth curves, but their behavior depends on a single global parameter controlling the overall stiffness of the curve. Such a global parameter cannot simultaneously accommodate different local behaviors along the polygon. In contrast, the proposed approach employs spatially distributed tensions, allowing more flexible control of the refinement process while preserving the analytical structure of the scheme.

5.3. **Comparison with Classical Non-Uniform Subdivision.** Finally, we compare our approach with a classical non-uniform subdivision in which the tension parameters are manually selected. Figure 4 presents the final curves obtained by both methods.

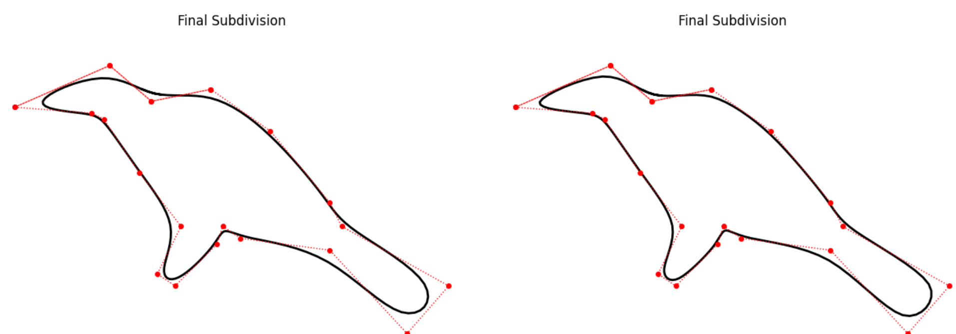

FIGURE 4. Comparison between classical non-uniform subdivision with manually selected tensions (left) and the proposed supervised-initialized subdivision (right).

The results show a strong similarity between the final curves obtained by the two approaches. This indicates that the supervised model is capable of generating tension distributions comparable to those obtained through careful manual adjustment. Unlike the conventional method, however, the proposed approach does not require manual parameter tuning. The initialization is learned from representative examples, improving reproducibility while preserving the theoretical properties of the non-uniform, nonstationary subdivision scheme.

Overall, these numerical experiments demonstrate that supervised tension initialization provides a practical alternative to manual parameter selection without altering the analytical foundations of the subdivision process.

6. Conclusions and Future Works

In this work, we proposed a supervised learning approach for the initialization of local tension parameters in non-uniform non-stationary subdivision schemes. Based on a fully connected neural network of moderate complexity, the method replaces manual parameter tuning with a learned initialization strategy derived from representative examples. The generated tension vector is directly integrated into the cubic ω-NSS subdivision scheme, while strictly preserving its analytical formulation. The numerical experiments indicate that the proposed approach produces stable and consistent tension distributions comparable to those obtained through careful manual adjustment, while improving reproducibility and reducing user intervention. Importantly, the analytical evolution and convergence properties of the subdivision scheme remain unchanged.

This framework opens perspectives for exploring alternative learning models for parameter initialization, as well as extending the methodology to surface subdivision and three-dimensional settings.

References

1. Amine Arhandou, Nour, M. Y., & Abdellah Lamnii. (2025). Rational function-based shape control for arbitrary topology surface subdivision. Gulf Journal of Mathematics, 21(2), 366-383. https://doi.org/10.56947/gjom.v21i2.3754
2. Barrera, D., Lammii, A., Nour, M.-Y., Zidna, A. A mixed A mixed hyper-bolic/trigonometric non-stationary subdivision scheme for arbitrary topology meshes.Math. Meth. Appl. Sci. 1–12 (2022). https://doi.org/10.1002/mma.8350.
3. D. Barrera, A. Lamnii, M.-Y. Nour, and A. Zidna, α-B-splines non-stationary subdivision schemes for grids of arbitrary topology, Computers & Graphics, 108 (2022), 34–48. https://doi.org/10.1016/j.cag.2022.09.004.
4. N. Dyn, D. Levin, and J. Yoon, A new method for the analysis of univariate non-uniform subdivision schemes, Constructive Approximation, 40(2) (2014), 173–188. https://doi.org/10.1007/s00365-014-9242-4.
5. D. Doo, M. Sabin. Behaviour of recursive division surfaces near extraordinary points. Computer Aided Design, 10(6) :177-181, 1978.
6. Fakhar, R., Lammii, A., Nour, M.Y., Zidna, A.: Mixed trigonometric and hyperbolic sub-division scheme with two tension and one shape parameters. Int. J. Comput. Math. 99(4),665–679 (2022). https://doi.org/10.1080/00207160.2021.1926453.
7. Fakhar, R., Lammii, A., Nour, M.Y., Zidna, A.: Mixed hyperbolic/trigonometric non-stationary subdivision scheme. Math. Sci. (2021). https://doi.org/10.1007/s40096-021-00406-4.
8. A. Lamnii, M.-Y. Nour, D. Sbibih, A. Zidna. Non-uniform subdivision schemes of B-splinecurves and surfaces with variable parameters, Computer-Aided Design, 2022. https://doi.org/10.1016/j.cad.2022.103420.
9. Lamnii, A., Nour, M.Y., Zidna, A.:A Reverse Non-Stationary Generalized B-Splines Subdivision Scheme. Mathematics 9(20), 2628 (2021). https://doi.org/10.3390/math9202628.
10. M.-Y. Nour, D. Barrera, A. Lamnii, A. Zidna, MTH subdivision scheme with sharp and semi-sharp features, journal of Computational and Applied Mathematics, 442, (2024), 115725. https://doi.org/10.1016/j.cam.2023.115725.

11. M.-Y. Nour, A. Lamnii, A. Zidna, and D. Barrera, Non-uniform UE-spline quasi-interpolants and their application to the numerical solution of integral equations, Applied Numerical Mathematics, 191 (2023), 29–44. https://doi.org/10.1016/j.apnum.2023.05.006
12. Nour, M.Y., Lamnii, A., Louzar, M., Zidna, A. (2024). A Generalized Cubic Expo-nential Non-stationary Subdivision Scheme for Arbitrary Topology Meshes. In: Sbibih,D., Remogna, S., Serghini, A. (eds) New Trends in Shape Modelling and Approxima-tion Methods. MACMAS 2023. SEMA SIMAI Springer Series, vol 37. Springer, Cham. https://doi.org/10.1007/978-3-031-62715-6_16.16.
13. Nour, MY., Lamnii, A., Lamnii, M., Zidna, A. (2024). A Non Uniform Non-stationaryMixed Trigonometric and Hyperbolic Subdivision Scheme with Variables Parameters. In:Sbibih, D., Remogna, S., Serghini, A. (eds) New Trends in Shape Modelling and Approxi-mation Methods. MACMAS 2023. SEMA SIMAI Springer Series, vol 37. Springer, Cham. https://doi.org/10.1007/978-3-031-62715-6_15.
14. J. Warren and H. Weimer, Subdivision Methods for Geometric Design: A Constructive Approach, Morgan Kaufmann, 2002.
15. Z. Wu, H. S. Seah, and F. Lin, NURBS volume for modelling complex objects, in Volume Graphics, A. E. Chen and R. Yagel (Eds.), Springer, 2000, pp. 159–167. https://doi.org/10.1007/978-1-4471-0737-8_9.
16. D. Zorin, P. Schröder, and W. Sweldens, Interactive multiresolution mesh editing, Proceedings of SIGGRAPH, 1997, pp. 259–268. https://doi.org/10.1145/258734.258863.

[1] Laboratory ISI, ENSATe, Abdelmalek Essaadi University, Tétouan, Morocco.
Email address: arhandou.amine@etu.uae.ac.ma, a_lamnii@yahoo.fr

[2] LIFAT EA 6300, University of Tours, Tours, France.
Email address: mohamedyassir.nour@gmail.com