Proceedings of the IASS 2024 Symposium
Redefining the Art of Structural Design
August 26-30, 2024, Zurich, Switzerland
Philippe Block, Giulia Boller, Catherine DeWolf,
Jacqueline Pauli, Walter Kaufmann (eds.)

Form Finding and Analyzing of Shells by Polynomial Equations and Artificial Neural Network

Abtin BAGHDADI[*a], Lukas LEDDEROSE [a], Harald KLOFT[a]

[*] Institute of Structural Design (ITE), Technical University of Braunschweig,
Pockel Str.4, 38106 Braunschweig, Germany
a.baghdadi@tu-braunschweig.de

Abstract

The article introduces an innovative form-finding approach for shell structures utilizing polynomial equations and their associated curves. This methodology, driven by four equations and nine coefficients, enables the versatile generation of diverse shell forms. Furthermore, the integration of an Artificial Neural Network (ANN) is proposed to represent parametric Finite Element (FE) software analysis, with the coefficients serving as parameters. The article illustrates various geometries generated through this approach, providing comprehensive insights into the underlying equations and coefficients. In a subsequent phase, a lookup table for the ANN is constructed, involving the generation and analysis of 20 shells with adjustments to three coefficients. The calculated eccentricity, functioning as the ANN output, underscores the tool's high accuracy. The article concludes by emphasizing the potential of the ANN tool, highlighting its ability to independently design optimal forms when interfaced with the proposed shell generator approach, incorporating geometric adjustments based on the introduced coefficients.

Keywords: Shell, Form Finding, Optimization, Artificial Neural Network, Share as Force, Polynomial Functions

1. State of the Art:

Throughout history, shell designers have applied optimization principles, initially using physical sag models pioneered by Galileo Galilei and refined by Heinz Isler. This approach found broad application, notably in Spain by Gaudí and across Switzerland, France, and Germany by Isler [**1**], [**2**]. In recent decades, advancements in software and computational techniques such as dynamic relaxation, the force density method, and best drift network analysis have enhanced precision in achieving optimal forms. Engineers have also developed techniques, aided by artificial intelligence, for optimizing structural elements such as beam design and shape search [**3**]. Moreover, sophisticated methods address challenges like geometric noise arising from optimization algorithm randomization [**5**].

However, each approach has its challenges and may not be universally applicable. Contemporary methods also have drawbacks, such as disregarding horizontal loads or requiring finite element analysis post-form finding, potentially leading to manual repetition for sub-optimal performance. Even after addressing issues like geometric noise, utilizing optimization algorithms for shaping shells requires numerous shell analyses, incurring significant computational expenses, and prompting the need for streamlined methods. In this context, a straightforward concept has been put forth, which segments a shell into distinct and manageable curves and seeks to determine the shell shape by these individual arcs and curvatures. A curve is a one-dimensional entity that can be straight, like a line, or possess curvature. Curves existing in two-dimensional space are referred to as plane curves, while those in three-dimensional space are termed space curves. In topology, a curve is delineated by a function mapping an interval of real numbers to another space. Differential geometry employs a similar definition but requires the defining function to be differentiable. Algebraic geometry delves into the study of algebraic curves, characterized as algebraic varieties of dimension one [**6**]. In contrast, the location and distribution of them through the length of each other can form the shell and surfaces [**6**]. A surface constitutes a two-dimensional entity, exemplified by shapes like a sphere or parabolic. In the realms of differential geometry and topology, surfaces are characterized by two-dimensional 'patches' (or neighborhoods) constructed via diffeomorphism or homeomorphisms, respectively. Within the domain of algebraic geometry, surfaces find their description in terms of polynomial equations, which concludes that the polynomial curves can be the roots and origins for finding the forms of the shells.

The term "Machine Learning (ML)" was coined by Arthur Samuel in the 1950s when he developed a checkers-playing program capable of independent learning [**7**]. Thanks to advancements in computer hardware and novel ML techniques, the field has achieved remarkable success in diverse areas such as image recognition [**8**], speech recognition, medical prediction, and recommendation systems [**9**]. ML excels at discerning and reconstructing complex internal relationships between input and output variables from extensive datasets. Consequently, ML can be applied to unveil correlations between a structure's shape and performance, making optimizing the structure's shape a straightforward task once these correlations are established. This concept has led researchers to employ ML in various optimization challenges across physical systems. For instance, neural network (NN) and convolutional neural network (CNN) models have been used for the inverse design of Kirigami, resulting in optimal designs that maximize elastic stretch-ability with minimal training data [**10**]. Additionally, compared to Fuzzy Logic, artificial neural networks (ANNs) have demonstrated their capabilities in the standard calculation of concrete beams [**11**].

Bayesian machine learning techniques have enabled the design and optimization of super-compressible meta-materials, transforming brittle polymers into lightweight and recoverable materials. Moreover, hierarchical composites, composed of stiff and soft materials, have been designed using neural networks, resulting in optimized patterns that significantly enhance both strength and toughness [**12**]. These studies highlight the effectiveness of ML in solving inverse problems and engineering structures with optimized performance. However, in the domain of beam optimization, existing research has predominantly focused on optimizing parameters that define specific shapes or profiles rather than thoroughly exploring the entire design space [**13**]. Similarly, ML has been utilized in designing curved beams with varying thickness distributions and optimizing three mechanical objectives.

2. Poly Nominal Forms and ANN Analyzing Approach

The study aims to define shell geometries para-metrically using polynomial curves and propose a straightforward method for developing these forms. Simple shell geometries were generated based on limited parameters and analyzed using SAP2000 to create a lookup table of geometrical parameters and forces. A neural network was then used to interpret this data for continuous form evaluation. Additionally, another mathematical approach was discussed in the final section as an outlook.

2.1. Definition of the Shell Geometries

Previous studies on remodeling Isler shells [**2**] have shown that shell forms can be developed using a limited number of arches. By optimizing the geometries of these arches based on the requirements of each part of the shell, an optimal shell form can be achieved [**6**]. However, existing approaches lack adjust-ability in the form of selected arches and shells, despite variations in arch geometries and height-to-span ratios. Therefore, this study proposes a method for para-metrically developing forms using

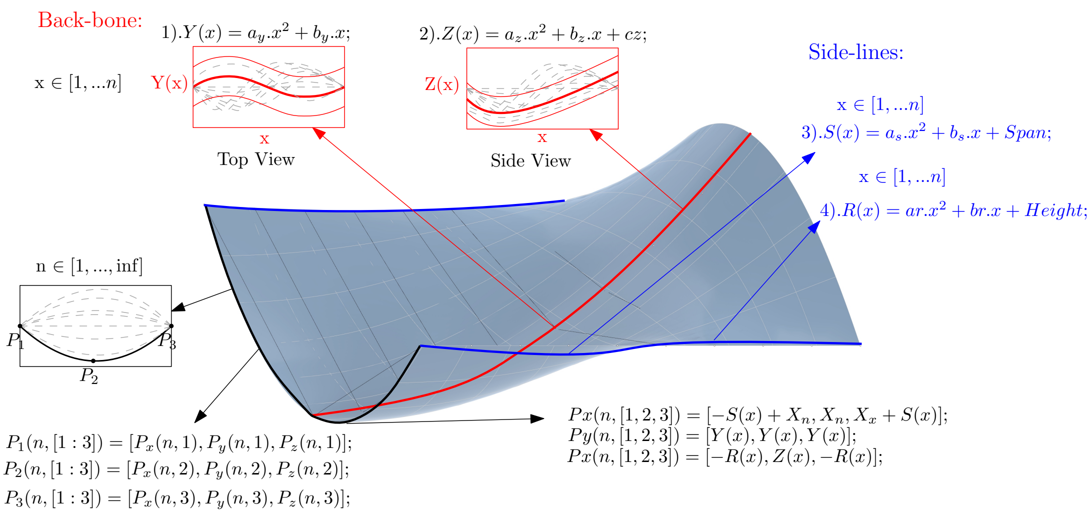

Figure 1: Parabolic equations for defining main lines (X, Y, Z) and the edges of the forms (X, S, R)

simple polynomial equations. The Polynomial-Based Form-Finding (PBF) method uses four equations and nine coefficients to define the entire shell geometry. These nine coefficients can be adjusted to generate a wide range of shell forms. The PBF can be easily calculated or coded on any platform; in this study, MATLAB was used. The PBF ultimately forms a range of arches, each based on three points and the central curvature as the backbone of the structure, as shown in Fig.(1).

- Development of the backbone curves is based on a selection of the X coordinates (x $\in [1, ...n]$), that by linspace in MATLAB for a desirable number of the arches (n) for any start and end points can be selected. The start and end points should be selected based on where the shell should be formed.

- The first polynomial was used for the backbone curvature (red line in Fig.(1) to convert the nodes to the line in X, Y space. The $Y(x)$ equations $(1).Y(x) = a_y.x^2 + b_y.x)$ using two coefficients (a_y, b_y), forms the adjustable curvature.

- Likewise, to convert the lines to 3D space $Z(x)$ was independent from $Y(x)$ and just based on the initial x coordinates developed. This second equation $(2).Z(x) = a_z.x^2 + b_z.x + cz)$, uses three coefficients $(a_z, 2 + b_z, +c_z)$, while c_z is assigned as a benchmark to start the Z coordinates, and can be based on the general plan selected as a fixed factor, to reduced the number of coordinates from 9 to 8.

- After calculation of the backbone curvature and based on the support places, the form should based on the cross line be found. On both sides of the backbone, these two lines demand an equation calculating their distances to the backbone in the X and Y spaces. As the S equation $(3).S(x) = a_s.x^2 + b_s.x + Span)$ displays this equation uses two coefficients and a fixed factor as the span, coming from the general architectural desirable span/2 in the form (blue line in Fig.(1)).

- Similarly, the heights of the side supporting blue lines were calculated by the last equation $(4).R(x) = a_r.x^2 + b_r.x + Height)$, in which (a_r, b_r) was the coefficient and $Height$ as the fixed number to apply the general desirable height were selected.

- After developing the R and S equations to have independent 3D forms of the side-lines, the nodes must be merged. It was quickly managed by addition and subtraction with the corresponding

	a_z	b_z	c_z	a_y	b_y	a_r	b_r	$Height$	a_s	b_s	$Span$		a_z	b_z	c_z	a_y	b_y	a_r	b_r	$Height$	a_s	b_s	$Span$
Sh_1	0	0	1	0	1	0	0	0	0	0	5	Sh_{13}	-2	-4	100	0	50	0	0	0	2	-4	30
Sh_2	0	0	5	0	1	0	0	0	0.2	0	5	Sh_{14}	5	-4	100	-5	50	0	0	0	2	-4	30
Sh_3	0	0	5	0	1	0	0	0	0	1	5	Sh_{15}	5	-4	100	-5	50	5	-4	0	2	-4	30
Sh_4	0.1	0	0	0	1	0	0	0	0	0	5	Sh_{16}	0	0	5	-0.1	1	0	0	0	0	0	5
Sh_5	0.1	0	0	0	1	0	0	0	0	0	5	Sh_{17}	0	0	5	-0.2	3	0	0	0	0	0	5
Sh_6	0.1	-2	5	0	1	0	0	0	0	0	5	Sh_{18}	0	0	5	-1	10	0	0	0	0	0	2.5
Sh_7	0	0	5	0	1	0.2	0	0	0	0	5	Sh_{19}	0.2	0	5	-1	10	0	0	0	0	0	2.5
Sh_8	0	0	5	0	1	-0.3	2	0	0	0	5	Sh_{20}	0.3	-1	5	-1	10	0.2	-2	0	0	0	2.5
Sh_9	0.1	-2	5	0	4	-0.3	2	0	0	0	5	Sh_{21}	0.3	-1	5	-1	10	0.2	-2	0	0.2	1	2.5
Sh_{10}	0.1	-2	5	0	4	-0.3	2	0	0	0	10	Sh_{22}	0.3	-1	5	-1	10	0.2	-2	0	0.2	-1	2.5
Sh_{11}	0.1	-2	5	0	4	-0.3	2	0	0.3	-0.4	10	Sh_{23}	0.6	-1	5	-1	10	0.2	-10	3	-0.4	5	5
Sh_{12}	-2	-4	100	0	50	0	0	0	0	0	30	Sh_{24}	5	5	5	5	5	5	5	5	5	5	5

Table 1: Selected coefficients for parametrically forming the shells

nodes in the backbone. To make the $P(x, y, z)$ matrix for each n point, Fig.(1). The 9×9 matrix can be reshaped to prepare the coordinates of three points in each arch to be connected to gather (e.g. by spline).

The array of individual arches forms the parametric shells, which, through four equations and nine coefficients, can generate a wide range of adjustable shells. The coefficient equations can be increased or reduced to cover various forms or simplify them. Table (2.1.) displays different selected coefficients, gradually changed to demonstrate their influence on the forms. The geometries start with the simplest, with the minimum essential parameters, and then the parameters are gradually adjusted, and the number of selected ones increases. In creating these geometries, the final P matrix, including the X, Y, Z coordinates of three points for each arch, was transferred to Rhino3D. The dimensions of the matrix depend on the selected number of arches (n), and increasing the complexity of the geometry makes higher values of n more desirable. The minimum matrix size in these models was 3×30. In Rhino3D, spiles connected the joints, and a surface was added.

Figure. (2), displays the geometries with the corresponding coefficients in the Table. (2.1.). It can be seen that all $'a'$, group (a_z, a_y, a_r, a_s), cause the curvature in the lines, and if the amount of them is equal to zero, the structure of the shell will be formed by lines, with specific angles. The $'a'$, group (b_z, b_y, b_r, b_s), defines the amount of these angles. If they are all zero, the forms are made by straight lines. In a specific range, if the $'b'$ in each equation has a higher amount with different signs than $'a'$ (e.g. $b : 5$ and $a : -0.2$), the turning point can be formed in all directions. The selected equation type was polynomial, chosen for its accessibility and ease of description. However, various types of two-dimensional functions, such as analytical, harmonic, trigonometric, exponential, etc., can be utilized in similar formats. Similarly, direct 3D functions can be employed. While using 3D functions may involve fewer coefficients, generating a wide range of geometries while respecting architectural components and limitations (e.g. heights and spans) is not always straightforward or feasible.

2.2. Neural Network for analyzing the forms:

One of the known issues in most developed form-finding approaches is the challenge of finding appropriate forms and simultaneously analyzing them to calculate forces and eccentricity for identifying optimum forms. Forms with low eccentricity are typically considered optimal. As described in the previous section, PBF can generate a wide range of forms, necessitating an analysis method. A classic solution is to write equilibrium equations, which have largely been replaced by Finite Element methods in subsequent steps. This requires an interface between the coding platform (e.g., MATLAB) and structural software (e.g., SAP2000) to enable real-time analysis [**7**]. Additionally, the potential of machine learning techniques, such as Fuzzy systems [**14**] or Neural Networks, can be explored as alternatives to traditional structural software. Neural Networks were implemented using MATLAB. The Network

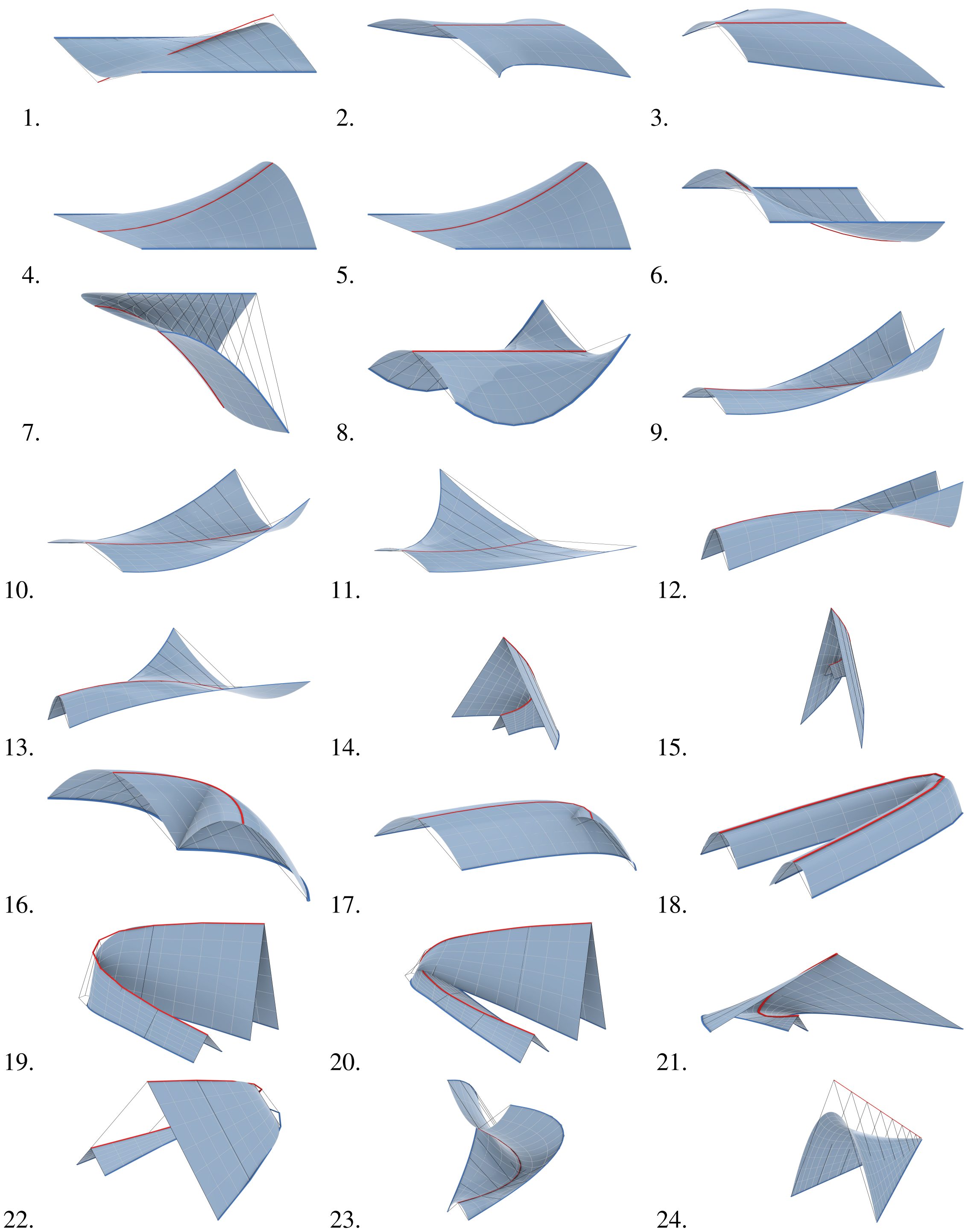

Figure 2: Some of the generated force based on the poly-nominal curves regarding the selected coefficients, Table. (2.1.)

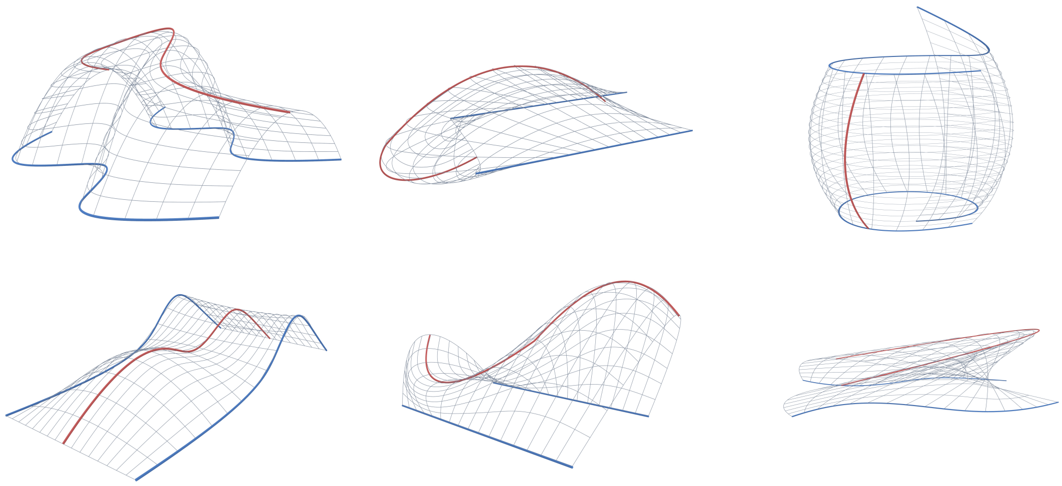

Figure 3: Possible geometries by PBF

configuration involved selecting activation functions, including Sigmoid ('$sigmoid$') and Hyperbolic tangent Sigmoid ('$tansig$'), along with the Linear function ('$purelin$') for input transformation. These functions, along with inputs, targets, and the number of hidden layers (NHL), were assigned to the Network using the '$newff$' command.

Furthermore, procedures were implemented to remove rows with constant values ('$removeconstantrows$') and mapping rows with minimum and maximum values ('$mapminmax$'). Random data division and sample selection ('$dividerand$' and '$sample$') were employed, allocating 75% of the data for training and an equal percentage for validation and testing of the Network ((25%)). To facilitate the training process, the '$Levenberg - Marquardt$' ('$trainlm$') algorithm and Mean Squared Error ('mse') were designated as the Network's performance functions.

To utilize the Neural Network (NN), a lookup table containing data about the parametric shell geometries and their corresponding analysis results must be prepared for interpretation. Since the NN is intended to represent the Finite Element (FE) software for analyzing the structures and calculating the eccentricity, the lookup table should include the analysis results of the shells as well. To prepare the lookup table, 20 parametric geometries were generated by selecting and adjusting three of all the coefficients ($a_z, a_y, \&b_z$), as shown in Fig.(4). The P matrix, including the x, y, z data, was exported to SAP2000, and each section was analyzed accordingly. The forces, including bending and axial forces in each section, were calculated. For the analysis output, the maximum eccentricity (abs(bending/Axial)) along with its standard deviation was selected. The coefficients $a_z, b_z, \&a_y$ were varied with equal steps from zero to $0.3, -2, -0.2$ respectively, as shown in Table (2). If the number of coefficients increases significantly, the evaluation of higher shells and more analysis output would be necessary. However, since the aim is to propose a new approach, a limited number of shells were analyzed to demonstrate the feasibility of the method.

Table (2) displays the results of the maximum bending (M) and shear (V) divided by axial load, along with the eccentricity (E_{cc}) and the standard deviation (Std) of them in all arches of each shell. These results are based on the variations of $a_z, b_z\&$ a_y. The table serves as an outlook table and will be interpreted by the Artificial Neural Network (ANN).

Figure (5) illustrates some of the results of the Artificial Neural Network (ANN) for interpreting the lookup table based on the analysis results of 20 shells in SAP2000. The selected coefficients are shown, and those not displayed are assumed to be zero. The comparison between the analysis results and the

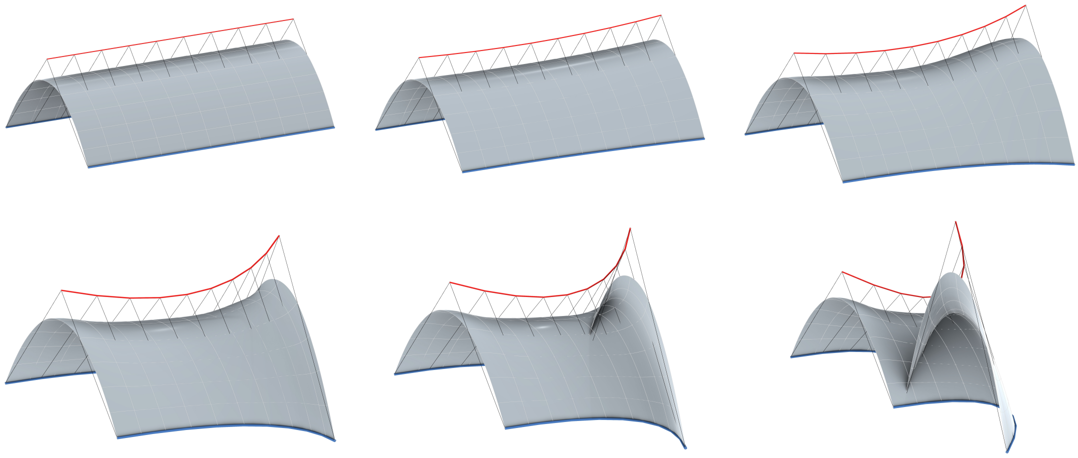

Figure 4: SAP 1-20 (Models: 1,8,12,16,20)

ANN's diagrams demonstrates the high capacity of the ANN in representing the Finite Element (FE) software. This is evident in both the regression diagrams and the overlaying of the analysis results. However, there may be discrepancies in the currency between the shear and bending diagrams, as well as between the deviations and maximum forces. It's important to note that despite being coded together, each output has an individual operation in the ANN. Since the complexity of the data varies and the ANN is a powerful tool for any type of regression, selecting a high number of hidden layers for simple operations may lead to over-fitting. Increasing the number of shell evaluations and the size of the lookup table, along with individual adjustments, can improve accuracy.

In conclusion, the integration of an artificial neural network (ANN) with the Polynomial Based Form-

-	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
E_{cc} **M**	2.96	2.96	3.26	3.58	3.90	4.21	4.21	5.03	5.03	5.25	7.34	9.78	8.01	5.33	5.36	5.36	5.47	5.50	5.57	6.05
E_{cc} **V**	1.47	1.47	1.56	1.65	1.73	1.81	1.81	1.95	1.95	1.97	2.82	3.45	2.75	1.84	1.74	1.74	1.68	1.65	1.63	3.49
Std M	1.08	1.08	1.12	1.16	1.20	1.24	1.24	1.34	1.34	1.36	1.47	1.67	1.55	1.41	1.37	1.37	1.35	1.34	1.32	1.55
Std V	0.65	0.65	0.67	0.68	0.70	0.71	0.71	0.74	0.74	0.74	0.78	0.81	0.77	0.73	0.71	0.71	0.69	0.68	0.66	0.84

Table 2: The maximum bending (M) and shear (V) divided by axial load along with the eccentricity (E_{cc}) and the standard deviation (Std)

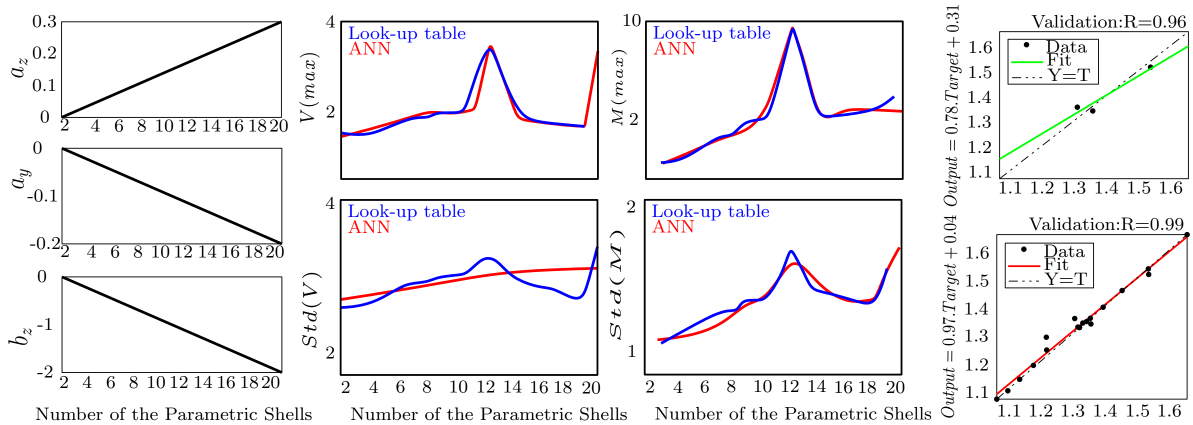

Figure 5: The performance of the ANN in comparison to the analyzing data (Look-up table) and training regression of the maximum forces

Finding (PBF) approach enables the analysis of parametric shells generated by PBF. The key achievement of this integration is the development of an independent tool that can form and analyze parametric shells in a systematic m ANNer. With sufficient shell analysis, this tool allows operators to manipulate the geometry and observe the resulting eccentricity (E_{cc}), aiding in the identification of coefficients that produce architecturally desirable shapes with minimal eccentricity, thus achieving an optimum form. The ANN serves to convert discrete analysis results from a limited number of shells into a continuous area, allowing for the exploration of unanalyzed results through manipulation of PBF coefficients. Evaluating a wide range of shells and saving the data from both PBF and ANN as two capable tools can lead to the full development of a new tool ($PBFA$), addressing the complexity of form-finding and overcoming any experienced issues. ANNs as operators can also be interfaced to an optimization algorithm on the same platform (such as MATALB). Then, the optimizer can find the optimal form by adjusting the defined polynomial parameters.

3. Principles of a form finding approach (Out-Look):

This section proposes an idea for further study on determining the form of a shell. The main question raised is: Can the force in shells be considered analogous to the flux in a surface? Before delving into this concept, let's define the general function of a surface in 3D space.

$$4D \rightarrow f : Ax^3 + By^3 + Cz^3 + Dw^3 + Ex^2yzw + Fxy^2zw + Gxyz^2w + Hxyzw^2 + Ix^2yzw \quad (1)$$

$$+Jxyz + Kxyw + Lyzw + Mwyx + Nxy + Oxz + Pxw + Qyz + Ryw + Szw + Tx + Uy + Vz + V1w + V2 = 0$$

Assuming the material as the constant property in the shells (e.g. concrete), the shell's geometries can be defined as the 3D surface.

$$3D \rightarrow f : Ax^2 + By^2 + Cz^2 + Dxy + Exz + Fyz + Gx + Hy + Iz + J = 0 \quad (2)$$

$$Z(x; y) = -(x^2 + y^2) \quad (3)$$

3.1. Form Finding as a Math problem

To efficiently find the form of shells, a mathematical approach can be employed. Rather than traditional methods, a plan from the R^2 space must be mapped to the R^3 space using a well-defined function to achieve an optimal form. This function can be defined in various spaces, such as Euclidean or Topological. It must convert the nodes of the initial shape (from (x, y) to (x, y, z)) into the optimal form.

$$R_i^2 \underrightarrow{F} R_i^3 \quad (4)$$

It would be beneficial to employ flux definitions applied in vector space within suitable analytical assumptions on a manifold topological space. This approach enhances properties crucial for mappings, such as form and dimension, and allows for the addition of new dimensions like nodal angle and their relationships. Principal topological properties like connectedness must be considered, along with other properties such as compactness, matrix capability, and boundary properties. The function and its inverse should ensure continuity to prevent separation and jumps, particularly useful when architects seek separated shells adjacent to each other, requiring asymptotic analysis. Additionally, the function should be differentiable, allowing for partial derivations, which provide effective properties over a surface in vector spaces (shared definitions). For instance, considering the amount of inertia and its mapping on Earth from each point could be insightful. Understanding the relationship between strain energy, stress, and work, mathematically calculating the work done by load vectors along a curvature or shell, and considering the energy in the gravity direction towards shell nodes could inform stress calculations under pressure. This approach, leveraging conservative gravity performance, could potentially maximize stress based on the stated goal[1].

$$F_i(x,y,z) = \frac{\iiint_R Z^2(\rho(f(x,y,z), t_f(x,y,z)dv}{f(x,y) \to m_f(x,y)} \tag{5}$$

It is internally related to shear[2], but this time, the question is about the amount of in-plane load within a shell section. This can be calculated using the Gauss-Divergence and Stokes theorems.

$$Share : \iint_\delta \rho(f(x,y,z), t_f(x,y,z) \times V(x,y,z) \times n(x,y,z)d\sigma \tag{6}$$

The Share definition is used in temperature transmission in shells with the same in-plane type of load. It aims to define the function that can map to a new shell shape with the maximum defined Share type, where:

$$R =\subseteq X(\tau, \Sigma, \mu)\&i \in I, I = \{0,1,2,...,n)\}, \alpha = (\alpha_1, \alpha_2, \alpha_3, ..., \alpha_n) \tag{7}$$

Where, n: vector perpendicular to the surface, V: speed function for passing the area, F: load vector, r: direction function, ρ: mass function, f: shape function, t_f: thickness function, m_f: mapped function.

4. Conclusion:

The article introduces a novel form-finding approach for shaping shells by integrating curves derived from polynomial equations. This method utilizes four equations and nine coefficients to generate diverse shell forms. Additionally, an Artificial Neural Network (ANN) is implemented to represent the analysis within Finite Element (FE) software. After explaining the equations and coefficients, various geometries produced using this approach are presented. However, the demonstrated shells represent only a limited range of what this approach can produce; the Polynomial Based Form-Finding (PBF) method can generate a wider variety of shells. Furthermore, employing other types of functions or series instead of polynomials in a similar approach could lead to an even broader range of shells. To analyze the parametric shells, a creative approach through ANN was utilized, and the coding features in MATLAB were described. Since approaches like Fuzzy Logic, ANN, and Adaptive Neuro-Fuzzy Inference System (ANFIS) require initial data about the parameters and calculation output of the corresponding problem, a comprehensive database should be built.

To prepare the Look-up table for export to the ANN, 20 shells were generated and analyzed by adjusting three coefficients. The resulting eccentricity, serving as the output of the ANN, demonstrated remarkably high accuracy. While this initial study utilized a small dataset, practical applications will necessitate a larger number of shells and a correspondingly expanded Look-up table, along with fine-tuning of the ANN. MATLAB was employed for code development, but exploring other programming languages like C_{++} and developing independent software could be considered for future steps. The

[1]The gravity performance is a conservative function.
[2]The amount of force passing through a unit area.

interface of PBF and ANN represents a comprehensive approach ($PBFA$), showcasing the effectiveness of this tool in independently designing optimal forms, particularly when coupled with the proposed shell generation approach.

References

[1] González, Genaro and Samper, Albert and Herrera, Blas, Classification by type of the arches in Gaudi's Palau Güell, Nexus Network Journal, V.20, P.174–186, 2018, Springer

[2] Baghdadi, Abtin and Heristchian, Mahmoud and Kloft, Harald, Structural assessment of remodelled shells of Heinz Isler, International Journal of Advanced Structural Engineering, V.11, 2019, Springer

[3] Nimtawat, Anan and Nanakorn, Pruettha, A genetic algorithm for beam–slab layout design of rectilinear floors, Engineering Structures, V.32, N.11, P.3487–3500, 2010, Elsevier

[4] Baghdadi, Abtin and Heristchian, Mahmoud and Kloft, Harald, A semi-supervised meta-heuristic shell form-finding approach, Iranian Journal of Science and Technology, Transactions of Civil Engineering, V.44, P.1031–1038, 2020, Springer

[5] Mumford, David, The red book of varieties and schemes: includes the Michigan lectures (1974) on curves and their Jacobians, V.1358, 1999, Springer Science & Business Media

[6] Samuel, Arthur L, Some studies in machine learning using the game of checkers, IBM Journal of research and development, V.3, N.3, P.211–229, 1959, IBM

[7] Baghdadi, Abtin and Heristchian, Mahmoud and Kloft, Harald, Arch-based form-finding technique in Design of optimum shell structures, BTProceedings of IASS ANNual Symposia, V.2020, N.22, P.1–12, 2020, International Association for Shell and Spatial Structures (IASS)

[8] Elkahky, Ali Mamdouh and Song, Yang and He, Xiaodong, A multi-view deep learning approach for cross domain user modeling in recommendation systems, BTProceedings of the 24th international conference on world wide web, P.277–288, 2015

[9] Hanakata, Paul Z and Cubuk, Ekin D and Campbell, David K and Park, Harold S, Accelerated search and Design of stretchable graphene kirigami using machine learning, Physical review letters, V.121, N.25, P.255304, 2018, APS

[10] Baghdadi A, Babovic N, Kloft H. Fuzzy Logic, Neural Network, and Adaptive Neuro-Fuzzy Inference System in Delegation of Standard Concrete Beam Calculations. Buildings. 2023 Dec 20;14(1):15.

[11] Gu, Grace X and Chen, Chun-Teh and Richmond, Deon J and Buehler, Markus J, Bioinspired hierarchical composite Design using machine learning: simulation, additive manufacturing, and experiment, Materials Horizons, V.5, N.5, P.939–945, 2018, Royal Society of Chemistry

[12] Enns, YB and Pyatishev, EN and Glukhovskoy, A, Bistable arch-like beams with modulated profile as perspective supporting structures of a microelectromechanical actuator, BTJournal of Physics: Conference Series, V.917, N.8, P.082013, 2017, IOP Publishing

[13] Baghdadi, A. and Meshkini, A. and Kloft, H. Parametric Design of in-plane concrete dry joints by FE method and Fuzzy logic toward utilizing additive manufacturing technique, IASS ANNual Symposia, V.2020, N.21, P.1–12, International Association for Shell and Spatial Structures