Proceedings of the Twelfth International Conference on
Engineering Computational Technology
Edited by: P. Iványi, J. Kruis and B.H.V. Topping
Civil-Comp Conferences, Volume 8, Paper 2.1
Civil-Comp Press, Edinburgh, United Kingdom, 2024
ISSN: 2753-3239, doi: 10.4203/ccc.8.2.1

PoliBrick: A Plug-In to Generate Stereotomy in Double Curvature Masonry Vaults

N. Pingaro, M. Pourfouladi and G. Milani

Department of Architecture, Built Environment and Construction Engineering, Politecnico di Milano Italy

Abstract

This paper focuses on the creation of a new Grasshopper's plugin within the Rhino software, specifically aimed at facilitating the modeling of diverse brickwork shells with complex designs, particularly focusing on brick arches. The plugin, dubbed PoliBrick, distinguishes itself by simplifying parametric modeling across a wide array of free-form shells, which proves invaluable for crafting intricate brick arch patterns. Its operational approach stands out from existing methods by enabling parametric modeling across all free-form shells. With a keen emphasis on user accessibility, PoliBrick is designed to feature a user-friendly interface, equipped with just six essential tools to streamline the modeling process. The primary objective of this endeavor is to establish a real-time, feedback-driven design revision method, empowering users to effortlessly generate various brick patterns, including basket weave, stretcher bond, herringbone, and more, while also seamlessly integrating post-analysis for the structural investigation of curved structures by integrating the tool into software of varying complexity, employing preferred methodologies such as finite or distinct elements. Through PoliBrick, a novel approach emerges, revolutionizing the way brick patterns are conceptualized and implemented within architectural design workflows. The procedure is validated through benchmarking against various types of ancient vaults and dome patterns selected by the authors for which the robustness of the proposed algorithm is confirmed.

Keywords: PoliBrick, Masonry, Bricklaying, Brick Pattern Modelling, Parametric Modelling, Single and Double Curvature Vaults.

1 Introduction

In the world-built heritage, masonry arches and vaults are long pivotal features of masonry structures, and they are evolved over time to accommodate various spatial, architectural, and load-bearing needs [1]. Buildings made by bricks are the predominant construction method for many centuries and curved structures emerged as crucial components in historical buildings thanks to the fact that they are conceived to support large vertical loads [2]-[3].

Stereotomy in masonry constructions refers to the practice of designing and constructing masonry structures, such as arches, vaults, and load-bearing walls, in a way that optimizes the arrangement of individual elements (such as bricks or stones) to ensure uniform load distribution and optimal structural stability. This concept is particularly important in vaulted structures, where the arrangement of bricks or stones can significantly influence the ability of structures to support vertical loads and effectively distribute them along the arch or vault. To apply stereotomy in such constructions involves to set up the three-dimensional geometry of individual elements and their interactions/disposition to achieve a robust and stable structure under gravitational loads.

In this regard, recent research highlights the significant role of brick patterns, or masonry textures, in shaping the behaviour of vaulted structures. Researchers are focused on understanding how micro-geometry influences the overall three-dimensional structural behaviour of these structures [4]. Various analytical methods, such as limit and finite element analysis, are employed to unravel the collapse mechanisms in historical masonry arches and vaults **Hiba! A hivatkozási forrás nem található.**-[9].

Not only are new research efforts aimed at understanding how historical structures with single or double curvature behave by altering the arrangement of bricks, but also in the application of masonry vaults in contemporary architecture. In particular, institutions like ETH Zurich, Georgia Institute of Technology, and Harvard University are leading efforts to integrate digital tools with robotic fabrication methods [10]-[14]. These endeavours aim to streamline design and construction processes, allowing designers to create intricate brick patterns with unprecedented precision.

In this context, the proposed paper presents a new approach to automatize brick patterns in vaults. The developed parametric procedure is implemented in Grasshopper (Rhinoceros) by developing a new plug-in called PoliBrick. PoliBrick allows researchers to obtain any curved geometry, from the simplest to the most complex ones, with the desired brick disposition such as basket weave, stretcher bond, and herringbone bond. Such new tool reduces in a considerable manner the computational burden in the pre-structural analysis modelling phase and allows designers to consider the non-negligible role of stereotomy in curved structures. The article is organized as follows: section 2 provides a detailed description of the algorithm used in the parametrization procedure; section 3 presents the benchmarking of the approach on various types of vaults already present the in existing literature;

and finally, section 4 outlines conclusions drawn from the study and suggests potential avenues for future development of the approach.

2 Parametrization

This plugin introduces a straightforward method for modeling various brick patterns. It begins by outlining curves on the surface to connect individual bricks, forming a repeating pattern. The initial step involves defining multiple curves, resembling threads, which represent specific types of bricks. To achieve this, the surface is either cut with a plane or projected with a specific curve to create the initial curve, see for instance Figure 1.

The subsequent step involves offsetting or copying the curve on the surface, depending on certain conditions. Since brick pattern vaults typically feature parallel threads of bricks, the curve is offset on the surface by a set distance to maintain this pattern. To introduce variations in the curve direction, patches are defined, which represent subsurfaces of the main surface. These steps enable the creation of diverse brick patterns within the plugin's framework, as illustrated in Figure 1.

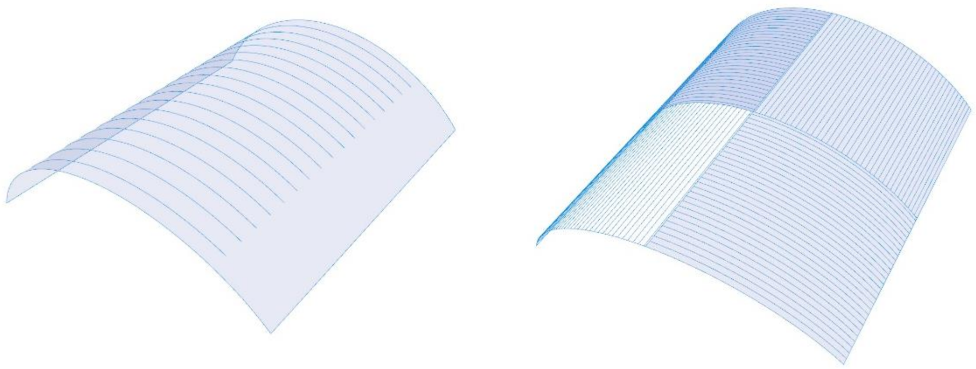

Figure 1: creating subsurface and base curves on the surface

The next step involves specifying the characteristics of the bricks within each thread. This task is carried out using a component called Brick Info within the plugin. Here, users define each brick by assigning a designated character, such as A, B, C, etc., which encapsulates the relevant data associated with that specific brick. Users are required to input the brick's name (an English alphabet character), its dimensions (u, v, and w), which align with the surface's u and v directions, and the desired dimension aligned with the surface's normal. To simplify input, users can combine these parameters with the mortar distance (another characteristic) of the brick with the next brick on the same thread into a single comma-separated text format (u, v, w, m). The Brick Info component then interprets this data accordingly.

Once the dimensions are defined, a mesh box is created to represent each brick at the origin point of the Cartesian coordinate system within the software. To move each brick from the origin to its intended position on the surface, certain considerations must be made. Firstly, the normal of the lower face of the brick should align with the surface's normal vector at the specified point. Secondly, the central point of the brick's face should coincide with the surface. To achieve this transformation, the following formula must be followed [15]:

$$\{R\} = \{N_b\} \times \{N_s\} = \begin{vmatrix} \hat{\imath} & \hat{\jmath} & \hat{k} \\ N_{b,x} & N_{b,y} & N_{b,z} \\ N_{s,x} & N_{s,y} & N_{s,z} \end{vmatrix} = \begin{Bmatrix} N_{b,y}N_{s,z} - N_{b,z}N_{s,y} \\ -N_{b,x}N_{s,z} + N_{b,z}N_{s,x} \\ N_{b,x}N_{s,y} - N_{b,y}N_{s,x} \end{Bmatrix} \quad (1)$$

$$cos\ \alpha = \{N_b\} \cdot \{N_s\} = N_{b,x}N_{s,x} + N_{b,y}N_{s,y} + N_{b,z}N_{s,z} \quad (2)$$

To orient a brick at a specific point on the surface, we calculate the rotation axis {R} and angle (α) based on the normal vectors of the brick $\{N_b\}$ and the surface $\{N_s\}$.This ensures that the brick aligns with the curve and normal of the surface. After this orientation, the brick needs to be moved from its origin to the designated point on the surface. Users may introduce additional transformations to accurately position the brick on the surface after the initial transformation. To streamline this process, all transformations can be implemented together. The secondary transformation of a brick on the surface is defined by another set of inputs, including:

- α: rotation around the surface normal axis;
- β: rotation around the tangent axis of the surface perpendicular to the curve;
- δ: displacement perpendicular to the curve;
- ε: displacement in the direction of the curve.

By combining these four parameters into one statement (α, β, δ, ε), we have five transformations to implement after the initial movement from the origin point. This includes two rotations and three displacements. Using a [4x4] matrix for transformations, known as a homogeneous coordinate system, instead of [3x3] matrices, allows us to represent all transformations, including translation. The general format for a translation matrix is:

$$v = \begin{bmatrix} 1 & 0 & 0 & a1 \\ 0 & 1 & 0 & a2 \\ 0 & 0 & 1 & a3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

To translate a point P (x, y, z) by a vector v< $a1$, $a2$, $a3$>, a transformation matrix is used. For instance, the following equation is derived:

$$P' = \begin{bmatrix} 1 & 0 & 0 & a1 \\ 0 & 1 & 0 & a2 \\ 0 & 0 & 1 & a3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} (1 * \mathrm{x} + 0 * \mathrm{y} + 0 * \mathrm{z} + \mathrm{a1} * 1) \\ (0 * \mathrm{x} + 1 * \mathrm{y} + 0 * \mathrm{z} + \mathrm{a2} * 1) \\ (0 * \mathrm{x} + 0 * \mathrm{y} + 1 * \mathrm{z} + \mathrm{a3} * 1) \\ (0 * \mathrm{x} + 0 * \mathrm{y} + 0 * \mathrm{z} + 1 * 1) \end{bmatrix} \quad (4)$$

In the following part it is demonstrated how to calculate a rotation around the z-axis from the origin [16]. This process uses trigonometry to create a matrix that represents rotation. Let's imagine a point P (x, y) on the x-y plane rotating by an angle (b), from which, the following expressions are derived:

$$x = d\cos(a)$$
$$y = d\sin(a) \tag{5}$$

After the rotation, the new coordinates (x', y') are described:

$$x' = d\cos(b + a)$$
$$y' = d\sin(b + a) \tag{6}$$

Expanding x' and y' using trigonometric identities, which involve the sine and cosine of the sum of angles, results in:

$$x' = d\cos(a)\cos(b) - d\sin(a)\sin(b)$$
$$y' = d\cos(a)\sin(b) + d\sin(a)\cos(b) \tag{7}$$

By substituting equations (5), the following expressions can be written:

$$x' = x\cos(b) — y\sin(b)$$
$$y' = x\sin(b) + y\cos(b) \tag{8}$$

The rotation matrix around the z-axis can be shown as:

$$Rz = \begin{bmatrix} cos(b) & -sin(b) & 0 & 0 \\ sin(b) & cos(b) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

The rotation matrix around the x-axis by angle b looks like:

$$Rx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(b) & -sin(b) & 0 \\ 0 & sin(b) & cos(b) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

And finally, the rotation matrix around the y-axis is reported:

$$Ry = \begin{bmatrix} cos(b) & 0 & sin(b) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(b) & 0 & cos(b) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

The same principle applies to other axes as well. Sometimes, users may not want to apply specific translations or rotations to all bricks of the same type in a brick pattern. In such cases, they can use additional input parameters called rotation pattern and displacement pattern. These patterns are represented by a text string containing "T" for true (to implement the transformation) and "F" for false (to not implement the transformation). For example, a pattern like "TFF" indicates that all bricks of that type should repeatedly follow the specified transformation pattern, as illustrated in Figure 2.

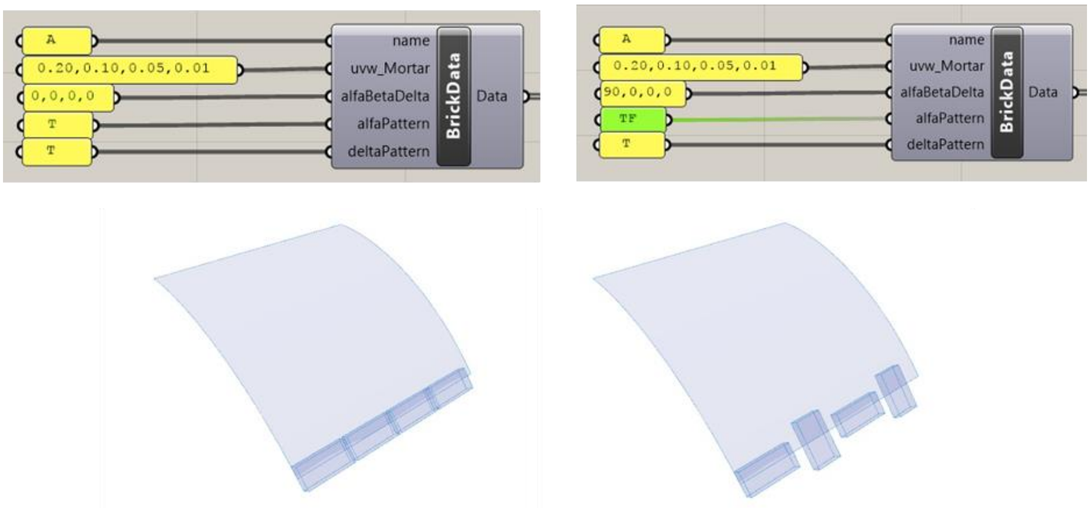

Figure 2: the process of implementing pattern for rotation

Once the user has provided all the necessary data, it is compiled for each type of brick and exported from the component as an output (data).

The aforementioned “Brick Info” component defines the characteristics of each type of brick and compiles them into output data. Multiple “BrickInfo” components can be used to define various brick types. The next step involves introducing the software, which allows users to specify the combination of bricks they wish to use to form a row. This task is carried out by the “pattGenerator” component, which combines different characters corresponding to the name of each type of brick to create a string of brick characters. The inputs for this component include character, repeat, and plus. Repeat denotes the number of times each brick type is repeated in the row, while plus is a tool used to combine different strings. Additionally, it is worth noting that sometimes it is simpler to define an empty brick with its characteristics rather than solely adjusting mortar dimensions. For this purpose, the character "S" is designated to represent an empty brick, as illustrated in Figure 3.

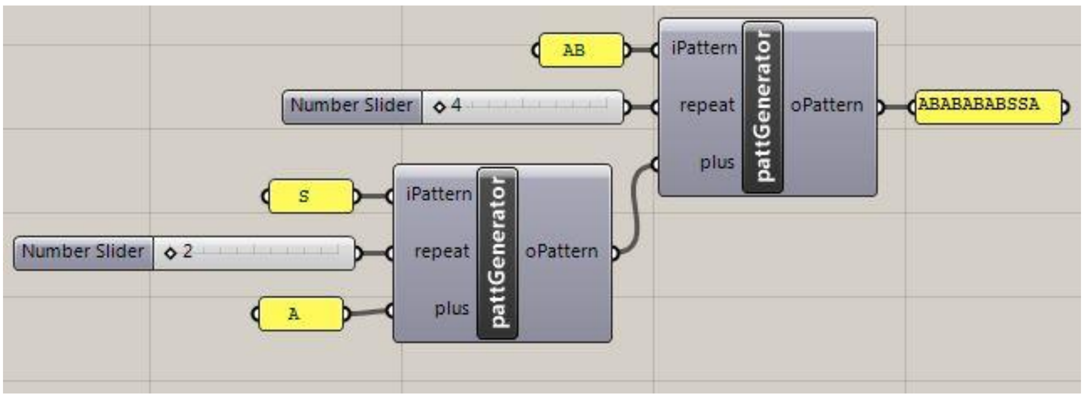

Figure 3: the process of generating a string as a pattern

By defining the strings and characteristics of each element within a string, all the bricks on the surface can be specified. The next component is used to include these bricks on the surface. This process of transforming bricks is executed within the solver. It takes input curves, input surfaces, patterns, and brick data as inputs, and produces the brick mesh as output. The solving procedure begins by dividing the curve into arches to locate points where each block is positioned and aligned with the local normal vector, as depicted in Figure 4.

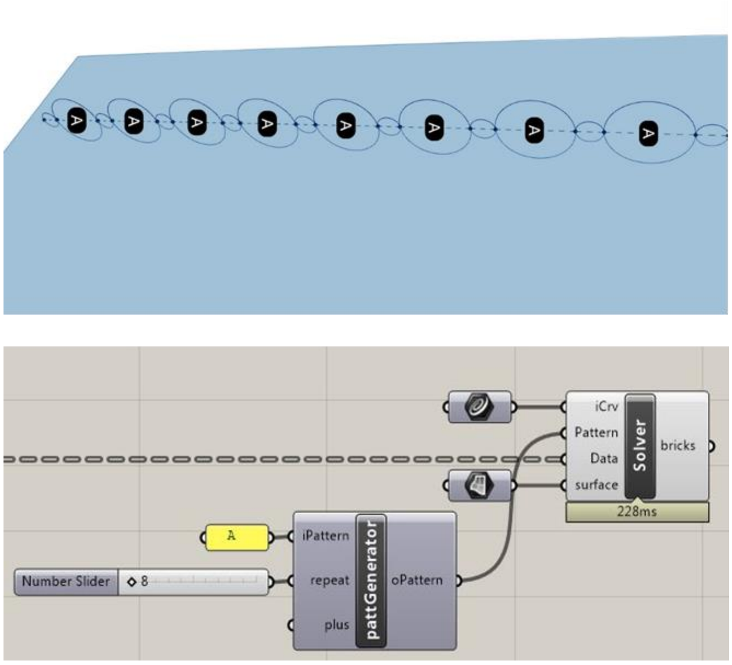

Figure 4: segmenting curves by solver to place blocks on them.

After generating meshes, it's important to inspect their borders, as some may extend beyond the surface edge. Users have the option to decide whether to cut these meshes at the border. To facilitate this decision, users can input the index of the surface border, which will be highlighted for them to review using the cutter component. Another modification process involves handling intersections between bricks. Instead of being part of the solver, this task is now performed by the “cullIntersection” component. This separation is necessary because intersections may occur due to incorrect dimension definitions. To identify intersections and check for blocks crossing specific lines, a loop is executed on all brick meshes exported from the solver. To improve efficiency, the RTree algorithm is employed to quickly find the nearest brick to the border and each block (see Figure 5). This algorithm groups nearby objects and represents them with their minimum bounding rectangle, which helps accelerate the search for the nearest neighbor at various distances [17].

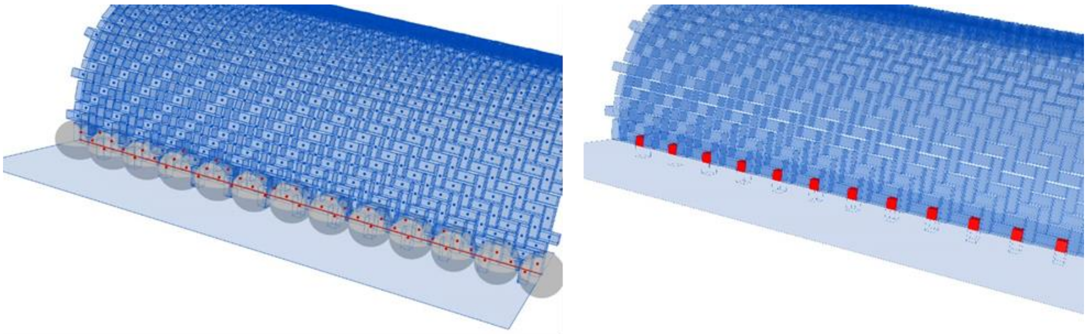

Figure 5: partition a curve into segments based on a specified number, aiming to identify the closest blocks that may require cutting at the surface border.

3 Modeling Specific Vaults and Domes

This paper explores various brick patterns, as for instance herringbone bond, basket weave, and running bond to understand how they affect the load-bearing capacity, stiffness, and overall stability of vaults. Different patterns distribute loads differently, impacting on how stresses are transmitted within the structure [18]. Recent numerical research emphasizes the significant influence of brick patterns, especially the micro-geometry, on the overall three-dimensional behavior of vaulted structures [19]. However, there's a noticeable absence of experimental studies addressing this matter, highlighting the need for further investigation.

Various authors [20]-[23] discuss prevalent pattern configurations for different types of vaults and when it's advantageous to choose one pattern over another. For barrel vaults, manuals provide examples of laying methods. The simplest involves longitudinal row arrangement, but it's unsuitable for wide-span, low-rise vaults because key joints are nearly vertical and parallel. In such cases, bricks are oriented at a 45° angle relative to plan edges for better structural integrity. This paper selects two patterns from previous studies [1], one of which is more complex to model. The first pattern arranges courses at a 45° angle with respect to plan edges, while the second employs a herringbone configuration.

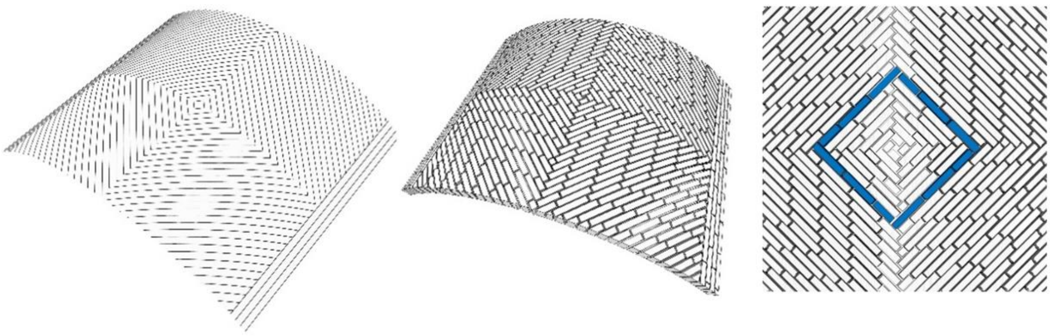

Figure 6: barrel vault which courses at a 45° angle with respect to plan edges

In the provided sample in Figure 6, one instance of modelled barrel vaults is depicted, where the brick arrangement is oriented at a 45-degree angle relative to the edge. To model the brick bonding lines on the vault, they were drawn on the vault.

However, as indicated by the blue circle, in each row, the bricks extend from one quadrant of the vault to another, and it is not possible to model one section and mirror or rotate it. Instead, brick rings in the four quadrants of the vault need to be considered together for modeling.

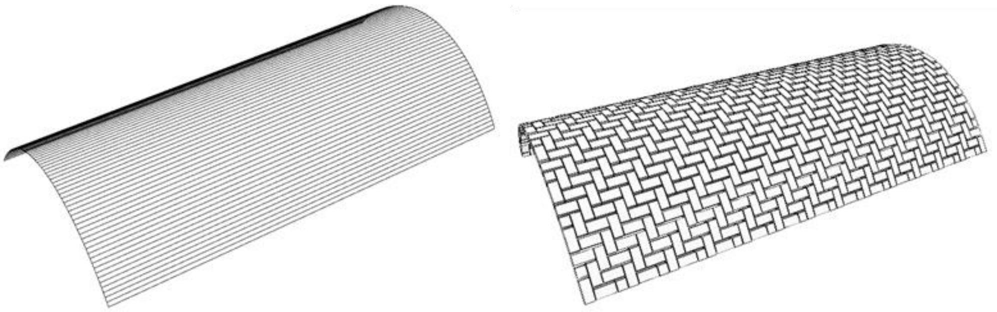

Figure 7: barrel vault which courses a herringbone pattern

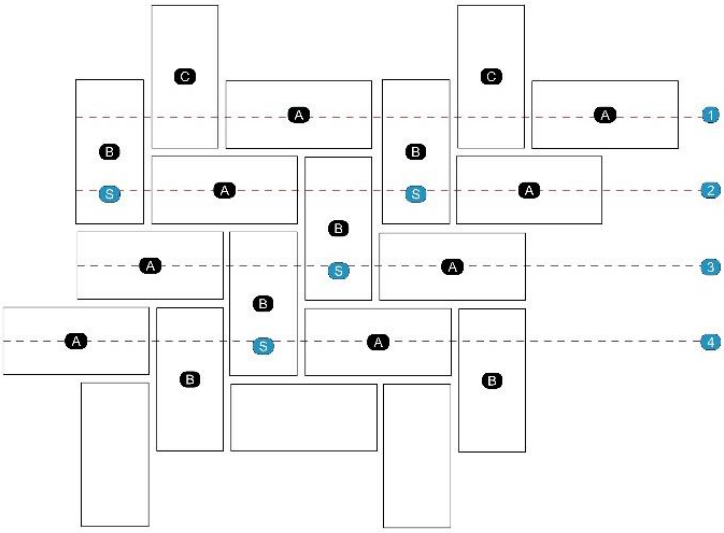

Figure 8: Herringbone pattern analysing with characters

The next sample illustrated in Figure 7 is a barrel vault modelled with a herringbone pattern. To achieve this, parallel lines were created on the vault based on the pattern shape. Various methods can be used to model this pattern, but the approach employed here involves modeling by identifying the repeating pattern in the rows. For this purpose, attention must be paid to four consecutive rows, considering the orientation of the bricks and the distance from the modeling line. With three brick models named A, B, and C, in the first row, the repeating pattern will be BCA. However, in the subsequent rows, it should be noted that since the bricks are shared, the character "S" can be used to create spacing with specific dimensions. Therefore, in the next three rows, the repeating pattern will be SAB, ABS, ABS respectively, taking into account that the starting lines are also different.

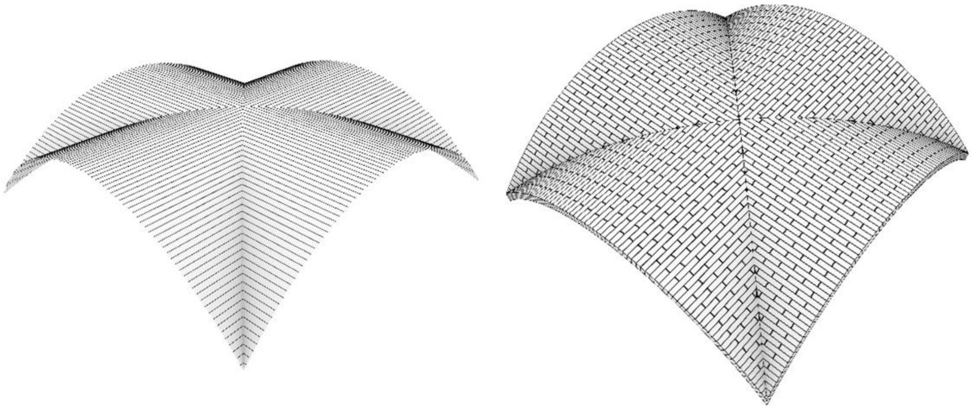

Figure 9: Cross vault modelling

Several studies have delved into the behavior of cross vaults with different brick patterns. According to one illustrated in [24], seismic response, especially collapse mechanisms, is significantly affected by the brick pattern. Typically, radial vaults are prone to global failures, while diagonal vaults exhibit local failures, irrespective of the direction of seismic action. This underscores the pivotal role of patterns in such vaults.

As can be seen in the Figure 9, cross vaults with modulated brick rows along the edge in plan and displacement of each row relative to the previous one have been modeled. Bricks at the edge of each section need to be cut, simplifying the modeling process to focus on one section of the vault. After completing the modeling, it is essential to mirror the edges to obtain the other three sections.

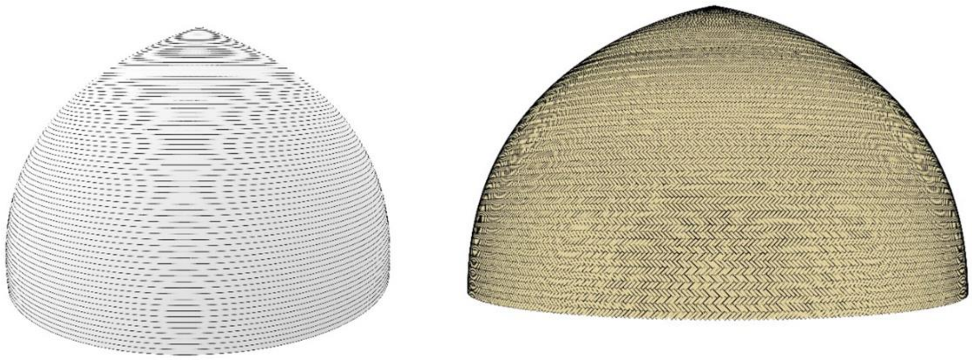

Figure 10: herringbone patterns on a dome shape

When studying brick arrangements, the running bond configuration is extensively researched, with numerous studies exploring its characteristics. However, research on herringbone bond masonry remains somewhat fragmented, despite recent proposals. Although less common for flat walls, herringbone patterns historically find use in curved structures like cross vaults and masonry domes.

One of such vaults is the so-called Arsalan jadheb Tomb in Iran (modelled by the authors in Figure 10), discussed in a paper highlighting the pattern's mechanical properties [25]. Modeling this vault involves arranging bricks along curves in each row, then rotating them, as depicted in Figure 11. Control over mortar spacing is limited, so the character "S" is used to manage pattern spacing. Additionally, as loop lengths decrease in each row, the number of bricks also decreases to maintain pattern consistency.

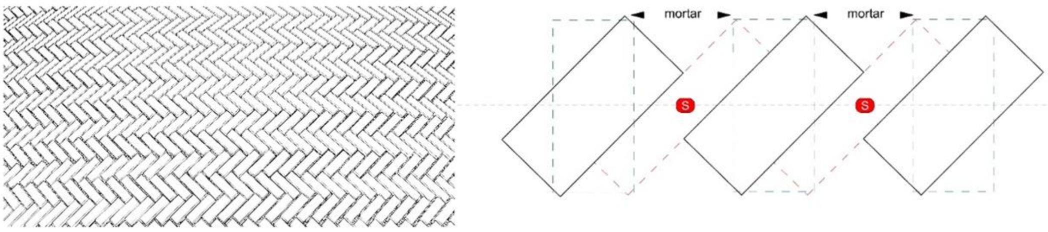

Figure 11: Defining Spaces by “S” character to control distances

Skew arches, which span obstacles at an angle rather than perpendicularly, are prevalent in regions with winding rivers or valleys, such as around delta estuaries. The industrial revolution saw a rapid increase in their construction, as maintaining straight railway tracks became crucial for design. Three primary construction techniques emerged in the 19th century: the false skew arch, the helicoidal method, and the logarithmic method. Previous researches [26] has explored how different patterns affect mechanical properties. In this study, helicoidal models were created by simply defining lines and patterns, as reproduced by Figure 12.

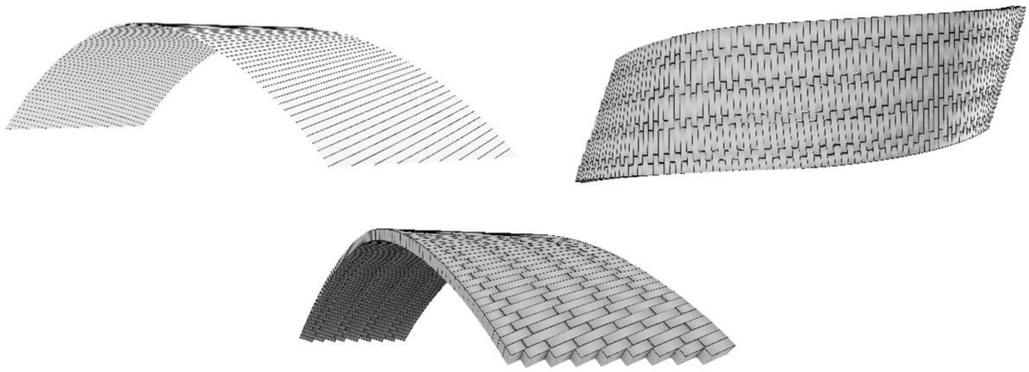
Figure 12: Helicoidal skew arch

4 Conclusions

This paper has introduced the PoliBrick plugin, aimed at simplifying the design process for brick structures arranged in a variety of different patterns. Developed with a keen focus on user accessibility, the plugin is characterized by an intuitive interface featuring just six essential tools, ensuring a streamlined modeling approach that simplifies considerably the heterogeneous discretization of double curvature vaults. The primary objective of this procedure has been to establish an iterative design approach driven by real-time feedback, empowering users to effortlessly create an array of brick arrangements, encompassing designs like basket weave, stretcher bond, herringbone bond, and many other textures. Moreover, it has the potential to integrate post-analysis capabilities for investigating the structural integrity of curved constructions, by interfacing with software of variable complexity, and employing preferred methodologies such as finite or distinct elements.

At present, the PoliBrick plugin is undergoing active development, and it is planned to make ti soon accessible for download via the food4Rhino platform. Given its focus on structural applications, future enhancements may involve deeper integration with finite and distinct element modeling, limit analysis, or other methodologies pertaining to structural engineering. At the current stage of development, users may require only a certain familiarity with Grasshopper software, to effectively utilize the native tools for vault modeling, which can occasionally need a preliminary training. A future priority will be to enhance tool efficiency and intuitiveness.

References

[1] C. Boni, D. Ferretti, E. Lenticchia, “Effects of Brick Pattern on the Static Behavior of Masonry Vaults”, International Journal of Architectural Heritage, 16, 1199-1219, 2022.

[2] F. Roselli, M. Alforno, A. Manuello Bertetto, F. Venuti, “Experimental investigation of the effect of brick pattern on the structural response of masonry arches and barrel vaults”, Construction and Building Materials, 368, 130434, 2023.

[3] J. Heyman, “The safety of masonry arches”, Int. J. Mech. Sci., 43, 209–224, 1969.

[4] M. Alforno, F. Venuti, A. Monaco, “The structural effects of micro-geometry on masonry vaults”, Nexus Netw. J., 22, 1237–1258, 2020.

[5] G.A. Breymann, H. Lang, “Allgemeine Bau-Constructions-Lehre, mit besonderer Beziehung auf das Hochbauwesen”, Constructionen in Stein, Hoffmann, Stuttgart, 1, 1849.

[6] A. Choisy, “L’art de batir chez les Byzantines”, Soci´et´ e anonyme de publications p´eriodiques, Paris, 1883.

[7] D. Wendland, “Traditional vault construction without formwork: masonry pattern and vault shape in the historical technical literature and in experimental studies”, Int. J. Archit. Heritage, 1 (4), 311–365, 2007.

[8] N. Pingaro, G. Milani, “Simple non-linear numerical modelling of masonry arches reinforced with SRG using elasto-fragile and elasto-ductile truss finite elements”, Engineering Structures, 293, 116637, 2023.

[9] A. Gandolfi, N. Pingaro, G. Milani, “Simple Nonlinear Numerical Modeling for Unreinforced and FRP-Reinforced Masonry Domes”, Buildings, 14, 166.

[10] K. Afsari, M. Swarts, T. Gentry., “Integrated generative technique for interactive design of brickworks”, Journal of Information Technology in Construction (ITcon), 19, 225–247, 2014.

[11] B. Tobias, F. Gramazio, M. Kohler., “Digitally fabricating non-standardised brick walls”, In: ManuBuild, Proceedings of the 1st International Conference, Rotterdam, Mark D.Sharp, pp. 191–196, 2007.

[12] L. Lorenz, M. Rippmann, P. Block, “Form finding to fabrication: A digital design process for masonry vaults”, In: Proceedings of the International Association for Shell and Spatial Structures Symposium, 2010.

[13] E. Moussavian, and R. Gentry, “Digital tools for automated generation of vaulted brick assemblies for construction and structural analysis”, In: Proceedings of the 9th International Masonry, 2014.
[14] R. Matthias, L. Lachauer, P. Block, “Interactive Vault Design”, International Journal of Space Structures, 27(4), 219–30, 2012.
[15] S. Rajabzadeh, M. Sassone, “Brick Patterning on Free-Form Surfaces”, Nexus Network Journal, 19, 5–25, 2017.
[16] R. Issa, “Essential Mathematics for Computational Design”, Robert McNeel & Associates, 2010.
[17] N. Roussopoulos, S. Kelley, F.D.R. Vincent, (1995). "Nearest neighbor queries", In: Proceedings of the 1995 ACM SIGMOD international conference on Management of data, p. 71, 1995.
[18] S. Rezaei, M.H. Ronagh, M. Kheirkhah Gildeh, “Effects of Brick Arrangement Patterns on the Mechanical Behavior of Barrel Vaults: Experimental and Numerical Study”, Construction and Building Materials, 134, 106033, 2023.
[19] F. Roselli, M. Alforno, A. Manuello Bertetto, F. Venuti, “Experimental investigation of the effect of brick pattern on the structural response of masonry arches and barrel vaults”, Construction and Building Materials, 368, 130434, 2023.
[20] G.A. Breymann, “Allgemeine Bau-Constructions-Lehre, mit besonderer Beziehung auf das Hochbauwesen ein Leitfaden zu Vorlesungen und zum Selbstunterrichte”, Constructionen in Stein, Stuttgart, Hoffmann, 1, 1849.
[21] D. Donghi, “Manuale dell'Architetto. Compilato sulla traccia del Bakunde des Architekten”, Torino, Utet, 1906.
[22] C. Formenti, “La pratica del fabbricare per l'ingegnere”, Milano, Hoepli, 1893.
[23] C. Levi, “Trattato teorico pratico di costruzioni civili, rurali, stradali ed idrauliche”, Milano, Hoepli, 1932.
[24] M. Alforno, F. Venuti, A. Monaco, C. Calderini, “Seismic behaviour of cross vaults with different brick patterns”, Bulletin of Earthquake Engineering, 20, 3921–3939, 2022.
[25] G. Milani, A. Cecchi, “Compatible model for herringbone bond masonry: Linear elastic homogenization, failure surfaces and structural implementation”, International Journal of Solids and Structures, 50, 3274–3296, 2013.
[26] T. Forgács, V. Sarhosis, K. Bagi, “Influence of construction method on the load bearing capacity of skew masonry arches”, Engineering Structures, 168, 612-627, 2018.