Feature-based topology finding of patterns for shell structures

Robin Oval, Matthias Rippmann, Romain Mesnil, Tom van Mele, Olivier Baverel, Philippe Block

Feature-based Topology Finding of Patterns for Shell Structures

R. Oval[a,b,*], M. Rippmann[b], R. Mesnil[a], T. Van Mele[b], O. Baverel[a], P. Block[b]

[a] *Laboratoire Navier, UMR 8205, École des Ponts, IFSTTAR, CNRS, UPE, Champs-sur-Marne, France*
[b] *ETH Zürich, Institute of Technology in Architecture, Block Research Group, Zürich, Switzerland*

Abstract

This paper introduces *topology finding* of patterns for shell structures such as beam grids for gridshells or voussoir tessellations for vaults, among others. The authors refer to topology finding, by analogy and in complement to form finding, as the design of the connectivity of these patterns in order to follow architectural, structural and construction requirements. This paper presents a computational approach relying on a specific design space and data structures based on singularity meshes, which encode the information about the singularities in patterns. The designed patterns are structured, i.e. with a low number of singularities, can include high-valency pole points, and respect alignment to surfaces, curves and points. A feature-based exploration approach is introduced with a generation procedure for singularity meshes following the boundaries of a surface as well as point and curve features, using a topological skeleton or medial axis. These features can stem from statics heuristics, whose efficiency is assessed in a case study. A rule-based editing approach for singularity meshes supplements feature-based topology finding, using a grammar of strip rules as parameters to further explore the singularity design space. This conceptual design approach and its algorithms are an aid for topological exploration of patterns for shell-like structures by architects and engineers.

Keywords: conceptual design, structural design, shell structures, patterns, meshes, topology, singularity, topological skeleton, grammar.

1. Introduction

1.1. Context

Shell structures span efficiently large areas thanks to their double curvature that provides geometrical stiffness. These structures are discretised in a pattern, which integrates the load-bearing and the cladding systems to be fabricated and assembled. Beam networks for gridshells, voussoir tessellations for vaults, cable layouts for cable nets and beam networks for nexorades are such examples of patterns for shells, as shown in Figure 1. The topology and geometry of these patterns influence and are influenced by many project aspects such as aesthetics, statics, fabrication, assembly, as well as sustainability and cost.

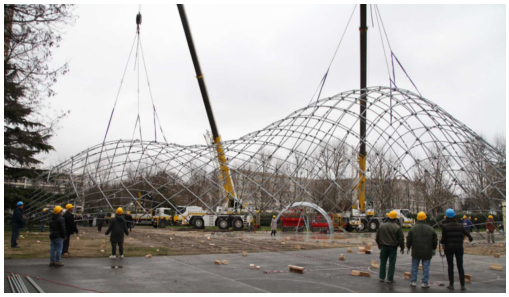
*Corresponding author
Email address: robin.oval@enpc.fr (R. Oval)

1.2. Designing patterns

Design strategies tackle the topology and geometry of patterns for shell-like structures in different manners.

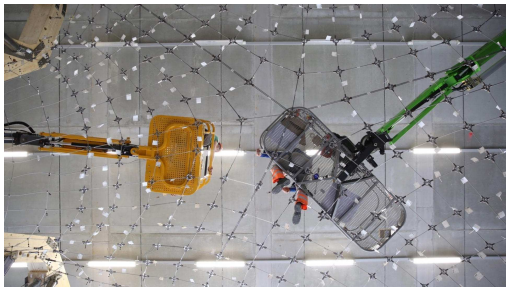
1.2.1. Form finding

Form finding explores the geometry of a pattern to achieve diverse criteria, using strategies such as force-based methods (Force Density Method [5], Dynamic Relaxation [6], Thrust Network Analysis [7], Update Reference Strategy [8], etc.) and fabrication-based methods (Scale-Trans Surfaces [9], Marionette Meshes [10], etc.). However, the pattern has a predefined topology, which constrains the form-found geometries to the same geometrical design space, which heavily depends on the experience of the designer with regard to the choice of topology.

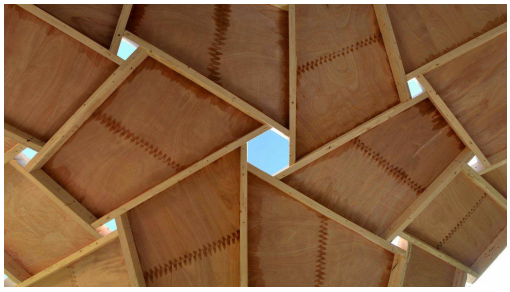


(a) Beam grid of the Ephemeral Cathedral in Créteil, France [1] (Photo credits: thinkshell.fr)

(b) Voussoir tessellation of the Armadillo Vault in Venice, Italy [2] (Photo credits: Iwan Baan)



(c) Cable layout of the prototype for the NEST HiLo roof in Dübendorf, Switzerland [3] (Photo credits: Naida Iljazovic)



(d) Beam network of a shell-nexorade hybrid at École des Ponts, Champs-sur-Marne, France [4] (Photo credits: Romain Mesnil)

Figure 1: Examples of patterns for shell-like structures.

1.2.2. Form optimisation

Form optimisation of the shape of the shell can be performed on the coordinates of the vertices of a surface or a mesh with a predefined topology [11, 12].

1.2.3. Field integration

Some design strategies generate the topology of the pattern as well. More specifically, vector- or cross-field integration methods generate the geometry as well as the topology of the pattern, whose singularities correspond to the ones of the field. These integration procedures can be applied to the principal stress directions for mechanical efficiency [13, 14] or to the principal curvature directions for fabrication properties [15].

1.2.4. Topology optimisation

Topology optimisation generates both the geometry and the topology of the pattern [16]. The resulting designs are highly optimised regarding mechanics but not necessarily feasible regarding construction considerations, as discussed by Borgart [17].

Although the last approaches do not constitute exploration strategies for the designer, such methods can be used as a collaboration means between architects and engineers for integrated design [18].

1.3. Research statement

In practice, architects and engineers resort to heuristics to draw a topology for a pattern, in a tedious project-specific procedure without automation process [19]. Although common practice in other industries like computer graphics, topological mesh modelling is not well spread in architecture, engineering and construction, and existing methods are not designed for this specific domain. Yet, the topology of a pattern matters because it sets the bounds of the available geometrical design space, within the more general design space. This geometrical space, which represents all the possible geometries for a given topology, may not contain efficient or even feasible designs. For this reason, designers need conceptual and practical tools for *topology finding* to deepen the design space, to allow them

to efficiently explore the topology of patterns for structural design at the early stages of the project, as already investigated for other architectural and structural concepts [20, 21, 22]. Topology finding empowers the existing geometrical design and optimisation algorithms to achieve efficient structures.

1.4. Contributions

This paper introduces topology finding of patterns for shell-like structures. A focus is set on the singularities in structured quad-based patterns, via the exploration of singularity meshes, which encode the information about the singularities in the pattern, including high-valency pole points. The presented algorithms allow to generate patterns that are structured, i.e. with a small number of singularities, justified by the implications of designing unstructured patterns on aesthetics, statics and construction [23]. Furthermore, they are aligned with features like surface boundaries, points and curves, which can stem from different aspects, like a column or a fold to integrate in the pattern in order to follow statics-aware heuristics.

Section 2 defines the approach with the design space and data structures for the exploration of singularity meshes. Section 3 develops feature-based topology finding with a skeleton-based generation procedure for singularity meshes via the shape's topological skeleton. The integration of point and curve features is validated as statics-aware heuristics in a case study. Section 4 presents a rule-based editing strategy for singularity meshes using a strip grammar to explore different topologies that still include the desired features.

This research is implemented in compas_pattern [24] as a Python package of COMPAS [25], an open-source Python-based computational framework for collaboration and research in architecture, engineering and digital fabrication.

2. Approach

This section shows how the design spaces and the related data are structured to tackle the design of patterns with a focus on their singularities.

2.1. Design space structure

The design spaces related to patterns can be structured as in Figure 2, with the ones related to topology (the singularities, the density and the general connectivity of the pattern) upstream from the one related to geometry.

1. The *pattern singularity space* encodes the data related to the singularities of the pattern with singularity meshes in the form of coarse pseudo-quad meshes (see Sections 2.2 and 2.3.1);

2. The *pattern density space* encodes the data related to the density of the pattern in the form of quad meshes, with optional pseudo-quads for poles. It is based on the selected singularity mesh and setting the density parameters per quad face strip (see Section 2.3.2);

3. The *pattern connectivity space* encodes the data related to the general connectivity of the pattern in the form of general meshes. It is based on the selected density mesh and defining global and/or local topological modifications (see Section 2.3.3);

4. The *pattern geometry space* encodes the data related to the geometry of the pattern in the form of general meshes as well. It is based on the selected connectivity mesh and modifying the coordinates of the vertices (see Section 2.3.4).

With this hierarchy, the singularity design pace is the most upstream, therefore, the most influential one, as it sets the limits of the downstream design spaces.
The pattern data structure needed to explore these design spaces is detailed in the next section.

2.2. Data structure

The data structure used to explore the topology of a pattern relies on the singularity mesh as a coarse pseudo-quad mesh, a specific type of mesh, as shown in Figure 3:

1. Meshes can be defined with a list of vertices as point coordinates for geometry, and a list of faces as lists of vertex keys for topology, from which a more efficient half-edge mesh data structure can be computed [26], as implemented in COMPAS;

2. Quad meshes are meshes in which all faces are quads, as lists of exactly four vertex keys,

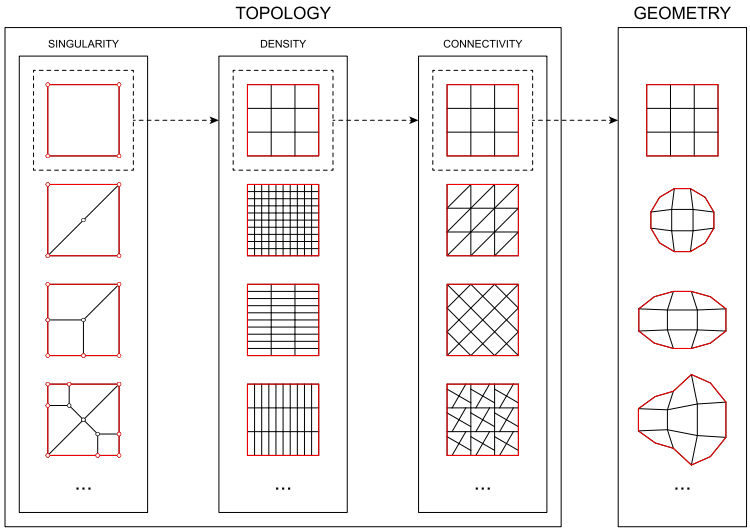


Figure 2: Design space structure of a pattern's singularities, density, connectivity and geometry, where each design space is defined by the design choices in the upstream spaces.

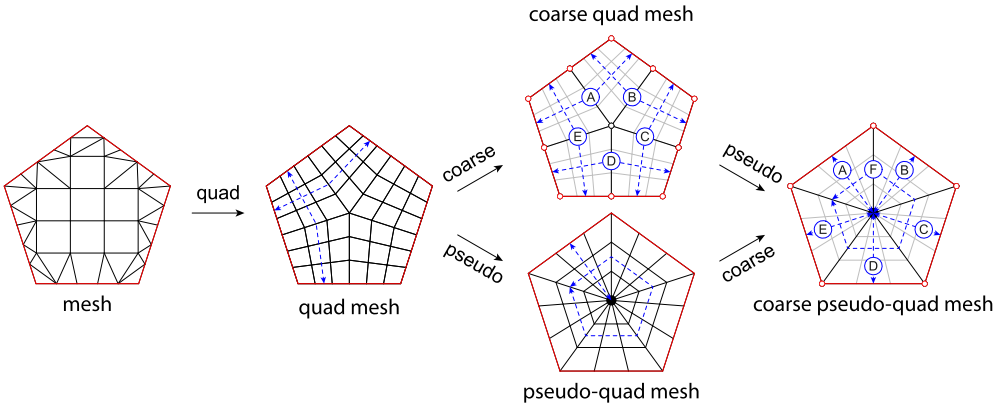


Figure 3: From the data structure of general meshes to the one of coarse pseudo-quad meshes for singularity meshes. Meshes are represented in black and their densified meshes in grey, boundaries in red, strips in blue, density parameters in capital letters and poles as filled dots.

which allows definition of quad face strips as lists of edges that are facing each other across the quad faces;

3. Coarse quad meshes are quad meshes with a density parameter defined for each strip for densification into a child quad mesh, whose vertex, face and edge elements inherit the attributes of the parent elements, such as to which strip in the coarse quad mesh corresponds a poly-edge in the quad mesh;

4. Pseudo-quad meshes are quad meshes with some pseudo-quad faces that are geometrically like triangles but topologically like quads, with a double vertex at the location of the pole encoded as a list of face vertices in the type [a, b, c, c] instead of [a, b, c, d];

5. Coarse pseudo-quad meshes are the combination of coarse quad meshes and pseudo-quad meshes.

The coarse pseudo-quad mesh of the singularity mesh defines the relationship between the singularities and the poles of the pattern. The strip density parameters define the pseudo-quad mesh of the density mesh. Applying topological modifications result in the general mesh of the connectivity mesh, whose vertices can be moved to explore the

geometry of the pattern.

This approach relates to mesh modelling environments with their low-poly meshes and subdivision algorithms, which got into structural design for their lightness compared to directly modelling a dense mesh [27, 28, 29].

The next section provides specific details for the exploration of the different design spaces.

2.3. Design spaces

Different parameters and constraints control the exploration of the different design spaces.

2.3.1. Singularity design space

Approaches to explore the singularity design space are presented in Sections 3 and 4. The specific complexity of the singularity space is discussed.

The surfaces representing the shell-like structures in Figure 1 are classified as compact two-manifolds [30], which permits to topologically characterise their mesh representations by their Euler's characteristic X, computed as:

$$X = V - E + F, \tag{1}$$

where V is the number of vertices, E the number of edges and F the number of faces of the mesh. The Euler's characteristic of such surfaces is actually independent from its mesh representation and can be directly computed as:

$$X = 2 - 2g - N, \tag{2}$$

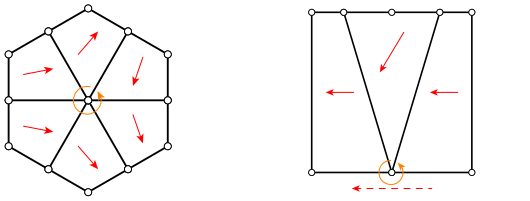
where g is the number of handles – or genus – and N the number of boundaries, in the case of an orientable surface. The Euler's characteristic sets a constraint on the choice of singularities in the mesh via the Poincaré-Hopf theorem:

$$X = \sum_{v \in V} i_v, \tag{3}$$

where V is the set of all the vertices and i_v the index of a vertex v. The index of a singularity relates to the deviation it induces in the orientation of the faces:

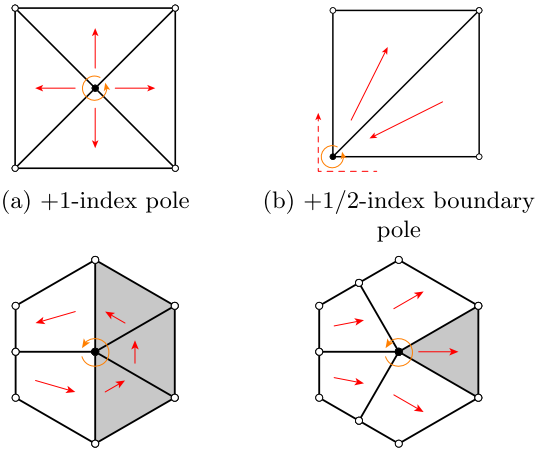
$$i_v = \frac{1}{2\pi} \sum_{f_v} d\theta, \tag{4}$$

where $d\theta$ is the signed anticlockwise angular deviation of the quad faces during an anticlockwise loop around the vertex v, as illustrated in Figure 4 for two negative singularities.



(a) $-1/2$-index singularity (b) $-1/4$-index boundary singularity

Figure 4: Computing vertex indices in a quad mesh with the direction deviation (in red) during a loop (in orange)around the vertex.



(a) +1-index pole (b) +1/2-index boundary pole

(c) 0-index partial pole (d) +1/2-index partial pole

Figure 5: Computing indices of poles and partial poles in pseudo-quad meshes.

These vertex indices can be directly computed in quad meshes from the valency n_v:

$$i_v = \frac{n_0 - n_v}{4}, \tag{5}$$

where n_0 is the regular valency, equal to 4 and 3 for a non-boundary and a boundary vertex, respectively.

In pseudo-quad meshes, the index of non-boundary and boundary poles are independent of their valency and are equal to 1 and 1/2, respectively, as shown in Figures 5a and 5b. The index of a partial pole in a quad mesh, which is adjacent to both pseudo-quads and quads, is computed as a non-pole vertex using Equation 5 after collapsing

the pseudo-quads, as in Figures 5c and 5d.

While applying topological modifications to a mesh, the Poincaré-Hopf theorem still applies and can be differentiated as:

$$0 = \sum_{v \in V_+} i_v - \sum_{v \in V_-} i_v + \sum_{v \in V^*} \Delta i_v, \qquad (6)$$

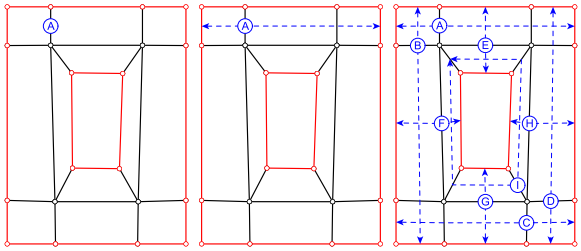
where V_+ is the set of added vertices, V_- the set of deleted vertices, V^* the set of modified vertices and Δi_v is the variation of the index.

The singularity design space of coarse pseudo-quad meshes is rich, even restricted to a shape with a given Euler's characteristic, but complex: topological modifications applied to the singularities have to be considered globally because of the relation between them, on the contrary to local modifications of a vertex coordinates. Therefore, this design space necessitates specific exploration approaches.
Nevertheless, thinking in terms of singularity indices directly informs the designer what topological modifications are possible according to the differential Poincaré-Hopf theorem in Equation 6. For instance, a 6-valent singularity of index $-1/2$ and a 3-valent singularity of index $+1/4$ can be merged in a 5-valent singularity of index $-1/4$, because the sum of the indices is preserved.

2.3.2. Density design space

The strips of quad faces are the key structure in quad meshes, as used in some modelling approaches [31, 32]. As illustrated in Figure 6, the corresponding strip data is collected as lists of edges as follows:

1. start with the complete list of quad mesh edges;
2. pop one edge from the list (Figure 6a);
3. collect the edges across the adjacent quad faces and repeat in each direction until the boundaries are met in the case of open strips (Figure 6b), or until it forms a loop in the case of closed strips;
4. mark the collected edges as one strip and remove them from the list of edges;
5. repeat from step 2 until the initial list of edges is empty (Figure 6c).



(a) (b) (c)

Figure 6: Collecting the strip data as lists of edges across the quad mesh faces. The boundaries are highlighted in red and the strips are represented as dashed blue lines.

Each strip corresponds to one independent density parameter in the singularity mesh, which is the subdivision of the strip edges for the density mesh. The strip density parameters form the densification degrees of freedom, are strictly positive integers and can all be different, opposed to a Catmull-Clark subdivision procedure with a unique global density parameter [33]. The density parameters d_i can be chosen by the user via the length of the strip edges, aided by automated computation based on a target length l_0 as:

$$d_i = \lceil f(l_i)/l_0 \rceil \,, \qquad (7)$$

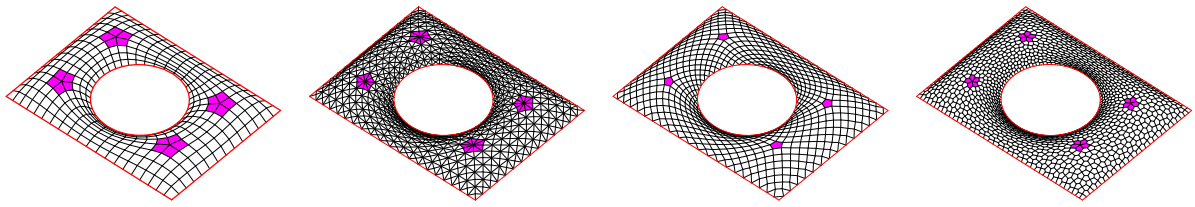
where f is a function applied to the lengths of the edges of the strip i, like the average, the minimum or the maximum. In practice, the average is used for all the examples in this paper.

Once each strip density parameter is set, each quad face in the coarse quad mesh is meshed. The geometry follows the hyperbolic paraboloid $\mathbf{S}$ that linearly interpolates the four face vertices:

$$\begin{aligned} &\forall (u,v) \in [0,1]^2, \\ &\mathbf{S}(u,v) = (1-u)\cdot(1-v)\cdot\mathbf{A} \\ &+u\cdot(1-v)\cdot\mathbf{B} + (1-u)\cdot v\cdot\mathbf{D} + u\cdot v\cdot\mathbf{C}, \end{aligned} \qquad (8)$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are the four ordered vertex coordinates of the face. The surface is discretised in a mesh with the U and V density parameters corresponding to the strips which include the facing edges $(A-B)$ and $(D-C)$, and the facing edges $(B-C)$ and $(A-D)$, respectively.

These meshes are then joined and their boundary vertices welded together to form the quad mesh of the density mesh.



(a) Seed (b) Kis (c) Ambo (d) Gyro

Figure 7: Exploring quad-based pattern symmetries with equivalent vertex or face singularities highlighted in pink by applying different Conway operators on a seed quad mesh.

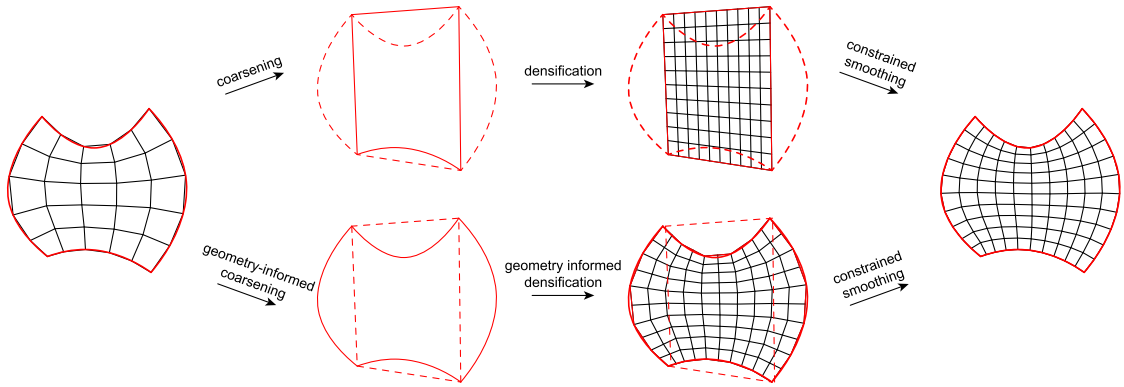


Figure 8: Taking into account geometry during densification for lighter constrained smoothing. The mesh are represented in black and the target boundaries in red.

2.3.3. Connectivity design space

Based on the density quad mesh, any modification can be applied to form the actual connectivity of the pattern.
Connectivity editing can include local edge operations, like add, delete, swap, split, or trimming to fit a landscape, when boundary alignment is not necessary [34, 35], for instance.
More specifically, one or multiple Conway operators [36] can be used to apply global modifications and allow exploration of different pattern symmetries with equivalent singularities, structuredness and feature-alignment, while not being constrained to quad patterns. Conway operators have already been explored in structural design for the optimisation of space frame structures [37, 38].
As illustrated in Figure 7, the seed quad mesh can yield a doubly-triangulated pattern with the Conway kis operator, a dual diagonal pattern with the Conway ambo operator or a pentagonal pattern with the Conway gyro operator, for instance, with equivalent vertex or face singularities, highlighted in pink.

2.3.4. Geometry design space

During exploration of the pattern geometry, any geometrical processing can be performed, related to form finding or form optimisation, based on the vertex coordinates as parameters. More specifically, smoothing or relaxation algorithms can be used to regularise the geometry after topological processing.

After setting the actual connectivity of a pattern during a geometry-blind process, relaxation can improve its geometrical quality using Laplacian smoothing [26, 39], before further processing. For a given number of iterations or until convergence below a given threshold value, each vertex of the pattern is moved towards the centroid of its adjacent vertices at:

$$\mathbf{V_f} = \mathbf{V_i} + (1 - d) \cdot (\bar{\mathbf{V}}_\mathbf{i} - \mathbf{V_i}), \tag{9}$$

where $\mathbf{V_f}$ is the final position of the vertex, $\mathbf{V_i}$ its initial position, $\bar{\mathbf{V}}_\mathbf{i}$ the centroid of its adjacent vertices with optional weights, and d a user-defined damping value between 0 and 1 for stability, classically set to 0.5. The values per

vertex are computed at each iteration and the vertex coordinates updated simultaneously at the end of the iteration to avoid the bias of starting from one random vertex.
Additionally, constraints are set on the vertices to project them back on surfaces, curves and points after each iteration, to fit a target shape and optional point and curve features. These constraints are stored at the level of the singularity mesh parent elements and then inherited by the child vertices of the other meshes.
This switch from topological to geometrical spaces can be a real challenge, as a highly distorted mesh with overlapping or collapsed elements is harder to geometrically process. Nevertheless, early geometrical regularisation with Laplacian smoothing of the singularity mesh can prevent these problems later.
To reduce additional smoothing computation induced by density modification of the singularity mesh faces, re-densification can be performed following the existing geometry to almost fit the smoothing constraints, as illustrated in Figure 8.

Based on the presented design space and data structures for patterns, the focus is set on the exploration of the singularity design space.
The next section presents feature-based topology finding using a skeleton-based generation procedure of a singularity mesh for an input surface, based on its topological skeleton, with optional point and curve features, which can stem from statics-aware heuristics.

3. Feature-based exploration

A topology-finding algorithm for feature-based exploration is introduced. The singularity mesh is generated based on a topological skeleton of the relevant features such as a surface's boundaries, as well as points and curves on the surface. These features can be modified to generate and explore different topologies. The main steps of the procedure are presented in Figure 9 to generate a pattern from the mentioned features. Starting with an input straight or curved surface, with optional point and curve features (Figure 9a), its topological skeleton or medial axis (Figure 9b), introduced by Blum [40], is generated and modified to yield a singularity mesh that includes the singular points of the medial axis (Figure 9c), which are also featured in the corresponding pattern (Figure 9d).

3.1. Core feature: surface

The input surface, here the geometry of the courtyard roof of the British Museum in London, England, analytically defined by Williams [41], is mapped to the plan following its UV-parameterisation. Other surface or mesh mapping strategies can change the planar map and induce different results. If the boundaries of the surface are seen as the core feature, the mapping strategy should not distort their geometry too much to faithfully integrate them in the topology.

3.1.1. Algorithm

As shown in Figure 10, the planar map is procedurally decomposed in four sided patches based on the medial axis, which consists of a set of curves called medial branches that serve as dimensional reduction of the shape, which are connected together at singular points.

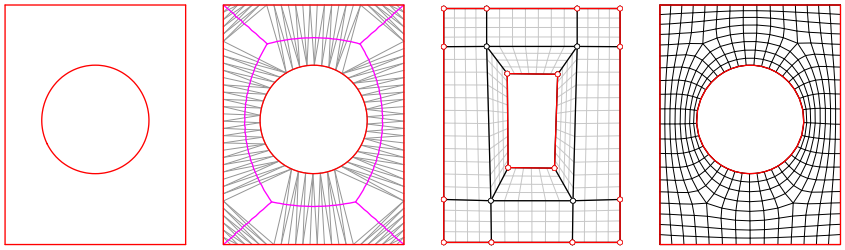
The boundaries of the planar map are subdivided into a set of points that serve as vertices for Delaunay triangulation with deletion of the faces lying outside the boundaries, as shown in Figure 10a. The discretisation parameter of each curve d_i should be tailored to capture the relevant curvature changes without inducing unnecessary heavy computation. A d_{scale} value as a percentage of the scale, the scale being the length of the bounding box diagonal D of the planar map, and a d_{min} value as a minimum number of subdivision are combined:

$$d_i = \min(d_{scale}, d_{min}), \tag{10}$$

with values of 5 for d_{min} and 1% to 5% of D for d_{scale} yielding good results in practice.

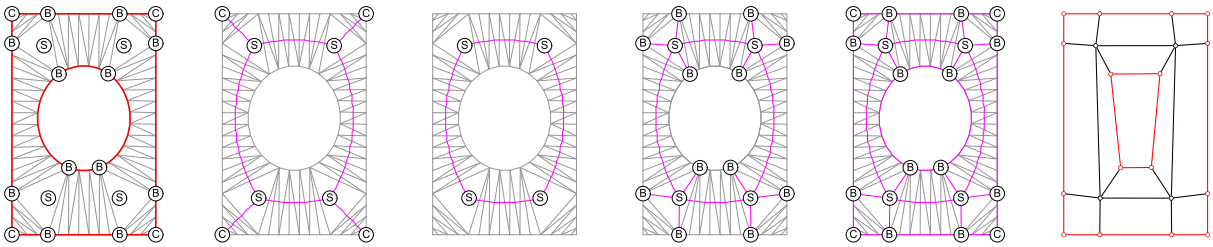
Three types of faces must be distinguished depending on the number of adjacent faces: faces with two neighbours are regular faces, with three neighbours singular faces, and with one neighbour corner faces.
Additionally, three types of points must be distinguished: singular points S are the circumcentres of the singular faces, boundary points B are the vertices of the singular faces, and corner points C are the two-valent boundary vertices.

The medial axis is constituted by the branches connecting the circumcentres of the adjacent Delaunay



(a) Input (b) Medial axis (c) Singularity mesh (d) Pattern

Figure 9: Skeleton-based generation of a singularity mesh and a pattern on an input shape based on the singularities of its medial axis. The input surface is marked in red, the Delaunay triangulation in dark grey, the topological skeleton in pink, the singularity mesh in black, the density mesh in light grey and the pattern in black.



(a) Key points (b) Medial axis (c) Pruning (d) Grafting (e) Closing (f) Coarse mesh

Figure 10: Skeleton-based generation procedure of a singularity mesh based on the medial axis. The input surface is marked in red, the Delaunay triangulation in grey with the key points labelled in black, the modified topological skeleton in pink and the singularity mesh in black.

faces, as shown in Figure 10b. The medial axis results in the S-S branches, connecting S points, and the S-C branches, connecting S points and C points.

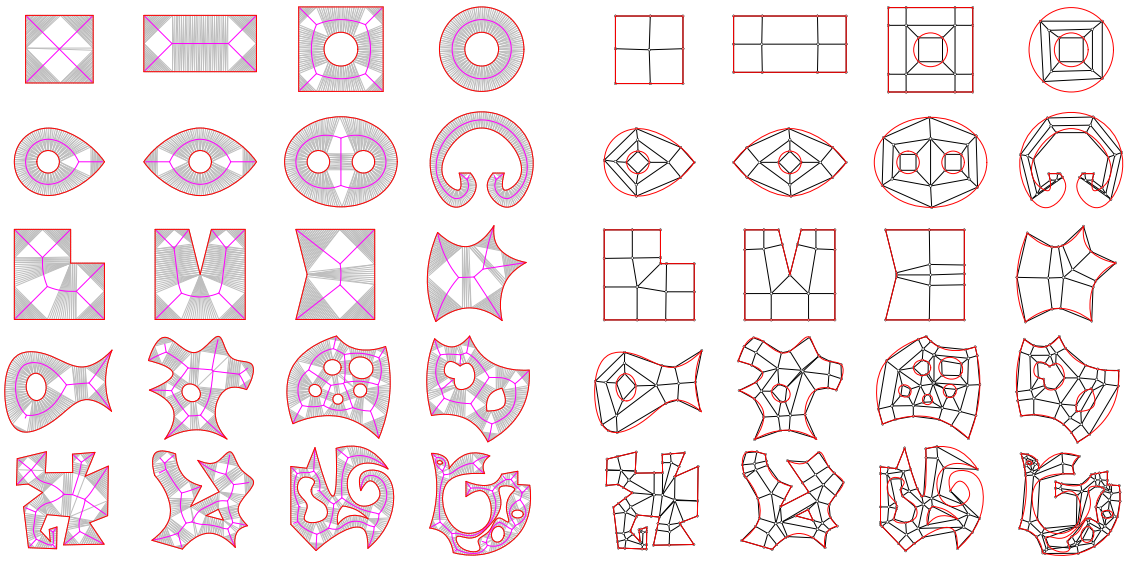
The medial axis is then modified to achieve a set of four-sided patches. The topological operations applied here are similar to the ones in the work of Rigby [42], to yield coarser quad meshes than other skeleton-based block decomposition approaches [43]. These topological operations only depend on the connectivity of the Delaunay mesh: pruning to remove the S-C branches (Figure 10c), grafting to add the S-B branches (Figure 10d), closing to add the B-B and B-C branches (Figure 10e). This decomposition yields four-sided patches which are all composed of one S-S element, two S-B elements and one B-B element, except for the corner patches which are composed of two S-B elements and two B-C elements.

The adjacency of the patches is extracted to define the coarse quad mesh of the singularity mesh, as shown in Figure 10f, and in Figure 11 on benchmark examples. The planar singularity mesh is finally mapped back onto the input, planar or curved, surface.

Further processing to form a pattern can include densification based on a target length combined with constrained relaxation on the input shape, as described in Sections 2.3.2 and 2.3.4, respectively. Another mesh modelling approach involves geometrical exploration of the singularity mesh combined with a Catmull-Clark subdivision procedure, as illustrated in Figure 12, after skeleton-based generation of a singularity mesh for a planar surface with multiple openings.

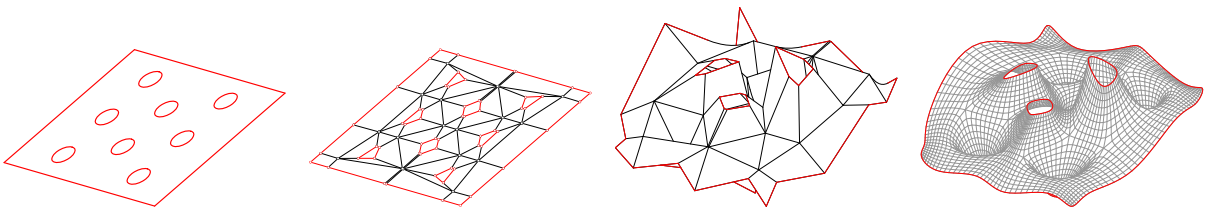
3.1.2. Corrections

Some additional corrections are needed on the set of patches during the decomposition algorithm to ensure the validity and the quality of the singularity mesh in capturing features, even though these corrections reduce the coarseness of the resulting singularity meshes, as some elements



(a) Topological skeletons (b) Singularity meshes

Figure 11: Benchmark surfaces for skeleton-based generation of singularity meshes. Boundaries in red, Delaunay mesh in grey, skeleton in pink and singularity mesh in black.
Parameters: $d_{scale} = 0.02D$, $d_{min} = 10$, $\theta_{kink} = \pi/8$, $\theta_{subd} = \pi/2$.



(a) Input surface (b) Singularity mesh (c) Geometrical exploration (d) Pattern

Figure 12: Mesh modelling starting with a planar surface for skeleton-based generation of a singularity mesh, which can be geometrically modified to form a curved pattern using a Catmull-Clark subdivision procedure. Boundaries in red, singularity mesh in black and mesh in grey.

could be removed without loosing any data about the singularities, but loosing features, like openings or kinks.
In the following descriptive figures, the meshes are represented by continuous lines and the surface boundaries by dashed curves.

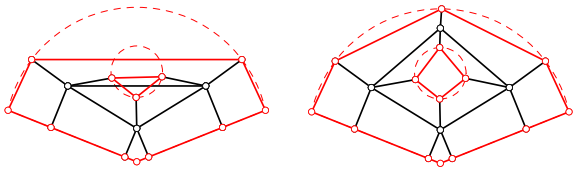
Some straight faces differ a lot and can be even flipped compared to their curved patch, resulting in strong distortions or overlaps of the elements in the singularity mesh, thus a loss of readability in spite of topological validity, as shown in Figure 13a. If so, such patches are subdivided, as shown in Figure 13b. The number of subdivisions is computed based on the total rotation of a medial branch between two S points as:

$$n = \left\lfloor \frac{\left|\sum_i \theta_i\right|}{\theta_{subd}} \right\rfloor + 1, \tag{11}$$

where θ_i for the vertex V_i is the angle between the adjacent edges $E_{i-1,i}$ and $E_{i,i+1}$ and θ_{subd} is the critical angle value for which one subdivision must occur. A recommended value θ_{subd} of $\pi/2$ yields good results, as used in the benchmark examples in Figure 11.

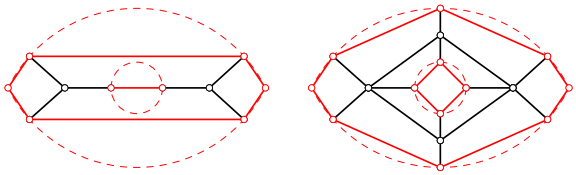
A subdivision criterion θ_{subd} equal or smaller than $\pi/2$ also avoids boundary collapses. Otherwise, some boundaries could be subdivided by two vertices or less and disappear in the singularity mesh, as show in Figure 14a, and corrected in

Figure 14b.



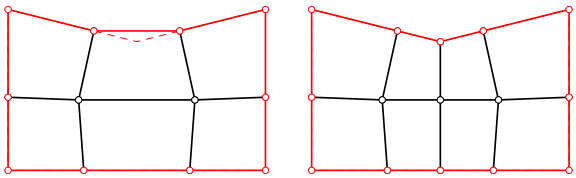
(a) Before correction (b) After correction

Figure 13: Correcting distorted faces by subdividing patches. Mesh in black with boundaries in red and input surface as dashed red curves.



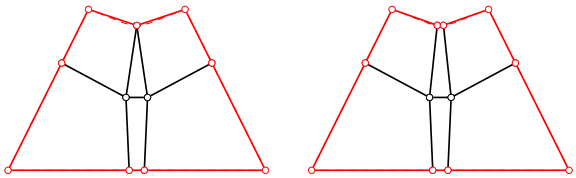
(a) Before correction (b) After correction

Figure 14: Correcting collapsed boundaries by subdividing patches. Mesh in black with boundaries in red and input surface as dashed red curves.



(a) Before correction (b) After correction

Figure 15: Correcting missed concavities by subdividing patches. Mesh in black with boundaries in red and input surface as dashed red curves.



(a) Before correction (b) After correction

Figure 16: Correcting triangular faces by inserting a zero-length fourth edge. Mesh in black with boundaries in red and input surface as dashed red curves.

Some boundary concavities – or inward kinks – are not marked by the medial axis, and are lost in the singularity mesh, as shown in Figure 15a. If so, the patches are subdivided, as shown in Figure 15b. The concavities are detected among the boundary vertices of the Delaunay mesh that are not two-valent corner points C by comparing the local angle with the average of the adjacent angles:

$$\theta_i - \frac{\theta_{i-1} + \theta_{i+1}}{2} \geq \theta_{kink}, \tag{12}$$

where θ_i at vertex V_i is the angle between the adjacent edges $E_{i-1,i}$ and $E_{i,i+1}$ and θ_{kink} is the critical angle value set to define the kink. A recommended value θ_{kink} of $\pi/8$ yields good results, as used in the benchmark surfaces in Figure 11.

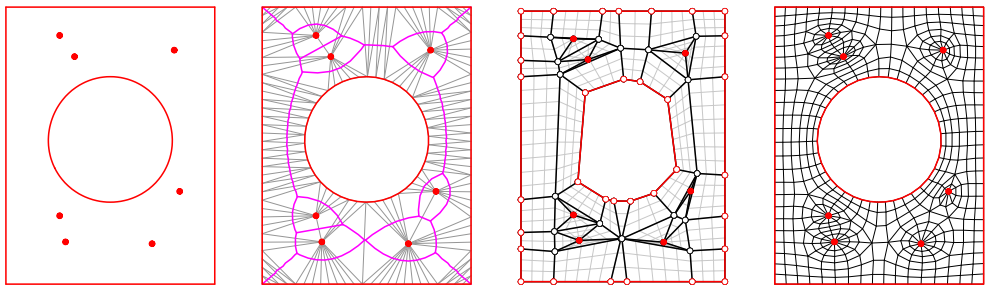
The previous corrections are applied at the level of the curved patches, as they relate to the geometry of the medial branches. The following corrections are applied at the level of the straight faces, as they relate to the topology. Some faces can be triangles in the singularity mesh, resulting either from two S points or two B points at the same location, as shown in Figure 16a. If so, a zero-length fourth edge is inserted at the location of the superimposed points to form a topological quad, as shown in Figure 16b.

3.2. Additional features

Additionally to the surface, important features represented as points or curves on the surface can be integrated in the feature-based topology finding by taking them into account in the medial-axis generation. These features can generate pole points to support statics considerations.

3.2.1. Pole points

As featured in the isostatic ribbed floors of Pier Luigi Nervi for the Gatti Wool factory in Rome, Italy [44], and of Hans-Dieter Hecker for the lecture hall of the zoological department of the University of Freiburg, Germany [45], principal stresses converge towards columns and walls, often featured by poles in the pattern, a specific type of singularities with a high valency that increases with the density. Therefore, poles attract forces but are harder to materialise. The designer has to choose whether to resort to them or not, and adjust their valency, as illustrated by the courtyard roofs of the Dutch Maritime Museum [46] in Amsterdam, the Netherlands, and of the British Museum [41], which share similar support conditions allowing thrust only at their corners: the former features poles, the latter



(a) Input (b) Medial axis (c) Singularity mesh (d) Pattern

Figure 17: Skeleton-based generation of a pattern integrating point features and poles represented as filled dots. The input is marked in red, the Delaunay triangulation in dark grey, the topological skeleton in pink, the singularity mesh in black, the density mesh in light grey and the pattern in black.

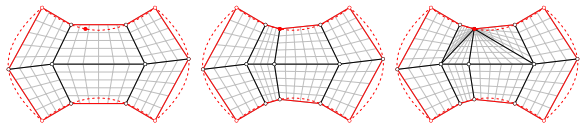
does not. However, accommodating boundary and support conditions as for the Rhön-Klinikum cable net in Bad Neustadt, Germany with many mast supports [47] to include as pole points in order to locally have radial patterns in the cable net is not straightforward when designing a pattern without procedure.
Moreover, poles in force patterns such as thrust networks [7] can provide an appropriately high number of loads paths at the location of concentrated loads and improve the results of funicular form fitting of target shapes [48]. Furthermore, load paths in thrust networks should also be aligned to curve features stemming from geometrical discontinuities like folds [49].

Therefore, additional features represented by points and curves are to be integrated in the design of patterns, to be able to follow such heuristics. These features can stem from discontinuities related to statics (point/line loads/supports), as well as geometry (peaks or folds). A case study validates the relevance of these statics-aware heuristics in Section 3.2.4. Additionally, these features can be used to influence the topology of the pattern by exploring different singularity meshes.

3.2.2. Point features

Point features are included in the input for skeleton-based generation as points on the surface, as shown in Figure 17a. These points are added to the set of vertices for the Delaunay triangulation for generation of the medial axis, which includes new singular areas adjacent to the point features, as shown in Figure 17b. Following the same algorithm, the resulting singularity mesh yields pseudo-quad faces around the point features, as shown in Figure 17c, resulting in a pattern with poles, as shown in Figure 17d.

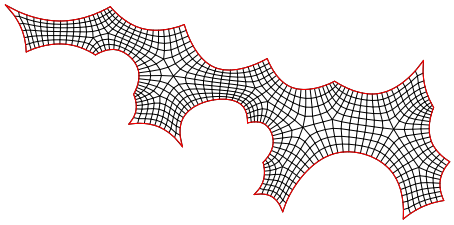
Boundary point features do not directly modify the singularity mesh, since boundary points are already part of the Delaunay triangulation, as in Figure 18a, and additional steps must be included. First, edges are added if the boundary point feature is not marked by a vertex in the singularity mesh, as shown in Figure 18b. Second, all the quad faces adajcent to the point feature are split into two pseudo-quads [50] to create the boundary pole, as shown in Figure 18c.



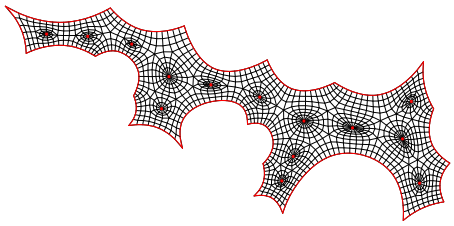
(a) Initial generation (b) Feature integration (c) Pole creation

Figure 18: Adding poles at point features on the boundary, marked as a filled dot. Input surface as dashed red curves, singularity mesh in black and density mesh in grey.

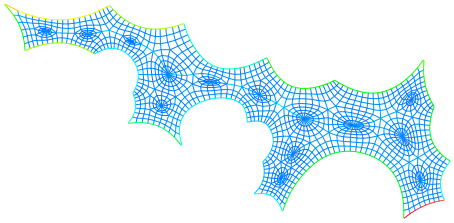
Revisiting the pattern of the Rhön Klinikum cable net with the presented method yields the planar pattern without poles in Figure 19a, characterised by structuredness and boundary alignment. Mast supports are integrated as point features to yield the planar pattern with poles in Figure 19b. The pattern is converted into a form diagram to perform funicular form finding of cable nets with an adapted Thrust Network Analysis [51], using RhinoVAULT [52]. The form diagram in Figure 19c and the force diagram in Figure 19d present clear visual identification of their reciprocal features, such as the hoop



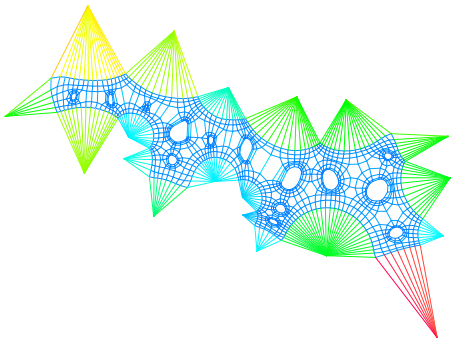
(a) Pattern without poles



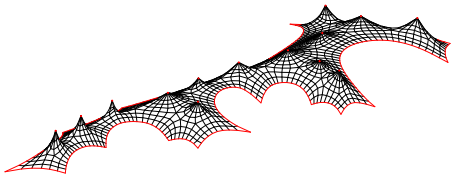
(b) Pattern with poles



(c) Form diagram



(d) Force diagram (rotated by 90°)



(e) Form-found cable net

Figure 19: Revisiting the topology of the pattern of the Rhön Klinikum cable net in Bad Neustadt, Germany [47] by including poles through point features at the location of the mast supports.

cables and the boundary cables, a key aspect for this graphical method to explore forms via different force equilibria. The form-found cable net corresponding to these two diagrams is shown in Figure 19e.

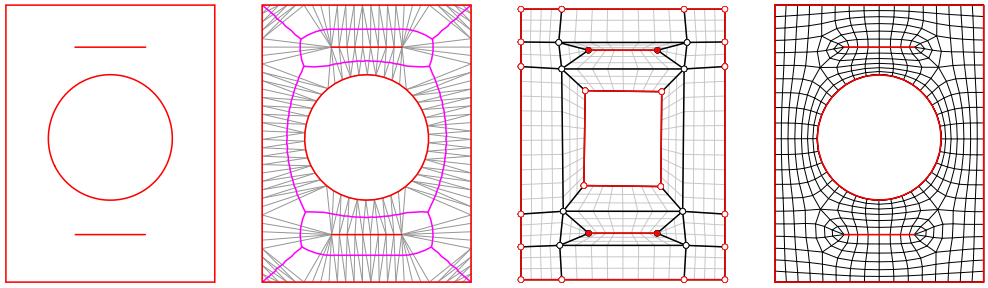
3.2.3. Curve features

Including curve features allows to orientate a pattern along specific directions, for instance along a wall support or a crease to be shaped during form finding, which have to be represented as a continuous set of edges. More generally, curve features can be used to explore different pattern topologies.

They are included in the input as curves on the surface, as shown in Figure 20a. These curves are subdivided into a set of points that are also added to the set of vertices for the constrained Delaunay triangulation, where the edges from the curve features are set as constraints [53], as shown in Figure 20b. The Delaunay mesh displays thereby new singular faces adjacent to the curve features. Additionally, topological cuts are made in the Delaunay mesh along the curve features to consider them as boundaries. Otherwise, some of the faces along these edges would have three adjacent faces and be considered as singular faces instead of regular faces and yield unwanted medial branches crossing the curve feature edges. The resulting singularity mesh in Figure 20c presents pseudo-quad faces at the curve feature extremities, which yield partial pole points that are adjacent to both pseudo quad and quad faces in the pattern in Figure 20d.

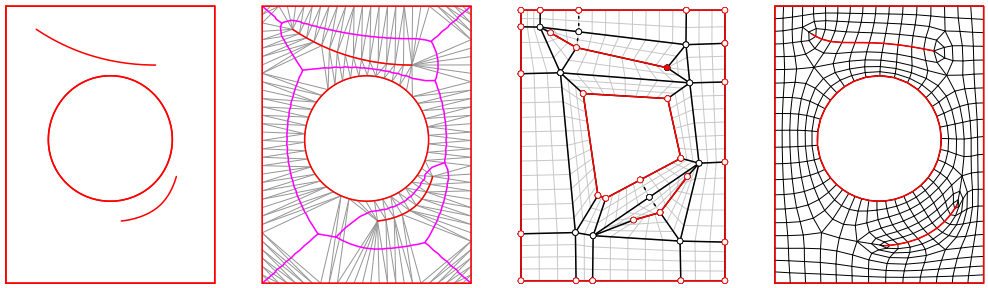
However, these poles do not systematically occur, as shown in Figure 21 where some extremities yield a two-valent singularity in a pattern that still respects curve feature alignment and edge path concentration. Nevertheless, the designer can choose to enforce or remove these partial poles, automatically or not, as presented in Section 4.1.2 using strip grammar rules. In the case of curve features spanning from boundary to boundary, no poles occur, as shown in Figure 22, unless boundary point features are included at the curve feature extremities.

Moreover, the topological cuts in the Delaunay meshes along the curve features can induce discrepancies which are compensated in a new step



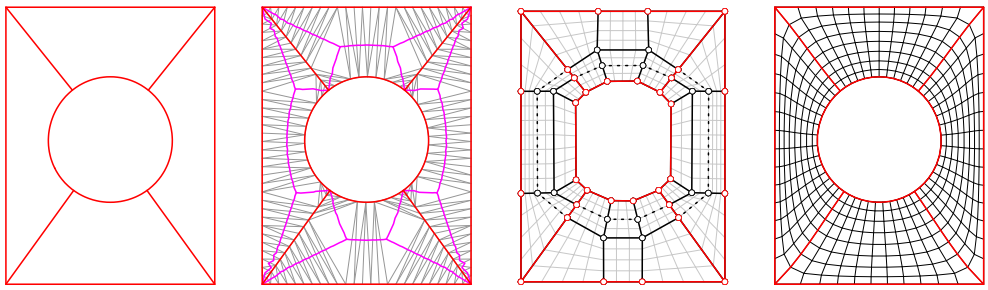
(a) Input (b) Medial axis (c) Singularity mesh (d) Pattern

Figure 20: Skeleton-based generation of a pattern integrating curve features with pole extremities. The input surface is marked in red, the Delaunay triangulation in dark grey, the topological skeleton in pink, the singularity mesh in black, the density mesh in light grey and the pattern in black.



(a) Input (b) Medial axis (c) Singularity mesh (d) Pattern

Figure 21: Skeleton-based generation of a pattern integrating curve features with hybrid extremities. The input surface is marked in red, the Delaunay triangulation in dark grey, the topological skeleton in pink, the singularity mesh in black, the density mesh in light grey and the pattern in black.



(a) Input (b) Medial axis (c) Singularity mesh (d) Pattern

Figure 22: Skeleton-based generation of a pattern integrating boundary-to-boundary curve features without pole extremities. The input surface is marked in red, the Delaunay triangulation in dark grey, the topological skeleton in pink, the singularity mesh in black, the density mesh in light grey and the pattern in black.

by adding edges across the curve features in the singularity mesh, represented as dashed lines in Figures 21c and 22c. This procedure is detailed in Section 4.2.2.

These curves can be set as constraints during smoothing, as in Figure 20d, or not, as in Figures 21d and 22d, if they are only meant as guiding features from the designer.

As shown in Figures 20 to 22, additional curves can be used within the skeleton-based decomposition algorithm as a means for feature-based topology finding to explore the topology of patterns. These features can stem from the a design intent, intuition or heuristic, as illustrated in the following

case study.

3.2.4. Heuristics

Point and curve features have been introduced in order to comply intuitively to requirements on the structural patterns. These heuristics are assessed with a short case study revisiting the British Museum roof and comparing designs with different topologies.

The actual shape [41] is used to map the generated patterns. Only quad mesh patterns are compared, therefore the actual triangulated pattern is not considered. Engineering and construction details are found in Sischka *et al.* [54]. Thrust is only permitted at the four corners, as the shell is supported along its boundary by sliding bearings to avoid applying thrust on the existing building. The additional stiffening systems are discarded to focus on the difference of structural behaviour due to the topology of the patterns and their singularities.

The four tested patterns are shown in Figure 23. Four singularity meshes are generated using feature-based topology finding to assess their efficiency for the given statics system. These quad mesh patterns stem from:

(a) the singularity mesh based only on the surface (Figure 23a);

(b) the singularity mesh based on the surface with four curve features to provide direct edge/force paths towards the thrust corners along the longest span following a designer's intuition (Figure 23c);

(c) the singularity mesh based on the surface with four point features to concentrate edge/force paths to the thrust corners based on the project's context (Figure 23b);

(d) the singularity mesh based on the surface with both the curve and point features (Figure 23d).

The density is set with a target length of 1.5m, similar to the real project [54], resulting in 5049 to 6049 beams, to be compared with the 4878 of the actual triangulated pattern. The quad mesh pattern is relaxed on the surface using Laplacian smoothing with constraints to re-project boundary vertices on the boundaries with fixed corners.

Based on the Eurocodes, the tested load cases are:

- the structural self-weight G;
- a downwards dead load $G' = 0.6kN/m^2$ for a 24mm thick glazing [54];
- a downwards projected live load $Q = 1.5kN/m^2$ for snow loads, without taking into account geometry factors;

and the relevant load combinations are:

- the Serviceability Limit State (SLS): $1.0(G + G') + 1.0Q$;
- the Ultimate Limit State (ULS): $1.35(G+G')+ 1.5Q$.

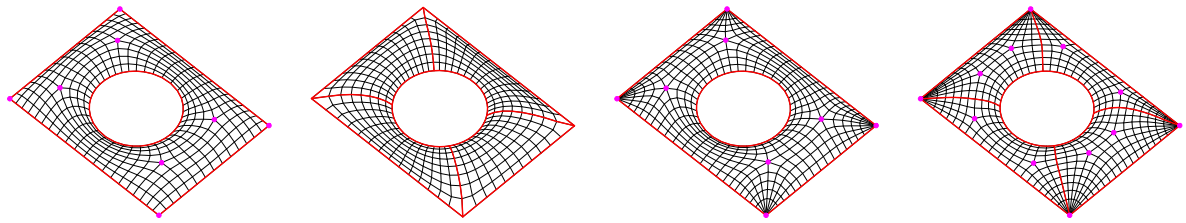
The beams of the actual structure have a box cross section with a width of 80mm and a height varying from 80mm to 200mm, oriented with the surface normal. For this case study, the S355 steel beams all have the same cross section to favour designs with a homogeneous force flow. The beams must be stiffer, as the quad mesh is not triangulated: they have a width of 250mm, and an assumed wall thickness of 20mm. The height of the beams is minimised to reduce the structural weight, while complying with the following structural requirements:

- a maximal SLS deflection of 140mm, corresponding to the maximal span over 200, though the deflection of the actual structure is compensated with a pre-deformation [54];
- a maximal ULS stress utilisation of 100%;
- a minimal ULS first load buckling factor of 4, as for the actual structure [54].

The pre-deformation as well as the imperfections, based on the first buckling mode with a maximal value of 140mm [54], are not taken into account. The poor support conditions favour the bending behaviour of the shell rather than its membrane behaviour, and therefore buckling is not expected to be the critical requirement.

A second order mechanical analysis is performed using the Finite Element Analysis tool Karamba [55], with the results displayed in Table 1.

The structural performance is assessed as the ratio of the structural mass over the projected area of the shell. As expected, the buckling requirement is secondary, and deflection is the decisive



(a) Standard (b) With curve features (c) With point features (d) With both features

Figure 23: Four patterns for the British Museum roof with different topologies, to assess the relevance of heuristic point and curve features, shown in red. For readability, the displayed density is reduced from a target length of 1.5m to 5m.

requirement. Except for the topology with the curve features, for which utilisation is the decisive requirement because of the stress concentrations in the hoops close to the corners, not featured in the other topologies. The unique cross-section requirement penalises this topology ad makes it the least efficient one. The most efficient design is the one including both point and curve features, with 62% of the weight of the design without features. This topology is close to an optimum for the given constraints as it uses 99% of the stiffness limit, 88% of strength one and 89% of the stability one. Moreover, including only the poles yields the second most efficient design, with 66% of the weight of the topology without features.

Table 1: Comparison of the structural performance after sizing optimisation of the designs with different topologies in Figure 23.

metric	(a)	(b)	(c)	(d)
number of edges [-]	5915	5049	5743	6049
beam height [mm]	430	590	220	180
projected surface weight [kg/m^2]	317	371	210	195
max. SLS deflection [mm]	138	95	138	138
max. ULS utilisation [-]	83%	99%	74%	88%
first ULS buckling load factor [-]	23.6	34.6	7.8	4.5

In this example, taking into account the support conditions via heuristics such as point and curve features when designing the topology of the pattern improved the mechanical behaviour of the structural pattern, especially thanks to poles at the location of concentrated forces.

3.3. Shape topology extension

The presented skeleton-based algorithm applies only to orientable surfaces with zero handles g and at least one boundary N ($g = 0$, $N \geq 1$), to allow seamless planar mapping for the Delaunay triangulation.
Edge network thickening approaches [56] can generate coarse quad meshes for high-genus skeletal surfaces.
Yet, the presented algorithm can be extended to generate shapes with any topology ($g \geq 0$, $N \geq 0$), as shown in Figure 24. This approach is relevant for high-genus surfaces that can be defined via a medial surface, as for the ICD/ITKE Research Pavilion 2015-16 [57]:

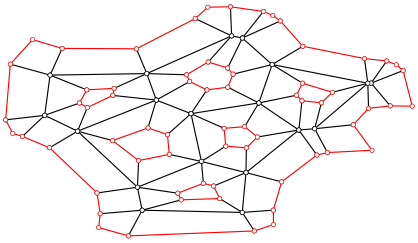
1. an open null-genus topology ($g = 0$, $N = a \geq 1$) is generated following the presented algorithm (Figure 24a);

2. the topology is thickened to obtain a closed non-null-genus topology ($g = a-1 \geq 0$, $N = 0$) after offsetting it and adding faces to join the boundaries together (Figure 24b);

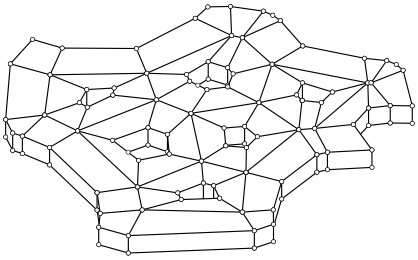
3. the topology becomes an open non-null-genus ($g = a - 1 \geq 0$, $N = b \geq 0$) by perforating some of its faces (Figure 24c).

3.4. Algorithm performance

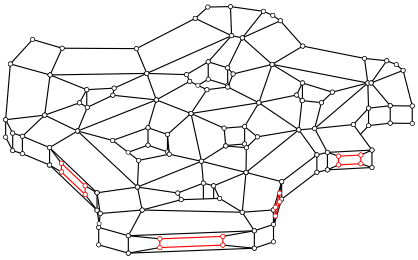
These topology-finding algorithms are meant to be used at an early design stage for the exploration of a large variety of designs. Therefore, the computation time of the algorithms should be fast enough for efficient application, whether for fluid user-machine interactivity or for automated or partially-automated generation of numerous designs.
The computation time of the singularity mesh



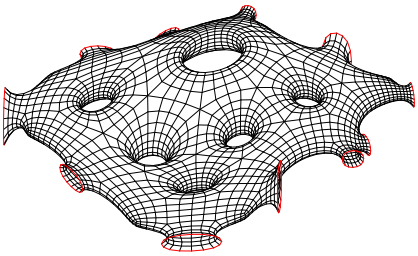
(a) Open null-genus topology
from the medial axis of an input surface



(b) Closed non-null-genus topology
after thickening



(c) Open non-null genus topology
after perforating faces



(d) Open non-null genus pattern
after further mesh modelling

Figure 24: Extension to general orientable compact manifold topologies with multiple handles and boundaries (in red): the generated pattern has a topology with 6 handles and 11 boundaries.

of the pattern in Figure 23a takes between 100 and 800ms with the same results for different Delaunay meshes with 50 to 483 faces, resulting from subdivision values between 8% and 0.8% of the length of the bounding box diagonal. For comparison, on the same machine, the structural analysis of the corresponding gridshell with 5915 beam elements took about 350ms for an elastic linear analysis and 2300ms for a buckling analysis of the first mode taking into account the axial forces in the geometrical stiffness matrix, on an average of 5 computations. The developed topology-finding algorithm takes less time than the structural analysis performed by a commercial software plugin, making it suitable for engineering design applications.

Skeleton-based generation of the singularities of a pattern provides an initial topology in the singularity design space. Even though the singularities can be modified indirectly using curve features, more explicit and controlled topological exploration of the design space can be performed by combining feature-based topology finding with rule-based exploration by applying grammar rules on the lower level of the singularity mesh elements, and more specifically on its strips.

4. Rule-based exploration

The singularity design space is a topological space without metric to organise it, on the contrary to a geometrical space like the one related to the shape of the pattern, which can be explored using the vertex coordinates as continuous-valued parameters. Nonetheless, topological spaces can be explored via grammars of rules that perform topological operations, instead of modifying values.

Shape grammars, introduced by Stiny and Gips [58], evolved into functional grammars and then into structural grammars to include non-geometrical data related to structures [59, 60, 61, 62]. Regarding shell-like structures, Shea and Cagan introduce a grammar used for optimisation of triangulated meshes of geodesic domes using shape annealing [63].

Grammar rules for singularity meshes must apply specifically to (coarse) pseudo-quad meshes, to constrain exploration to the singularity design space, as performed by existing quad mesh simplification operations, like deletion or rotation, but on dense and unstructured quad meshes [64, 65, 66, 67]. This grammar must allow both an increase and a decrease in the complexity of

the singularity meshes, and include pole editing via pseudo-quads. This section introduces thus a grammar based on the addition and deletion of strips, for user- and algorithm-guided exploration of the singularity design space.

4.1. Lowest-level strip grammar

The fundamental grammar is composed of two opposite atomic rules that apply at the lowest level possible on the natural element description of quad meshes: 'add strip' and 'delete strip'. Poles are optionally included through pseudo-quad faces at the strip extremities to extend to pseudo-quad meshes. These rules ensure:

- to perform exploration constrained to the space of singularity meshes, unlike rules such as adding and deleting edges which can generate any polygonal mesh;
- to achieve any singularity mesh, unlike high-level rules, such as the ones in *et al.* [68] that apply too specific modifications.

These rules can be applied to all orientable quad meshes with any shape topology ($g \geq 0, N \geq 0$) and used as parameters for the singularity design space to perform algorithmic search.

4.1.1. Grammar rule algorithm

The two opposite rules are:

- 'add strip', which inserts a strip along a poly-edge;
- 'delete strip', which collapses a strip into a poly-edge.

The algorithm is illustrated in Figure 25 in the case of an open boundary-to-boundary strip or poly-edge. The input to add a strip is the poly-edge [A, B, C, D]. Starting from a poly-edge extremity, each edge is unzipped by inserting a pseudo-quad face with its double vertex towards the downstream edges, which converts the previous pseudo-quad into a quad, with an exception for the last edge that directly generates a quad. If an extremity of the strip is on the boundary, it can be converted into poles or not, as in Figure 26, but an extremity lying outside the boundary has to be a pole. The zero-length edges from pseudo-quads can be included in poly-edges along which to add strips. Deleting a strip follows the reversed steps of adding a strip.

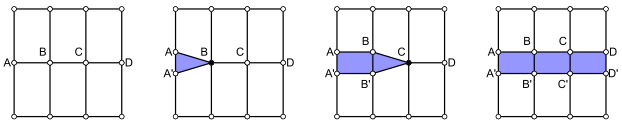


Figure 25: Add/delete an open strip (in blue).

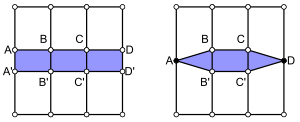


Figure 26: Strip (in blue) with or without poles

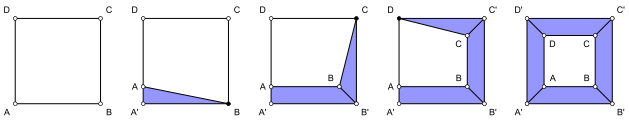


Figure 27: Add/delete a closed strip (in blue).

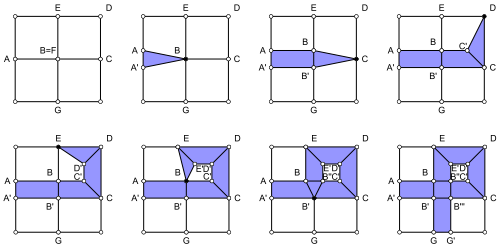


Figure 28: Add/delete a self-crossing strip (in blue).

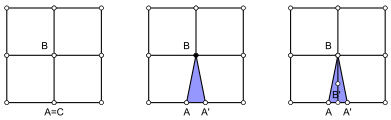


Figure 29: Add/delete a self-overlapping strip (in blue).

Additional specific operations are necessary to be able to add/delete strips in any configuration.

To add a closed strip from the poly-edge [A, B, C, D, A] in Figure 27, the extremity from the last edge (D-A) connects to the one from the first edge (A-B).

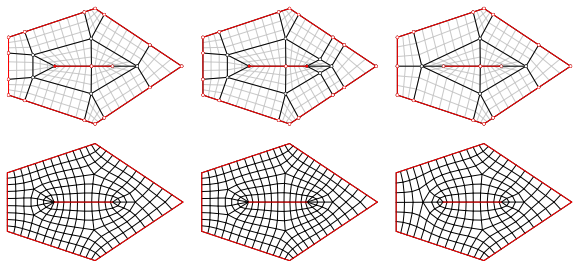
To add a self-crossing strip from the poly-edge [A, B, C, D, E, B, G] in Figure 28, the multiply occurring vertex B is updated in the poly-edge becoming [..., C, D, E, B, B', G] to integrate the edges (E-B), (B-B') and (B-G).

To add a self-overlapping strip from the poly-edge [A, B, A] in Figure 29, the multiply occurring edge (A-B) is updated in the poly-edge becoming [..., B, A'].

4.1.2. Applications

The rules can be applied in different manners, to generate specific designs or explore multiple ones combined with feature-based exploration.

Editing curve features extremities. Curve feature extremities can yield a pole or not using the skeleton-based generation procedure, as in Figure 30a, where one of the curve feature extremities is a pole and the other one a two-valent singularity. Strip rules can be applied to either add strips with poles from the curve feature extremity to the boundary via the shortest poly-edges, as in Figure 30b, or delete the strips with poles, as in Figure 30c.



(a) Hybrid extremities (b) Poles after adding strips (c) No poles after deleting strips

Figure 30: Editing the extremities of curve features (in red) by adding or removing poles.

Editing skeleton-based singularity meshes. Starting from the skeleton-based singularity mesh, strip rules can apply specific and controlled modifications to explore further the singularity design space, as illustrated in Figure 31. To preserve point and curve features, deletions of all the strips integrating them is prohibited.

Editing point features. Pole points can be generated using point features in skeleton-based generation. Nevertheless, pole points can still be added using strip rules with pseudo-quads at their extremities, on the boundary or not, as illustrated in Figure 32. The dual force diagrams used to explore force equilibria with Thrust Network Analysis and RhinoVAULT highlight the dual modifications of the strip rules, which modify the degrees of freedom for funicular form finding. In this example, the topology of a fully-supported vault is modified by adding pole point features at the corners and at the centre.

The strip rules are the lowest level of topological operations that allow to evolve new singularity meshes, and can be used as parameters for exploration of the singularity design space. However, the designer may think in terms of more specific and local modifications. The next section proposes thus to combine the lowest-level strip rules into high-level ones.

4.2. High-level grammar

4.2.1. Extended rules

Strip rules can be combined to develop any high-level rules that evolve a singularity mesh into another one, as in Figure 33. Such extended rules can be stored to form a practical grammar [68]. For the designer, practical grammars are meant to be efficient, as specific high-level modifications can be applied faster than multiple low-level modifications, and flexible, as more rules can be added to the grammar. A practical rule can be applied locally to a part of the singularity mesh before propagating these modifications to the mesh.

4.2.2. Propagation

When applying a local high-level rule, such as the one from Figure 33, new boundary vertices modify adjacent quad faces that become pentagons or general polygons, as in Figure 34. A propagation procedure is necessary to spread the local modifications from these source points. New edges are added that subdivide the faces adjacent to the sources. Termination occurs when the source points reach a boundary edge in the case of open strips, as in Figure 34a, or meet another source in the case of closed strips, as in Figure 34b.

However, if propagation of an asymmetric number of sources must be spread in a closed strip, singularities must be added, as seen in Figure 34c. The patterns developed by Takayama *et al.* [69] are used to quadrangulate four-sided polygons with a given number of edge subdivisions resulting

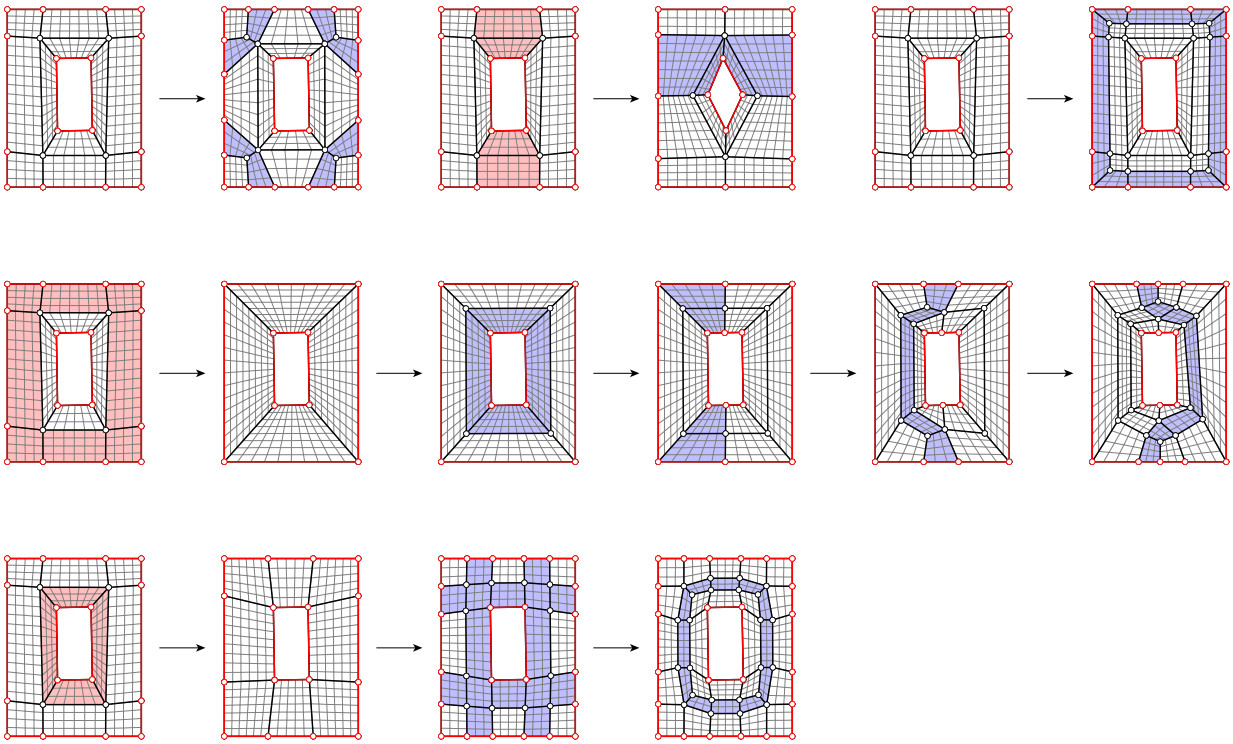


Figure 31: Editing skeleton-based singularity meshes using rule-based exploration. The singularity meshes are in black with boundaries in red, the newly added strips are highlighted in light blue and the to be deleted strip in light red.

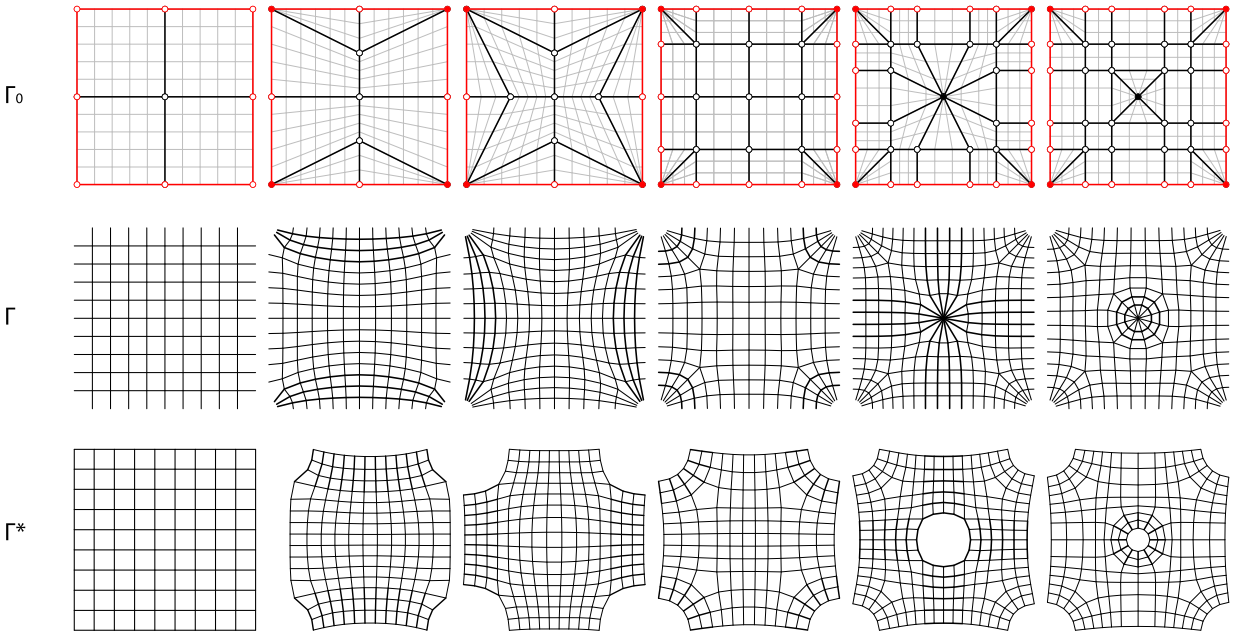


Figure 32: Editing point features sequentially using strip rules with pole extremities. In the context of Thrust Network Analysis, the modifications in the primal form diagram and the dual force diagram are highlighted in blue, as added strips and poly-edges, respectively. Singularity meshes are in black with boundaries in red, and density meshes and patterns in grey.

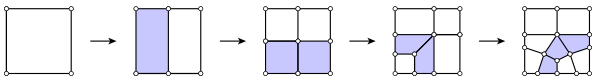
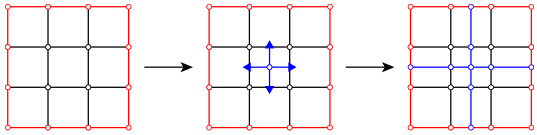
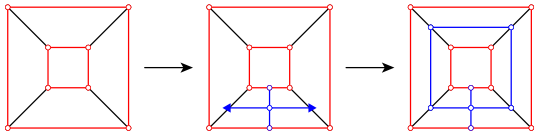


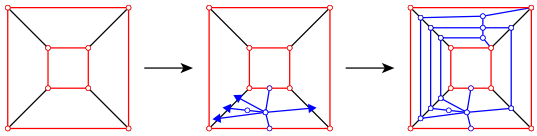
Figure 33: Combining lowest-level strip rules to develop high-level rules. New strips are highlighted in blue.



(a) Open strip



(b) Closed strip with symmetric subdivision



(c) Closed strip with asymmetric subdivision

Figure 34: Propagation procedure from source vertices resulting from local editing. Boundaries are shown in red and new edges in blue.

in a minimal number of singularities. Takayama's algorithm requires an even number of polygon edges, ensured by subdividing the strips of one of these edges, if necessary.

The same propagation procedure is used for the integration of curve features in the skeleton-based generation scheme. The unwelding of the Delaunay mesh along them induces discrepancies, which are used as propagation sources. More precisely, the sources result from the B points that lie on the curve feature instead of the boundary.

4.2.3. Simplification

Whereas the two opposite strip rules can compensate each other, a new simplification rule is necessary to compensate any of the previously applied high-level rules, independently from the sequence history, to allow non-linear exploration instead of linearly cancelling the previous steps.

The simplification rule is inspired by the approach of Verma and Suresh [70] to reduce the number of singularities in quad meshes but resorts again to the quadrangulated patterns of Takayama *et al.* [69], more relevant for coarse quad meshes. As shown in Figure 35, a closed poly-edge is selected, the encompassed faces are merged and one of Takayama's patterns is applied. The polygon is defined by the main vertices in the poly-edge, controlled by the designer to choose among different topologies.

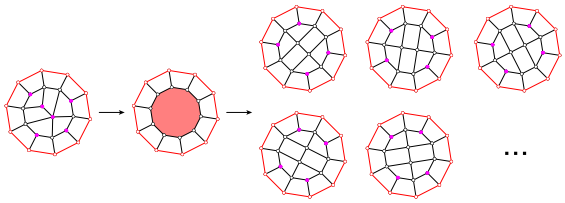


Figure 35: Local simplification rule of a singularity mesh based on Takayama's patterns *et al.* [69], yielding a minimal number of singularities. Singular vertices and faces are highlighted in pink.

The presented rule-based approach for topological exploration of singularity meshes supplements feature-based exploration, with a lowest-level strip grammar and an extended high-level grammar.

5. Conclusion and future work

This paper tackled *topology finding* of patterns for shell-like structures with a focus on singularities in quad-based mesh patterns, complementary to form finding and other geometrical approaches. The designed patterns are structured, i.e. with a low number of singularities, and aligned with features like surface boundaries, points and curves.
To allow this topological exploration, a specific design space structure with its data structure and parameters were introduced, based on singularity meshes as coarse pseudo-quad meshes that encode the information related to the singularities in the pattern, including high-valency pole points. A skeleton-based generation procedure for feature-based exploration of singularity meshes on input surfaces with point and curve features was presented. These features can heuristically integrate statics considerations, as illustrated on a case study. A rule-based exploration means supporting feature-based exploration was developed with a lowest-level grammar based on strip editing and a high-level grammar resulting from the combination of strip rules.

References

[1] Lionel Du Peloux, Frédéric Tayeb, Jean-François Caron, and Olivier Baverel. The ephemeral cathedral of Créteil: a 350m2 lightweight gridshell structure made of 2 kilometers of GFRP tubes. In *Conférence Internationale de Géotechniques, des Ouvrages et Structures 2015: Innovations in Construction*, 2015. URL `https://hal.archives-ouvertes.fr/hal-01199044`. Accessed on 26/10/2018.

[2] Matthias Rippmann, Tom Van Mele, Mariana Popescu, Edyta Augustynowicz, Tomás Méndez Echenagucia, Cristián Calvo Barentin, Ursula Frick, and Philippe Block. The Armadillo Vault: computational design and digital fabrication of a freeform stone shell. In Sigrid Adriaenssens, Fabio Gramazio, Matthias Kohler, Achim Menges, and Mark Pauly, editors, *Advances in Architectural Geometry 2016*, pages 344–363. vdf Hochschulverlag an der ETH Zürich, 2016. doi: 10.3218/3778-4_23.

[3] Tomás Méndez Echenagucia, David Pigram, Andrew Liew, Tom Van Mele, and Philippe Block. A cable-net and fabric formwork system for the construction of concrete shells: design, fabrication and construction of a full scale prototype. *Structures*, 2018. doi: 10.1016/j.istruc.2018.10.004.

[4] Romain Mesnil, Cyril Douthe, Olivier Baverel, and Tristan Gobin. Form finding of nexorades using the translations method. *Automation in Construction*, 95:142–154, 2018. doi: 10.1016/j.autcon.2018.08.010.

[5] Hans-Jörg Schek. The force density method for form finding and computation of general networks. *Computer Methods in Applied Mechanics and Engineering*, 3(1):115–134, 1974. doi: 10.1016/0045-7825(74)90045-0.

[6] Michael R. Barnes. Form finding and analysis of tension structures by dynamic relaxation. *International Journal of Space Structures*, 14(2):89–104, 1999. doi: 10.1260/0266351991494722.

[7] Philippe Block and John Ochsendorf. Thrust Network Analysis: A new methodology for three-dimensional equilibrium. *Journal of the International Association for Shell and Spatial Structures*, 48(3):167–173, 2007. ISSN 1996-9015.

[8] Kai-Uwe Bletzinger and Ekkehard Ramm. A general finite element approach to the form finding of tensile structures by the updated reference strategy. *International Journal of Space Structures*, 14(2):131–145, 1999. doi: 10.1260/0266351991494759.

[9] James Glymph, Dennis Shelden, Cristiano Ceccato, Judith Mussel, and Hans Schober. A parametric strategy for free-form glass structures using quadrilateral planar facets. *Automation in Construction*, 13(2):187–202, 2004. doi: 10.1016/j.autcon.2003.09.008.

[10] Romain Mesnil, Cyril Douthe, Olivier Baverel, and Bruno Léger. Marionette meshes: modelling free-form architecture with planar facets. *International Journal of Space Structures*, 32(3-4):184–198, 2017. doi: 10.1177/0266351117738379.

[11] Kai-Uwe Bletzinger and Ekkehard Ramm. Form finding of shells by structural optimization. *Engineering with computers*, 9(1):27–35, 1993. doi: 10.1007/BF01198251.

[12] Romain Mesnil, Cyril Douthe, Christiane Richter, and Olivier Baverel. Fabrication-aware shape parameterisation for the structural optimisation of shell structures. *Engineering Structures*, 176:569–584, 2018. doi: 10.1016/j.engstruct.2018.09.026.

[13] Anthony George Maldon Michell. Lviii. The limits of economy of material in frame-structures. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 8(47):589–597, 1904. doi: 10.1080/14786440409463229.

[14] George I. N. Rozvany and William Prager. Optimal design of partially discretized grillages. *Journal of the Mechanics and Physics of Solids*, 24(2-3):125–136, 1976. doi: 10.1016/0022-5096(76)90022-3.

[15] Yang Liu, Helmut Pottmann, Johannes Wallner, Yong-Liang Yang, and Wenping Wang. Geometric modeling with conical meshes and developable surfaces. *Association for Computing Machinery Transactions on Graphics*, 25(3):681–689, 2006. doi: 10.1145/1141911.1141941.

[16] Martin Philip Bendsøe and Ole Sigmund. *Topology optimization: theory, methods, and applications*. Springer Science & Business Media, 2013. ISBN 978-3-662-05086-6.

[17] Andrew Borgart. New challenges for the structural morphology group. *Journal of the International Association for Shell and Spatial Structures*, 51(3):183–189, 2010. ISSN 1996-9015.

[18] Lauren L. Beghini, Alessandro Beghini, Neil Katz, William F. Baker, and Glaucio H. Paulino. Connecting architecture and engineering through structural topology optimization. *Engineering Structures*, 59:716–726, 2014. doi: 10.1016/j.engstruct.2013.10.032.

[19] Jörg Schlaich, Hans Schober, and Kai Kürschner. New trade fair in Milan – grid topology and structural behaviour of a free-formed glass-covered surface. *International Journal of Space Structures*, 20(1):1–14, 2005. doi: 10.1260/0266351054214326.

[20] John Harding, Sam Joyce, Paul Shepherd, and Christopher J. K. Williams. Thinking topologically at early stage parametric design. In Lars Hesselgren, Shrikant Sharma, Johannes Wallner, Niccolo Baldassini, Philippe Bompas, and Jacques Raynaud, editors, *Advances in Architectural Geometry 2012*. Springer, 2012. ISBN 978-3709112502.

[21] Anders Holden Deleuran, Mark Pauly, Martin Tamke, Ida Friis Tinning, and Mette Ramsgaard Thomsen. Exploratory topology modelling of form-active hybrid structures. *Procedia Engineering*, 155:71–80, 2016. doi: 10.1016/j.proeng.2016.08.008.

[22] Seiichi Suzuki and Jan Knippers. The design implications of form-finding with dynamic topologies. In *Humanizing Digital Reality: Design Modelling Symposium Paris 2017*, pages 211–223. Springer, 2017. ISBN 978-9811066108.

[23] Soeren Stephan, Jaime Sánchez-Alvarez, and Klaus Knebel. Reticulated structures on free-form surfaces. *Stahlbau*, 73(8):562–572, 2004. doi: 10.1002/stab.200490149.

[24] Robin Oval. compas_pattern: a python framework for topology finding of patterns for shell structures, 2017. URL `https://github.com/BlockResearchGroup/compas_pattern`. Accessed on 09/01/2019.

[25] Tom Van Mele, Andrew Liew, Tomás Méndez Echenagucia, Matthias Rippmann, et al. compas: A framework for computational research in architecture and structures, 2017. URL `http://compas-dev.github.io/compas/`. Accessed on

01/09/2018.

[26] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon mesh processing*. CRC press, 2010. ISBN 978-1568814261.

[27] Paul Shepherd and Paul Richens. The case for subdivision surfaces in building design. *Journal of the International Association for Shell and Spatial Structures*, 53(4):237–245, 2012. ISSN 1996-9015.

[28] Vishu Bhooshan, Shajay Bhooshan, and Philippe Block. Mayavault - A mesh modelling environment for discrete funicular structures. *Nexus Network Journal*, pages 1–16, 2018. doi: 10.1007/s00004-018-0402-z.

[29] Romain Mesnil, Cyril Douthe, Olivier Baverel, and Bruno Leger. Generalised cyclidic nets for shape modelling in architecture. *International Journal of Architectural Computing*, 15(2):148–168, 2017. doi: 10.1177/1478077117714917.

[30] John Renze, Todd Rowland, and Eric W. Weisstein. Compact manifold. From MathWorld – a Wolfram web resource, 2018. URL `http://mathworld.wolfram.com/CompactManifold.html`. Accessed on 25/11/2018.

[31] Marcel Campen, David Bommes, and Leif Kobbelt. Dual loops meshing: quality quad layouts on manifolds. *Association for Computing Machinery Transactions on Graphics*, 31(4):110, 2012. doi: 10.1145/2185520.2185606.

[32] Ergun Akleman, Shenyao Ke, You Wu, Negar Kalantar, AliReza Borhani, and Jianer Chen. Construction with physical version of quad-edge data structures. *Computers & Graphics*, 58:172–183, 2016. doi: 10.1016/j.cag.2016.05.008.

[33] Edwin Catmull and James Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, 1978. doi: 10.1016/0010-4485(78)90110-0.

[34] Jörg Schlaich and Hans Schober. Glass roof for the Hippo House at the Berlin Zoo. *Structural Engineering International*, 7(4):252–254, 1997. doi: 10.2749/101686697780494581.

[35] Lionel Du Peloux, Olivier Baverel, Jean-François Caron, and Frédéric Tayeb. From shape to shell: a design tool to materialize freeform shapes using gridshell structures. In *Rethinking Prototyping: Design Modelling Symposium Berlin 2013*, 2013. ISBN 9783844268454.

[36] John H. Conway, Heidi Burgiel, and Chaim Goodman-Strauss. *The symmetries of things*. CRC Press, 2016. ISBN 978-1568812205.

[37] Paul Shepherd and Will Pearson. Topology optimisation of algorithmically generated space frames. In *Proceedings of the Annual Symposium of the International Association for Shell and Spatial Structures 2013*, 2013.

[38] Antiopi Koronaki, Paul Shepherd, and Mark Evernden. Layout optimization of space frame structures. In *Proceedings of the Annual Symposium of the International Association for Shell and Spatial Structures 2017*, 2017.

[39] Christopher J. K. Williams. Patterns on a surface: the reconciliation of the circle and the square. *Nexus Network Journal*, 13(2):281–295, 2011. doi: 10.1007/s00004-011-0068-2.

[40] Harry Blum. A transformation for extracting new descriptors of shape. *Models for Perception of Speech and Visual Forms, 1967*, pages 362–380, 1967.

[41] Christopher J. K. Williams. The analytic and numerical definition of the geometry of the British Museum Great Court Roof. In *Mathematics & Design*, pages 434–440. Deakin University, 2001. ISBN 0-7300-2526-8.

[42] David Rigby. Topmaker: a technique for automatic multi-block topology generation using the medial axis. In *American Society of Mechanical Engineers/Japan Society of Mechanical Engineers 2003 4th Joint Fluids Summer Engineering Conference*, pages 1991–1997, 2003. doi: 10.1115/FEDSM2003-45527.

[43] Harold J. Fogg, Cecil G. Armstrong, and Trevor T. Robinson. Enhanced medial-axis-based block-structured meshing in 2-D. *Computer-Aided Design*, 72:87–101, 2016. doi: 10.1016/j.cad.2015.07.001.

[44] Allison B. Halpern, David P. Billington, and Sigrid Adriaenssens. The ribbed floor slab systems of Pier Luigi Nervi. *Journal of the International Association for Shell and Spatial Structures*, 2013(23):1–7, 2013. ISSN 1996-9015.

[45] Hans-Dieter Hecker. Der Hörsaal des Zoologischen Instituts der Universität Freiburg. *Freiburger Universitätsblätter*, 25:49–52, 1969. ISSN 0016-0717.

[46] Sigrid Adriaenssens, Laurent Ney, Eric Bodarwe, and Christopher J. K. Williams. Finding the form of an irregular meshed steel and glass shell based on construction constraints. *Journal of Architectural Engineering*, 18(3):206–213, 2012. doi: 10.1061/(ASCE)AE.1943-5568.0000074.

[47] Horst Dürr. Seilnetze-Planung, Berechnung, Ausführung und Werkplanung. *Stahlbau*, 69(8):585–594, 2000. doi: 10.1002/stab.200002190.

[48] Tom Van Mele, Daniele Panozzo, Olga Sorkine-Hornung, and Philippe Block. Best-fit thrust network analysis – Rationalisation of freeform meshes. In Sigrid Adriaenssens, Philippe Block, Diederik. Veenendaal, and Christopher J. K. Williams, editors, *Shell Structures for Architecture: Form Finding and Optimization*, chapter 13, pages 157–169. Routledge, London, 2014. doi: 10.4324/9781315849270.

[49] Daniele Panozzo, Philippe Block, and Olga Sorkine-Hornung. Designing unreinforced masonry models. *Association for Computing Machinery Transactions on Graphics*, 32(4):91, 2013. doi: 10.1145/2461912.2461958.

[50] Robin Oval, Matthias Rippmann, Tom Van Mele, Olivier Baverel, and Philippe Block. Patterns for Masonry Vault Design. In *Proceedings of the Annual Symposium of the International Association for Shell and Spatial Structures 2017*, Hamburg, Germany, 2017.

[51] Tom Van Mele and Philippe Block. A novel form finding method for fabric formwork for concrete shells. *Journal of the International Association of Shell and Spatial Structures*, 52(217224):31, 2011. doi: 10.1.1.701.6049.

[52] Matthias Rippmann, Lorenz Lachauer, and Philippe Block. Interactive vault design. *International Journal of Space Structures*, 27(4):219–230, 2012. doi: 10.1260/0266-3511.27.4.219.

[53] Paul L. Chew. Constrained Delaunay triangulations. *Algorithmica*, 4(1-4):97–108, 1989. doi: 10.1007/BF01553881.

[54] Johann Sischka, St Brown, Erich Handel, and Gunther Zenkner. Die Überdachung des Great Court im British Museum in London. *Stahlbau*, 70(7):492–502, 2001. doi: 10.1002/stab.200101690.

[55] Clemens Preisinger and Moritz Heimrath. Karamba - A toolkit for parametric structural design. *Structural Engineering International*, 24(2):217–221, 2014. doi:

10.2749/101686614X13830790993483.

[56] Vinod Srinivasan. *Modeling high-genus surfaces*. PhD thesis, Texas A&M University, 2004. URL `https://pdfs.semanticscholar.org/f7c7/64fb76f617172152ef57e3184ddc8792d9f5.pdf`. Accessed on 22/10/2018.

[57] Daniel Sonntag, Simon Bechert, and Jan Knippers. Biomimetic timber shells made of bending-active segments. *International Journal of Space Structures*, 32 (3-4):149–159, 2017. doi: 10.1177/0266351117746266.

[58] George Stiny and James Gips. Shape grammars and the generative specification of painting and sculpture. In *Proceedings of the Congress International Federation for Information Processing 1971*, pages 1460–1465, 1971. doi: 10.1.1.151.7931.

[59] William J. Mitchell. Functional grammars: an introduction. In *Reality and Virtual Reality: Association for Computer Aided Design in Architecture Conference Proceedings 1991*, pages 167–176. University of California at Los Angeles, 1991. ISBN 1-880250-00-4.

[60] Robert Baldock. *Structural optimisation in building design practice: case-studies in topology optimisation of bracing systems*. PhD thesis, University of Cambridge, 2007. URL `https://www.ethz.ch/content/dam/ethz/special-interest/mavt/design-materials-fabrication/engineering-design-and-computing-lab/DoctoralTheses/PhD_Baldock.pdf`. Accessed on 26/10/2018.

[61] Philipp Geyer. Multidisciplinary grammars supporting design optimization of buildings. *Research in Engineering Design*, 18(4):197–216, 2008. doi: 10.1007/s00163-007-0038-6.

[62] Caitlin T. Mueller. *Computational exploration of the structural design space*. PhD thesis, Massachusetts Institute of Technology, 2014. URL `https://dspace.mit.edu/handle/1721.1/91293`. Accessed on 01/09/2018.

[63] Kristina Shea and Jonathan Cagan. Innovative dome design: applying geodesic patterns with shape annealing. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 11(5):379–394, 1997. doi: 10.1017/S0890060400003310.

[64] Joel Daniels II, Cláudio T. Silva, Jason Shepherd, and Elaine Cohen. Quadrilateral mesh simplification. *Association for Computing Machinery Transactions on Graphics*, 27(5):148, 2008. doi: 10.1145/1409060.1409101.

[65] Joel Daniels II, Cláudio T. Silva, and Elaine Cohen. Localized quadrilateral coarsening. *Computer Graphics Forum*, 28(5):1437–1444, 2009. doi: 10.1111/j.1467-8659.2009.01520.x.

[66] Marco Tarini, Nico Pietroni, Paolo Cignoni, Daniele Panozzo, and Enrico Puppo. Practical quad mesh simplification. *Computer Graphics Forum*, 29(2):407–418, 2010. doi: 10.1111/j.1467-8659.2009.01610.x.

[67] Chi-Han Peng, Eugene Zhang, Yoshihiro Kobayashi, and Peter Wonka. Connectivity editing for quadrilateral meshes. *Association for Computing Machinery Transactions on Graphics*, 30(6):141, 2011. doi: 10.1145/2070781.2024175.

[68] Robin Oval, Matthias Rippmann, Romain Mesnil, Tom Van Mele, Olivier Baverel, and Philippe Block. Topology finding of structural patterns. In Lars Hesselgren, Axel Kilian, Olga Sorkine-Hornung, Samar Malek, Karl-Gunnar Olsson, and Christopher J. K. Williams, editors, *Advances in Architectural Geometry 2018*. Klein Publishing GmbH (Ltd.), 2018. ISBN 978-3-903015-13-5.

[69] Kenshi Takayama, Daniele Panozzo, and Olga Sorkine-Hornung. Pattern-based quadrangulation for N-sided patches. In *Proceedings of the Symposium on Geometry Processing 2014*, pages 177–184. Eurographics Association, 2014. doi: 10.1111/cgf.12443.

[70] Chaman Singh Verma and Krishnan Suresh. A robust combinatorial approach to reduce singularities in quadrilateral meshes. *Computer-Aided Design*, 85:99–110, 2017. doi: 10.1016/j.proeng.2015.10.137.