Computer-Aided Design 137 (2021) 103030

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

# Two-Colour Topology Finding of Quad-Mesh Patterns

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

[a] Laboratoire Navier, UMR 8205, École des Ponts ParisTech, IFSTTAR, CNRS, UPE, Champs-sur-Marne, France
[b] ETH Zurich, Institute of Technology in Architecture, Block Research Group, Zurich, Switzerland
[c] University of Cambridge, Department of Engineering, Structures Research Group, Cambridge, United Kingdom

ARTICLE INFO

*Article history:*
Received 6 July 2020
Received in revised form 30 December 2020
Accepted 18 March 2021

*Keywords:*
Structural design
Topological design
Rule-based design
Grammar
Singularities
Colourability

ABSTRACT

The patterns of many structural systems must fulfil a property of two-colourability to partition their elements into two groups. Such examples include top versus bottom layers of continuous beams in elastic gridshells, corrugated versus non-corrugated directions in corrugated shells or warp versus weft threads in woven structures. Complying with such constraints does not depend on the geometry but on the topology of the structure, and, more specifically, on its singularities. This paper presents a search strategy to obtain patterns that fulfil this topological requirement, which represent only a fraction of the general design space. Based on an algebra for the exploration of the topology of quad meshes, including a grammar and a distance, a topology-finding algorithm is proposed to find the closest two-colour quad-mesh patterns from an input quad-mesh pattern. This approach is expressed as the projection to the two-colourable subspace of the design space. The distance underlying the definition of the projection measures the similarity between designs as the minimum number of topological grammar rules to apply to modify one design into another. A design application illustrates how two-colour topology finding can complement workflows for the exploration of structural patterns with singularities informed by the system's topological requirements.



## 1. Introduction

### 1.1. Context

Many structural systems and patterns rely on a bi-partition of their elements into two groups where elements of the same groups are not connected. This partition property is further called *two-colouring*. Two-colouring relates to colouring the elements using only two colours without having adjacent elements of the same colour. Fig. 1 illustrates this property for different systems with partitions in red and blue of nodes, panels or beams.

A necessary condition for a nexorade to have an alternation of nodes turning right and left is the two-colouring of its nodes (Fig. 1(a)). This alternation provides uniformity and preserves symmetry, as opposed to a design with non-alternated nodes [1]. A necessary condition for origami to be flat-foldable is the two-colouring of its faces in a checkerboard pattern, as demonstrated by Maekawa's theorem [2] (Fig. 1(b)). This property guarantees the alternation of panels facing upwards or downwards when folded. A necessary condition for an elastic gridshell to be made of two independent sets of continuous beams forming the top and bottom layers is the two-colouring of its beams (Fig. 1(c)). This partition avoids having continuous beams weaving between the two layers, to ease the planar layout process and avoid inducing additional bending pre-stresses due to the lifting process [3].

This partition also occurs at the scale of the structure of orthotropic materials, for directions with different properties, as in wood, composite or textile materials.

These two-colour requirements and properties are purely topological, independent from the geometry of the pattern.

### 1.2. Literature review

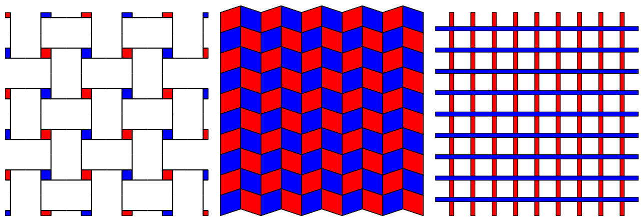
Table 1 provides a literature review of the structural systems for which pattern elements must respect a two-colouring property to provide a binary partition between two states.

Different types of two-colouring exist for patterns based on quad meshes, which do not entail the same topological requirements. Quad meshes are meshes with quads, meaning quadrilateral faces, only. Regular vertices in quad meshes have a valency of four, meaning that they are adjacent to four other vertices, or three if they are on the boundary. Singular vertices, or singularities, are vertices that do not fulfil these rules, as they are adjacent to a different number of vertices. These singularities control the fulfilment of two-colouring properties.

Map colouring inspires the term *colouring*, coming from graph theory [25]. The chromatic number of a graph is the minimum

* Corresponding author at: University of Cambridge, Department of Engineering, Structures Research Group, Cambridge, United Kingdom.
*E-mail address:* rpho2@cam.ac.uk (R. Oval).

https://doi.org/10.1016/j.cad.2021.103030

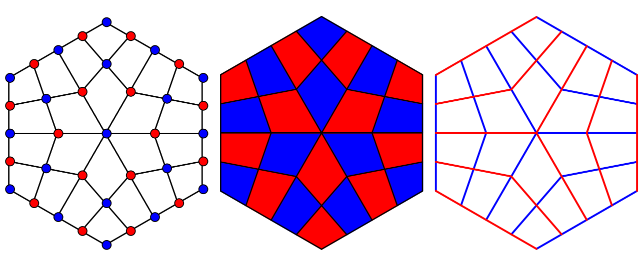


(a) Alternating nodes (b) Folding panels (c) Crossing beams

**Fig. 1.** Two-colour organisation of elements in patterns based on the partition into two groups, in red and blue, without having elements of the same group connected to each other. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

**Table 1**
Review of structural systems based on patterns with two-colour elements.

| Structural system | Pattern element | Binary states | |
|---|---|---|---|
| Reciprocal structures: nexorades, interlocking structures [1,4–6] | Vertex | Rightwards or leftwards turn | |
| Origami and folded shells [2,7–10] | Face | Upward or downward facing | |
| Elastic gridshells [11–14] | Polyedge | Top or bottom layer | |
| Woven structures [15–17] | Polyedge | Warp or weft direction | |
| Membranes | Polyedge | Strip width or length | – |
| Corrugated shells [18,19] | Polyedge | Corrugation direction | |
| Developable envelopes [20,21] | Polyedge | Ruling direction | |
| Circular and cyclidic meshes [22–24] | Face | X or Y axis | |

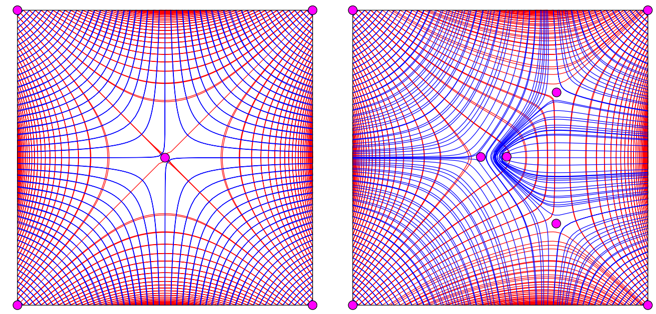


(a) Vertices (b) Faces (c) Polyedges

**Fig. 2.** The three types of two-colourability for quad-mesh patterns.

number of colours that can be used to colour all the nodes while no pairs of adjacent nodes have the same colour. In Fig. 2(a), the vertices of the mesh can be two-coloured as only two colours are necessary. Colouring is a labelling operation, where the colours can be replaced by the relevant binary states to encode specific data, like 'up' versus 'down' or 'left' versus 'right'. Similarly, face colouring relates to colouring all faces while no adjacent faces have the same colour, as in Fig. 2(b). Only face adjacency over the edges, not the vertices, counts. Polyedge colouring relates to colouring all polyedges while no crossing polyedges have the same colour, as in Fig. 2(c). Connected extremities of polyedges do not count as crossings.

Polyedges refer specifically to quad-mesh polyedges. Quad-mesh polyedges connect edges that are topologically opposite to each other across the regular four-valent vertices. They stop at singularities, at boundaries or when forming a closed loop.



(a) Symmetric loading (b) Asymmetric loading

**Fig. 3.** Patterns stemming from the principal stress directions for different loading conditions on a plate supported on its four corners. Integration of cross fields yields two-colour patterns by definition. Each cross field direction corresponds to one of the two groups of elements, in red or blue. But this design method has some limitations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Quad mesh vertex and face colouring do not correspond to the same problem on two dual meshes, despite the dual relation between vertices and faces. Indeed, the dual mesh of a quad mesh is, a priori, not a quad mesh, as singular vertices become singular faces, which are not quads.

Not any quad mesh guarantees these two-colouring properties and therefore is suitable for application to the systems in Table 1.

Quad-mesh patterns resulting from the integration of cross fields such as the principal curvature directions [20,21,26] and the principal stress directions [27] fulfil the polyedge two-colouring property by definition. Indeed, two groups of non-overlapping elements partition the resulting polyedges. The groups correspond to the two directions in the cross-field, as in Fig. 3 for two patterns stemming from the principal stress directions for different loading conditions on a plate supported on its four corners. Note that the singularities in pink have even valencies, of two, six or eight, for those off the boundary. Nevertheless, designing two-colour patterns relying on a cross-field requires information regarding the geometry or the statics system, for instance, and is hard to combine with other design strategies.

Caigui et al. [28] generate two-colour checkerboard patterns following the diagonals of quad meshes. This approach is equivalent to applying the ambo Conway operator [29] on any mesh, producing two families of faces: one from the initial vertices and one from the initial faces. However, the resulting patterns are not quad meshes due to singular vertices or faces in the initial quad mesh, which translate into singular faces in the final mesh, breaking the quad-mesh constraint, which is necessary for some structural patterns.

### *1.3. Problem statement*

Not all quad meshes can be two-coloured, and actually, most quad meshes cannot, as seen farther in this paper. The three-valent singularities in pink in Fig. 4 do not allow this alternation between two groups of strip elements, in red and blue. Therefore, a third group, in green, is necessary to avoid having elements of the same group overlapping each other. This topological design does not allow the partition of continuous beams into two layers, as suitable for elastic gridshells to prevent having beams weaving between the two layers, for instance.

Most generation algorithms, with the noteworthy exception of the specific cross-field integration scheme, do not handle this requirement. Designers need exploration algorithms to address the

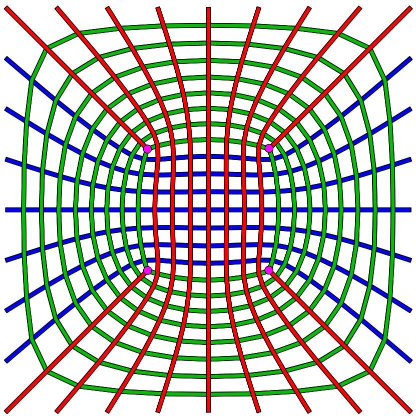


**Fig. 4.** A quad-mesh pattern with three-valent singularities, in pink, is an example of a topology that cannot be two-coloured in red and blue. A third group of strip elements, in green, is necessary to avoid having elements of the same group overlapping each other. This topology is not suitable for the design of an elastic gridshell made of continuous beams organised in two layers without having beams weaving between the two layers. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

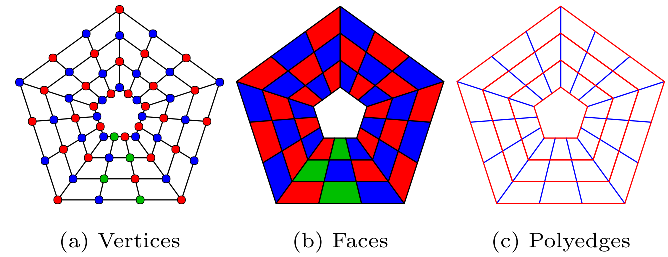



**Fig. 5.** Quad mesh with two-colour polyedges but without two-colour vertices nor faces due to the odd number of elements along the closed polyedges.

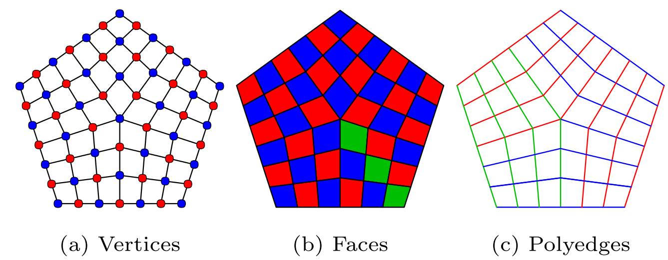



**Fig. 6.** Quad mesh with two-colour vertices but without two-colour faces nor polyedges due to the odd number of elements around the singularity.

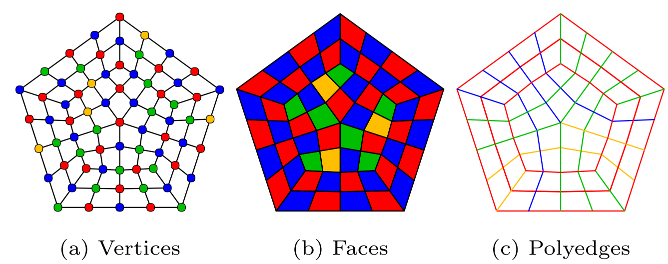



**Fig. 7.** Quad mesh without two-colour vertices, faces nor polyedges due to the odd number of elements along the closed polyedges and around the singularities.

specific problem of two-colouring of quad meshes. Such methods can be used as a post-refinement step to the generation of structured quad meshes with a low number of singularities already placed as to preserve geometrical qualities and features, completing the toolbox of mesh design [30–33].

### 1.4. Research objectives

For the aforementioned structural systems, design strategies and search algorithms are required to modify quad meshes into two-colour quad meshes. The input quad mesh should inform the generation of the output two-colour quad mesh by sharing some degree of similarity with it, to preserve the initial design intent. This search relates to finding the projection of a quad mesh onto the subspace of two-colour quad meshes. This similarity must be defined and this projection developed. The designs found should be the best in the sense of topological similarity, which does not directly relate to a geometry-dependent performance. Moreover, the most similar topology is not necessarily unique if several designs are found at the same distance. Such designs can then be sorted and selected based on other application-specific metrics, related to topology or not.

In structural design, a variety of exploration means allow the designer, architect or engineer, to interactively explore design spaces constrained to structurally sound or fabrication affordable designs. Correspondingly, to provide exploration freedom to the designer, the projection algorithm should be extended to provide not only the most similar design but a *set* of most similar designs that offer different design directions.

### 1.5. Contributions

This paper introduces *two-colour topology finding* for the design of singularities in quad-mesh patterns using a projection search. Section 2 details the problem parameterisation of the two-colouring of the different elements in quad meshes. The challenge of singularity requirement is identified and an index-based and a graph-based characterisations are presented. Section 3 presents a grammar for topological exploration of quad-mesh singularities inducing the definition of a topological distance based on this grammar. Section 4 develops the projection algorithm to find the most similar two-colour quad meshes from a quad mesh, according to the grammar-based distance. The projection is extended to provide two-colour quad meshes that are less similar but that offer other design directions. Section 5 tests this algorithm on validation examples providing insightful numerical values and applies it on a design case for two-colour topology finding of patterns for an elastic gridshell.

This research is implemented in compas_singular [34] as a Python package of COMPAS [35], an open-source Python-based computational framework for collaboration and research in architecture, engineering and digital fabrication.

## 2. Two-colour problem parameterisation

For the search of two-colour quad meshes, identification and characterisation of the requirements of the three different types of two-colouring are necessary.

A quad mesh can respect all, some or none of the three types of element two-colouring: vertices, faces and polyedges.

The quad mesh in Fig. 5 has two-colour polyedges. Nevertheless, its vertices and faces cannot be two-coloured. Indeed, the closed polyedges have an odd subdivision, which does not allow this binary alternation of the vertices and faces along it.

The quad mesh in Fig. 6 has two-colour vertices. Nevertheless, its faces and polyedges cannot be two-coloured. Indeed, the singularity with an odd valency, equal to five, does not allow this binary alternation of the five faces and polyedges around it.

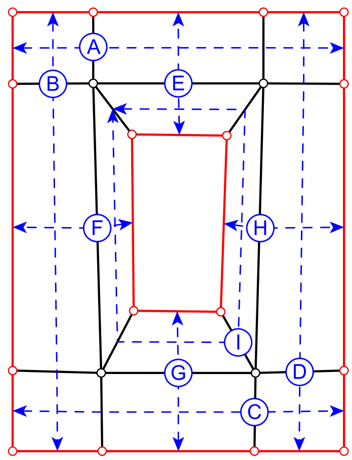



**Fig. 8.** The strip data in a quad mesh as lists of opposite edges across the quad faces.

**Table 2**
Dependencies between two-colouring types and topological aspects in quad meshes.

| Two-colourability type | Vertices | Faces | Polyedges |
|---|---|---|---|
| Singularity requirement | No | Yes | Yes |
| Density requirement | Yes | Yes | No |

The quad mesh in Fig. 7 mixes odd-subdivided polyedges and odd-valency singularities. Therefore, its vertices, faces and polyedges cannot be two-coloured.

The different types of two-colouring depend on different topological aspects of a quad mesh: its singularities and its density. Table 2 summarises these dependencies.

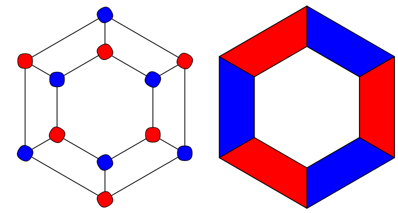
The two requirements for two-colouring are treated separately using a coarse quad mesh and by considering its topological face strips. On the one hand, the density requirement depends on the density of the strips. On the other hand, the singularity requirement depends on the combination of the strips. The following section details the strip structure in quad meshes.

### 2.1. Strip structure

The strips of quad faces provide a suitable structure in quad meshes to perform exploration without breaking the quad constraint. Indeed, quad meshes contain a structure of strips of quad faces, which constitute a topological description of quad meshes at a larger scale than quad faces. Quad mesh strips depend on the topology of a quad mesh, not its geometry. Strips are constructed based on the relationship between pairs of opposite edges across quad faces. The strip data is collected as a list of edges facing each other across the adjacent quad faces, as shown in blue in Fig. 8 for a nine-strip coarse quad mesh in black and its boundary highlighted in red. A strip is open when its extremity edges are on the boundary, as strips A to H, or closed if it forms a loop, as strip I.

These strips also correspond to the independent parameters for densification of a quad mesh [33]. Strips also apply in different ways for digital [36] or physical [37] modelling approaches, also referred to as loops, rings or chords.

Strips can also be used to describe and parameterise the two requirements for quad-mesh two-colouring.



(a) Vertices (b) Faces

**Fig. 10.** The vertex and face elements along a closed strip are two-coloured if subdivided by an even number of elements.

### 2.2. Density requirement

Vertex and face two-colouring depend on density, on the contrary to polyedge two-colouring.

For an open strip as in Fig. 9, any density subdivision can be chosen for the open strip while having vertex and face two-colouring. Therefore, if the quad mesh only has open strips, vertex and face two-colouring applies.

However, a requirement applies regarding closed strips. In Fig. 10, the closed strip is subdivided by six elements so its vertices and faces are two-coloured. However, in Fig. 11, the closed strip is subdivided by five elements and does not fulfil the constraint. Therefore, the density of the crossing strips is corrected to an even number to provide two-colour vertices and faces.

The even number of subdivisions is necessary and sufficient for each closed strip of the quad mesh. This requirement sets a constraint on the sum of the density parameters $d_j$ of the strips $j$ crossing the closed strip $i$ to be even. A strip counts each time it crosses the closed strip, as there can be multiple crossings $k_{ij}$ between strips $i$ and $j$. This constraint formalises as:

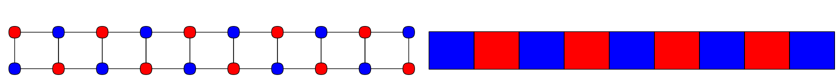
$$\forall i \in S_c, \sum_{j \in S_i^{\perp}} k_{ij} d_j \propto 2, \quad (1)$$

where $S_c$ is the set of closed strips and $S_i^{\perp}$ the set of strips crossing the strip $i$.

Characterisation and fulfilment of the density requirement are more direct than for the singularity requirement, as shown in the following section. Therefore and according to Table 2, fulfilling face two-colouring, which depends both on the densities and singularities, and polyedge two-colouring, which depends only on the singularities, are more complex and become the focus of the next investigations.

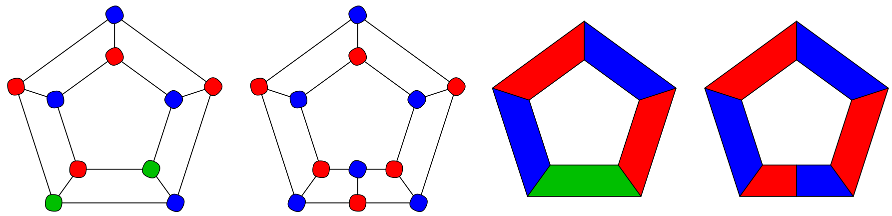
### 2.3. Singularity requirement

Face and polyedge two-colouring depend on the singularities, on the contrary to vertex two-colouring. Indeed, the number of faces and polyedges looping around a singularity in a quad mesh is equal to its valency $n$ and is not necessarily even. On the



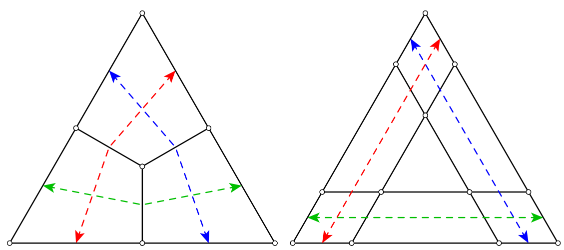
(a) Vertices (b) Faces

**Fig. 9.** The vertex and face elements along an open strip are always two-coloured.



(a) Vertices (b) Faces

**Fig. 11.** The vertex and face elements along a closed strip are not two-coloured if subdivided by an odd number of elements. This number of elements can be tuned into an even number by modifying the density of the crossing strips.



(a) Around a non-boundary vertex (b) Along a boundary

**Fig. 12.** Non-two-colourable coarse quad meshes because of odd numbers of strips.

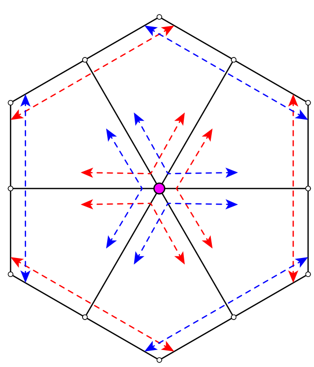


**Fig. 13.** The even numbers of strips around the non-boundary vertex in pink and along the boundary guarantee the fulfilment of the singularity requirement for two-colouring. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

contrary, the number of vertices is equal to twice its valency $2n$ and is therefore even. The number of strips is equivalent to the number of faces and polyedges.

In Fig. 12, three colours are necessary to colour the three strips around the inner vertex in Fig. 12(a) and along the boundaries in Fig. 12(b) without crossings of strips of the same colour. Therefore, these topologies do not fulfil the singularity requirement for face and polyedge two-colouring, as two-colour alternation is not possible.

However, the topology in Fig. 13 fulfils face and polyedge two-colouring thanks to the even number of strips: six strips around the non-boundary vertex in pink and six strips along the boundary.

A quad mesh fulfilling the singularity requirement for two-colouring is equivalent to fulfilling the following condition: the number of strips around each non-boundary vertex and along each boundary in the coarse quad mesh are even. The requirement on the boundaries depends on all the singularities along the boundary. The evenness requirement is equivalent to strip two-colouring: strips should be coloured using only two colours without having crossings between strips of the same colour.

For further dealing with the singularity requirement, a coarse mesh is used in parallel, which reduces the density while preserving the singularity structure, for a more efficient computation. Fig. 14 illustrates how such a coarse mesh is obtained from an input dense mesh: the polyedges that form the boundaries and the ones that stem from the singularities are extracted to obtain the connectivity of the coarse mesh. The relation between the parent elements in the coarse mesh with the child elements in the dense mesh is stored to find the corresponding modifications to apply. As such, the exploration of the set of singularities is conceptually and computationally independent of density thanks to the coarse mesh.

Two complementary approaches characterise the singularity requirement for two-colouring, with different pros and cons for the search of two-colour quad-mesh patterns.

*2.3.1. Index-based characterisation*

The two conditions on the even number of strips around the inner vertices and along the boundaries can be translated from the number of strips to the indices of the vertices. The index of q vertex describes its irregularity in the mesh. A regular vertex has an index of 0 and a singular vertex a non-null index, positive or negative. The index $i_v$ of a vertex $v$ in a quad mesh is expressed as:

$$i_v = \frac{n_0 - n_v}{4}, \tag{2}$$

with $n_v$ the actual valency of vertex $v$ and $n_0$ the regular valency, equal to 4, or 3 if the vertex is on the boundary. Using the index instead of the valency allows a more uniform expression of the characterisation of the singularity requirement.

An even number of strips around each vertex $v$ in the set of non-boundary vertices $V \setminus \partial V$ translates into their index $i_v$ being proportional to 1/2:

$$\forall v \in V \setminus \partial V, i_v \propto 1/2, \tag{3}$$

and an even number of strips along each boundary $\partial V_k$ translates into the sums of vertex indices $i_v$ along each boundary being proportional to 1/2 as well:

$$\forall \partial V_k \in \partial V, \sum_{v \in \partial V_k} i_v \propto 1/2. \tag{4}$$

This vertex-based characterisation allows directly assessing if a quad mesh fulfils the singularity requirement for two-colouring. A metric to minimise is derived to provide a measure on the degree of fulfilment of this requirement by integrating the two conditions in Eqs. (3) and (4):

$$\sum_{v \in V \setminus \partial V} (|i_v| \bmod 1/2) + \sum_{\partial V_k \in \partial V} (|\sum_{v \in \partial V_k} i_v| \bmod 1/2). \tag{5}$$

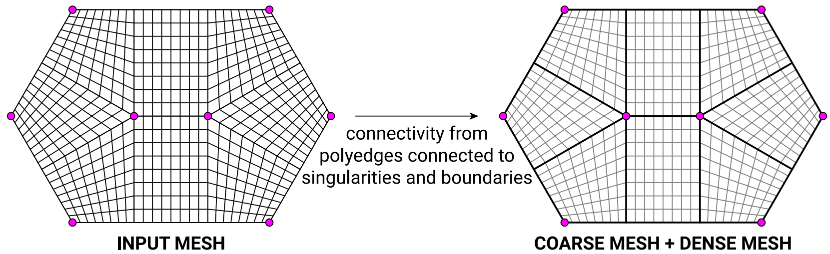



**Fig. 14.** From an input quad mesh, the connectivity between the singularities in pink is extracted to obtain a lighter coarse mesh in black and its relation with the underlying dense mesh in grey for more efficient computation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

This characterisation does not provide direct information on how to fulfil the singularity requirement for two-colouring, even though this metric can serve for stochastic search.

Another characterisation, more suitable for deterministic search, is expressed on a graph encoding the strip data, presented in the following section.

### 2.3.2. Graph-based characterisation

The singularity requirement can be expressed via state-of-the-art graph colouring, on a graph encoding the connectivity of the quad-mesh strips.

*Strip graph.* A graph is introduced to extract and compress the necessary connectivity data of the quad-mesh strips. This strip connectivity consists of the crossings between the strips over the faces, as having a set of two-colour strips without strips of the same colour overlapping is equivalent to fulfilling the singularity requirement.

Each strip of the mesh converts into a node in the graph and each face of the mesh converts into an edge of the graph, connecting the nodes of the two corresponding strips that cross over the face. The strip graph is constructed as illustrated in Fig. 15:

1. collect the strips, in dashed lines, in the quad mesh (Fig. 15(a));
2. for each mesh strip $S$, a graph node $N$ is added at the centroid of the mesh strip: $S_{mesh} = N_{graph}$ (Fig. 15(b));
3. for each mesh face $F$, a graph edge $E$ is added connecting the two crossing strips, potentially the same strip in the case of self-crossings: $F_{mesh} = E_{graph}$ (Fig. 15(c)).

The quad mesh and the strip graph, as shown in Fig. 15(d), feature a duality, though different from the type of duality between two meshes.

The graph encodes some but not all of the topological data. However, this simplified graph encodes all the necessary data to assess and explore two-colouring of quad-mesh strips.

For efficiency, the graph is computed on the coarse singularity mesh of the initial mesh, meaning the mesh that has a minimal density while capturing the connectivity between the singularities. Fig. 16 shows the data management process of the presented algorithm:

1. from the dense input quad mesh $M$, the coarse singularity quad mesh $M^O$ is generated, which also stores the parent–child relations between the strips of the two meshes;
2. from the coarse mesh $M^O$ and its strip structure, the strip graph $G^0$ is computed;
3. the two-colouring algorithm, based on element deletion, is applied on the strip graph $G^0$ to obtain a two-colour strip graph $G^0_2$ (detailed later);
4. the strips in the coarse mesh $M^O$ that correspond to the nodes deleted from graph $G^0$ to graph $G^0_2$ are deleted to obtain a two-colour coarse quad mesh $M^O$; and
5. the multiple strips in the dense input mesh $M$ that correspond to the strips deleted to obtain the two-colour coarse mesh $M^O_2$ are deleted, knowing the parent–child relations, to obtain the two-colour mesh $M_2$. Finally, the elements of $M_2$ are coloured.

As such, the densities of the strips of the input and output dense meshes are the same but the evaluation of the topological attributes via the coarse meshes and the computation of the two-colour solutions via the strip graphs is done efficiently, independently of density.

*Graph colouring.* The singularity requirement can be expressed as colouring all the strips with only two colours without crossings between strips of the same colour. This problem is equivalent to node colouring of the strip graph. Indeed, each graph node encodes a strip, and each graph edge encodes a strip crossing over a face.

Node colouring is a classic problem of graph theory that aims at finding the chromatic number. The chromatic number of a graph is the minimum number of colours that can be used to colour all the nodes while no pairs of adjacent nodes have the same colour. The existing algorithms offer different benefits and drawbacks in terms of speed and robustness to solve this NP-hard problem [25,38]. However, determining the chromatic number is not necessary here. Assessing whether the graph is two-colourable or not is sufficient.

To do so, the two-colouring process starts by colouring one node. Then, the iterative process colours the nodes that are adjacent to the coloured nodes but not yet coloured. If an uncoloured node cannot be coloured because already adjacent to nodes of both colours, the process returns a False statement. If eventually all nodes are assigned a colour, the process returns a True statement. In Fig. 17(a), two-colouring is not possible because one node is adjacent to both colours, highlighted with dashed edges. A third colour is necessary. In Fig. 17(b), two-colouring is possible, after deleting one of the nodes of the previous graph.

The complexity of the algorithm equals the number of colour checks between adjacent nodes, i.e. the number of graph nodes multiplied by their valency. Let $N$ be the number of nodes in the graph (i.e. the number of strips in the mesh) and $E$ be the number of edges in the graph (i.e. the number of faces in the mesh). Let us consider a line graph, which presents a minimum connectivity for a two-colour graph. A line graph with $N$ nodes has $E = N - 1$ edges and needs $2E = 2(N - 1)$ checks between adjacent nodes. Therefore, the line graph has a complexity $O(N)$ and $O(E)$. Let us consider a complete bipartite graph, which presents a maximum connectivity for a two-colour graph. A complete bipartite graph with $N$ nodes has $E = (N/2)^2$ edges and needs $2E = N^2/2$ checks between adjacent nodes. Therefore,

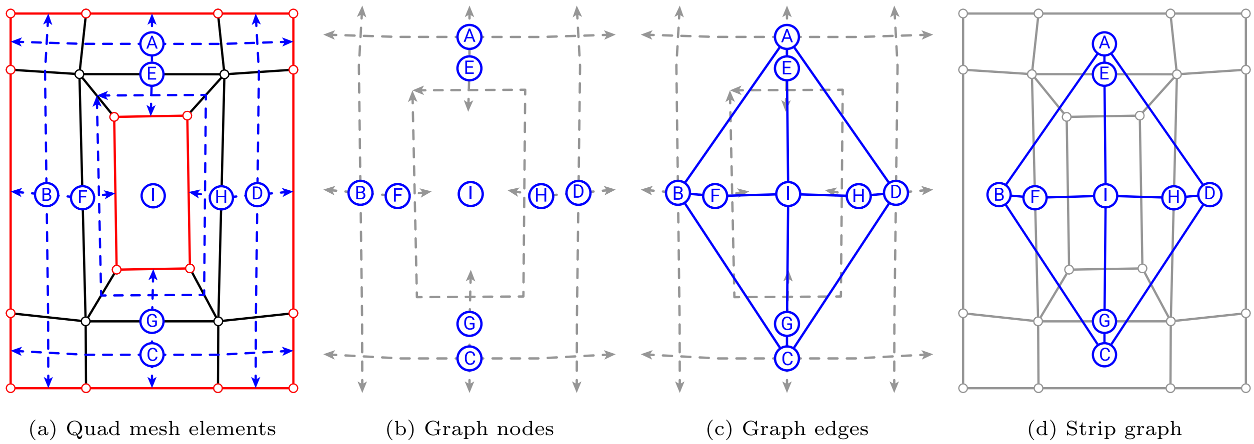



**Fig. 15.** Construction of the strip graph of a quad mesh, encapsulating the strip connectivity of the crossings over the faces. Each mesh strip, in dashed lines, becomes a graph node and each mesh face becomes a graph edge connecting the nodes of the crossing strips.

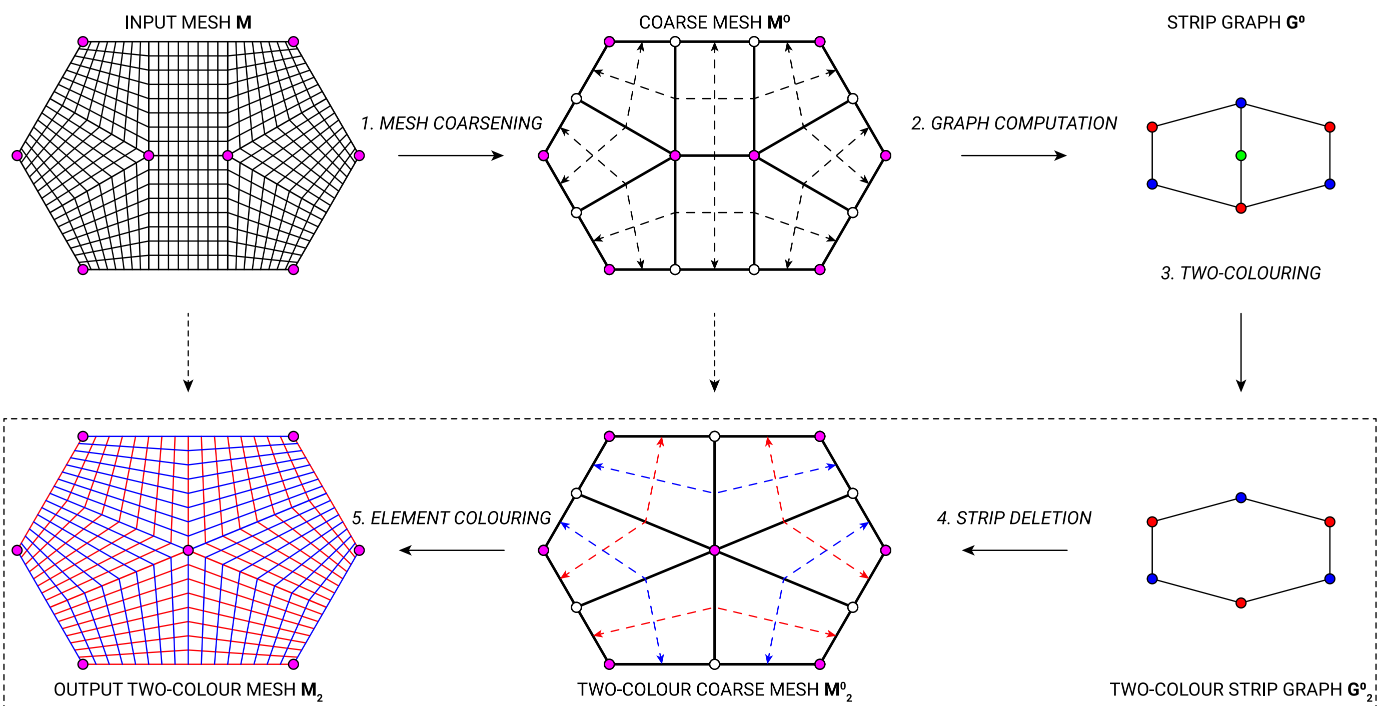



**Fig. 16.** An input quad mesh $M$ is modified into a two-colour mesh $M_2$ via the coarse mesh $M^0$ that provides the connectivity between the singularities and the strip graph $G^0$ that is directly modified to become a two-colour graph $G_2^0$. The deleted nodes between the graphs give the corresponding strips to delete in the two-colour coarse mesh $M_2^O$ and the parent–child relations from the coarse mesh $M^O$ give the corresponding strips to delete in $M$ to finally obtain $M_2$.

the complete bipartite graph has a complexity $O(N^2)$ and $O(E)$. This worst-case scenario shows that the algorithm has a quadratic worst-case complexity regarding the graph nodes $O(N^2)$ and a linear one regarding the graph edges $O(E)$. This depth-/breadth-first approach to check two-colourability in a linear or quadratic time is opposed to the various algorithms for colouring, which are NP-complete to check a k-colourability and NP-hard to find the minimum k-colourability, i.e. the chromatic number [25].

*Examples.* The following examples apply graph-based characterisation by building the strip graph and trying to colour it with two colours. The equivalence with index-based characterisation is provided.

In Fig. 18, the coarse quad mesh has two-colour strips, evidenced by its two-colour strip graph. Therefore the densified quad mesh fulfils the singularity requirement and has two-colour polyedges. This graph-based characterisation is in line with the

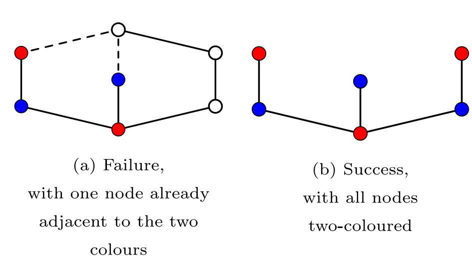
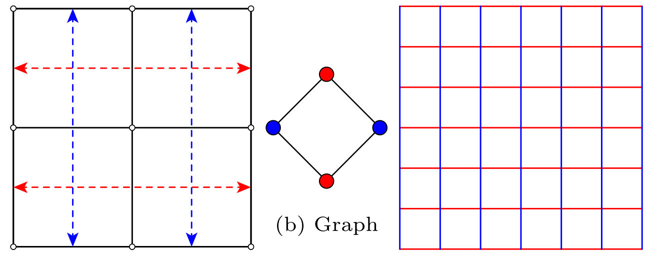


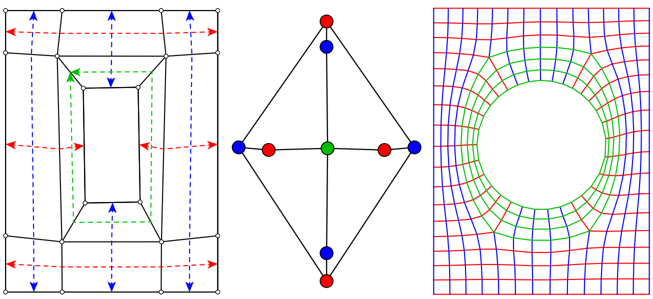

**Fig. 17.** Attempts at two-colouring the nodes of a graph.

index-based characterisation. Indeed, the regular four-valent inner vertex has an index of 0, which is proportional to 1/2, and the sum of the indices of the four regular three-valent and four



(a) Mesh (b) Graph (c) Polyedges

**Fig. 18.** Quad mesh with two-colour polyedges characterised by the two-colour strip graph of its coarse quad mesh.



(a) Mesh (b) Graph (c) Polyedges

**Fig. 19.** Quad mesh without two-colour polyedges evidenced by the three-coloured strip graph of its coarse quad mesh.

irregular two-valent vertices along the boundary equals 1, which is proportional to 1/2. Therefore the singularity requirement for two-colouring is fulfilled.

In Fig. 19 on the contrary, the coarse quad mesh does not fulfil this requirement, as shown by its three-coloured strip graph. For representation, full colouring is completed using the greedy Welsh–Powell algorithm [39]. This graph-based characterisation is in line with the index-based characterisation. Indeed, the irregular five-valent inner vertices have an index of −1/4, which is not proportional to 1/2. Therefore the singularity requirement for two-colouring is not fulfilled and the polyedges in the densified mesh cannot be two-coloured.

This graph-based characterisation of the singularity requirement for two-colouring of quad meshes using its strip structure provides a means for assessment. Combined with a suitable grammar for the topological exploration of quad-mesh strips, this approach can develop into an interactive search algorithm.

## 3. Quad-mesh topological exploration

Based on the strip structure, suitable for two-colouring characterisation, a grammar that modifies the quad-mesh strips, and consequently the singularities, is detailed. Then, a distance based on this strip grammar is defined to evaluate the topological similarity between two quad meshes, before developing the two-colour projection in the next section.

### 3.1. Topological grammar

A quad-mesh grammar with two low-level rules for *addition* and *deletion* of strips is detailed [33]. This grammar is purely topological, independent from geometry. Fig. 20 illustrates the two reciprocal rules for different configurations of strips with the insertion of a strip along a polyedge for the addition rule and the collapse of a strip into a polyedge for the deletion rule, with the strip and polyedge elements highlighted in blue. This grammar allows non-linear exploration as any rule can be undone by its reciprocal rule.

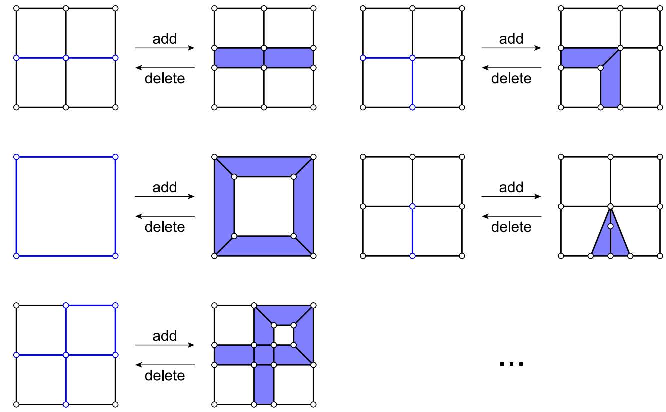



**Fig. 20.** Quad-mesh grammar of low-level rules for addition and deletion of strips in blue with different configurations.

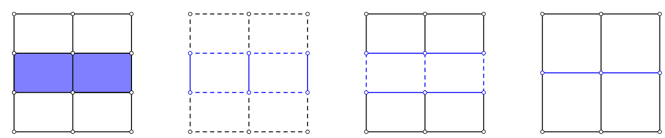


**Fig. 21.** Steps for the deletion of a strip, collapsing into a polyedge.

The addition and deletion of strips have corresponding operations on the strip graph. Deleting a mesh strip with its faces induces deletion of the graph node with its edges. Adding a mesh strip along a polyedge induces addition of a graph node connected to the graph nodes of the strips crossing the polyedge.

To make a graph two-coloured, nodes must be removed. Only adding nodes cannot solve this problem. Therefore, the search method is based on the exploration of the combination of strip deletions. Only the strip deletion rule is detailed.

#### 3.1.1. Strip deletion

To delete a strip by collapsing it into a polyedge, the following operations are sequentially applied on the strip, as illustrated in Fig. 21:

1. get the edges of the strip to delete;
2. group the vertices per connecting edges;
3. delete the faces of the strip;
4. merge the grouped vertices into a new vertex.

Visually, the strip edges are collapsed to zero-length edges, resulting in the collapse of the strip faces.

#### 3.1.2. Boundary collapse

Deleting a strip can cause the collapse of a boundary if less than three edges represent the boundary after deletion of the strip edges. Strip refinement prevents this boundary collapse, without changing the singularities, as shown in Fig. 22. To avoid the collapse of the inner boundary due to the deletion of the strip in blue, the remaining strips that end at this boundary are refined. Before deleting a strip, a verification predicts boundary collapse if:

$$|E_{strip} \cap E_{boundary}| < 3, \tag{6}$$

where $E_{strip}$ is the set of edges of the strip and $E_{boundary}$ is the set of edges of the boundary.

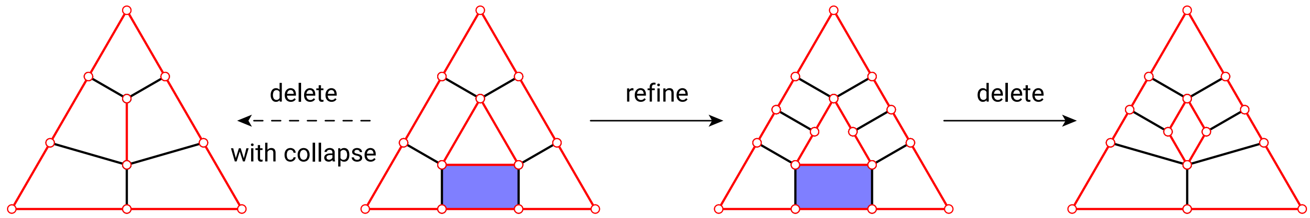



**Fig. 22.** When deleting some strips, in blue, refining other strips avoids boundaries to collapse to less than three edges and close a boundary. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Using this grammar to explore the design space of quad meshes, a topological similarity can be expressed to define a distance underlying the projection algorithm.

### *3.2. Grammar-based topological distance*

A distance – or metric – measures the similarity between two objects of the same design space. Here, the introduced distance measures the topological difference between quad meshes based on the previous grammar rules.

Metrics in computer science and information theory are also ruled-based, like the Hamming distance [40], which measures the similarity between two strings as the necessary number of substitutions to obtain one from the other, or the Levenshtein distance [41], which resorts to substitutions, insertions and deletions. A key requirement for any distance is to respect the triangle inequality, which states that the distance between two objects is equal or shorter than the sum of the distances between these two objects and any third object.

The quad-mesh topological distance is defined as the minimal number of strips to add and delete to go from one quad mesh to another. The minimality condition is necessary to exclude reciprocal addition and deletion of the same strip.

The distance $d$ applies to two elements of the space $E$ of all quad meshes, with the same shape topology, meaning the same number of handles and boundaries, and yields a positive integer in $\mathbb{N}^+$:

$$d : E \rightarrow \mathbb{N}^+. \tag{7}$$

The three distance properties are verified:

- the distance is symmetric. The same minimal number of rules apply from a topology $A$ to a topology $B$ and from $B$ to $A$, as for each addition or deletion strip rule there is a reciprocal one:

$$\forall (A, B) \in E^2, d(A, B) = d(B, A); \tag{8}$$

- the distance from a topology to itself is null because no rules need to be applied and if no rules need to be applied then two topologies are the same:

$$\forall (A, B) \in E^2, d(A, B) = 0 \iff A = B; \tag{9}$$

- the triangle inequality is satisfied: the same or a lower number of rules have to be applied to go directly from a topology $A$ to a topology $C$ than going through an intermediary topology $B$:

$$\forall (A, B, C) \in E^3, d(A, C) \leq d(A, B) + d(B, C). \tag{10}$$

  Equality occurs if no rules between $A$ and $B$ and between $B$ and $C$ are reciprocal and can compensate each other.

Successive application of strip rules generally increases by one the distance and therefore decreases by one the similarity with the initial design. Only the application of a reciprocal rule cancelling a previous rule reduces by one the distance.

The number of deletion rules is not necessarily the number of deleted strips. Deleting some strips can cause collateral deletion of other strips whose faces are all included in the deleted strips. Therefore, the application of a deletion rule counts as the application of multiple deletion rules, equal to one plus the number of collateral deletions.

Computing the distance between any two quad meshes is not necessary for the projection algorithm. Only keeping count of the number of deleted strips is sufficient and equivalent to the topological distance, to find the most similar quad meshes.

## 4. Projection search algorithm

To perform two-colour topology finding, a search algorithm is developed using the strip grammar rules. Starting with a quad mesh that cannot be two-coloured, the most similar two-colour quad meshes are found, according to the topological distance. The output mesh is not necessarily unique when different two-colour topologies are found at the same distance and the best design can be selected based on various metrics specific to the design application. This search for the closest two-colour quad mesh is a projection onto the two-colour design subspace. Moreover, the projection is extended to not only yield the closest design but the closest ones in several directions that provide different independent options to the designer. The presented algorithm does not take addition rules into account as it uses only deletion rules, simplifying the combinatorial search while still providing the closest design. The resulting pool of solutions at greater distances, i.e. less similar, is therefore reduced.

### *4.1. Search approach*

The application of strip rules drives this search for two-colour quad meshes, informed by the colouring of the strip graph.

#### *4.1.1. Third-colour deletion*

The design in Fig. 18 fulfils the two-colour requirement. However, the design in Fig. 19 does not. The graph cannot be two-coloured, but completing the colouring process with a general colouring algorithm like the Welsh–Powell algorithm [39] informs on a solution to make this design two-coloured.

The graph has one node of the third colour green. Deleting the corresponding third-colour strip yields the design in Fig. 23 that can be two-coloured, evidenced by the strip graph with one node less due to the strip deletion. Deleting the strip moved the five-valent non-boundary singularities to the boundary, along which the sum of the vertex indices is proportional to 1/2, in line with the vertex-base characterisation.

However, multiple combinations exist to colour a graph with three colours or more. Fig. 24 hints at this richness that goes beyond the unique option yielded by a colouring algorithm. This selection of nine coloured graphs suggests different sets of third-colour nodes. They each correspond to a different set of strips to delete to obtain a two-colour quad mesh.

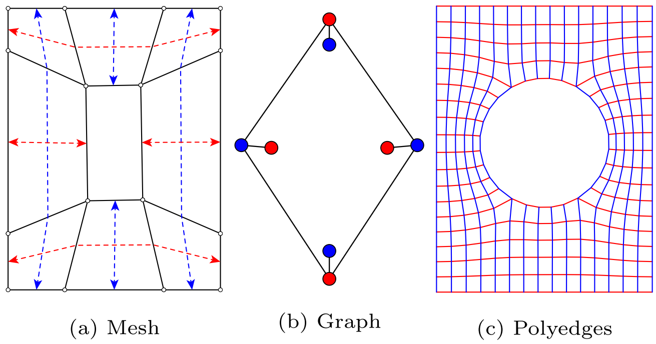



**Fig. 23.** Quad mesh with two-colour polyedges after deletion of the third-colour strips, evidenced by the new two-colour strip graph of its coarse quad mesh.

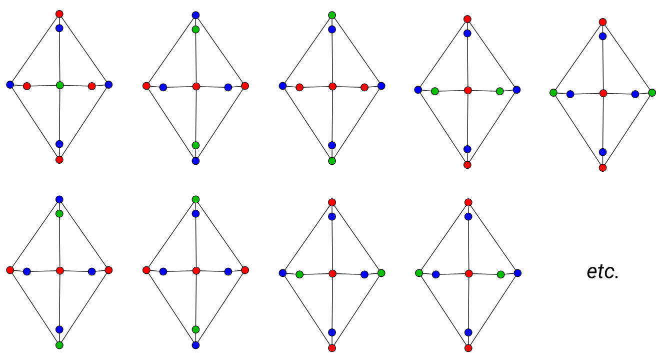



**Fig. 24.** Some of the combinatorial richness of third-colour nodes, in green, to delete to obtain a two-colour graph. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

This combinatorial richness opens up an opportunity to yield multiple quad meshes of the two-colour subspace. The search algorithm proposes a strategy to sort out the most similar quad meshes that provide different design directions. This approach relates to a projection to the two-colour subspace, based on the previous topological distance.

#### *4.1.2. Unidirectional projection*

The search algorithm only applies deletion rules. The reciprocal addition rule is excluded, which does not allow a bidirectional search. The search is therefore oriented by the successive application of deletion rules, defining the search direction.

Fig. 25 shows the directed graph for the search through the $2^3 = 8$ combinations for the deletions of three strips.

The empty combination is upstream to all the combinations and the complete combination (1,2,3) is downstream to all the combinations. Considering a specific combination like (1), one part of the graph is upstream, namely, the combination (-) and one part of the graph is downstream, namely the combinations (1,2), (1,3) and (1,2,3). Any other combination lies in a different direction of the graph compared to combination (1). The search algorithm limited to deletion rules cannot obtain the mesh corresponding to combination (2) from the mesh corresponding to combination (1). Nevertheless, the search can obtain the mesh corresponding to combination (1,2) from the mesh corresponding to combination (1) by deleting strip 2. Indeed, combination (1,2) is part of the downstream direction from combination (1).

Finding the closest or most similar two-colour design is a projection to the design subspace of two-colour designs. This projection $P$ applies to one element of the space $E$, of all quad

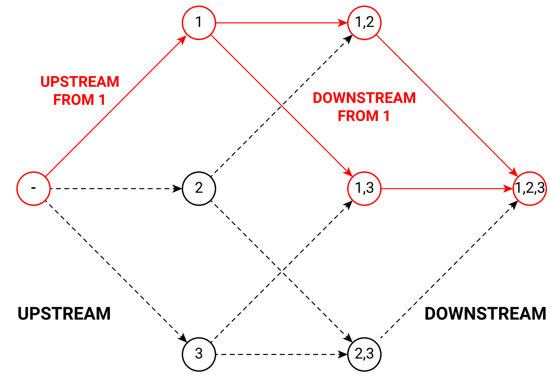



**Fig. 25.** This graph represents the search organisation through the combinations of three deletion rules 1, 2 and 3. The search algorithm applies only deletion rules, defining an orientation from the upstream combination (-) to the downstream combination (1,2,3).

meshes, with the same shape topology, meaning the same number of handles and boundaries, and yields another one, or several ones if there is equidistance to several two-colourable designs:

$$\begin{aligned} P :& E \to E, \\ & M \mapsto M_2 \end{aligned} \tag{11}$$

where $M$ is a quad mesh and $M_2$ the most similar two-colour quad mesh, obtained from the projection on the subspace of two-colour quad meshes.

If a quad mesh can already be two-coloured, applying two-colour projection yields the same quad mesh. Therefore, the idempotence definition of a projection is respected:

$$P^2 = P. \tag{12}$$

This property justifies the use of the term projection and provides a visual means of understanding this algorithm: a geometrical equivalent is to consider the space of quad meshes as a N-D space and the space of two-colour quad meshes as a projected subspace.

#### *4.1.3. Multi-direction projection*

To find the closest designs first, the number of strip deletions applies in increasing order. All combinations of strip deletions are tested starting with one strip, then two strips, until the maximum number of strips.

A seven-strip quad mesh that is not two-coloured serves as example in Figs. 26 to 28. The figures show completely coloured strip graphs for better understanding, though not needed for computing.

In Fig. 26, some strip deletions yield two-colour quad meshes, as deleting strips 1 and 2, but some do not, as deleting strip 3. Here, the resulting two-colour quad meshes are at a distance of one from the input quad mesh.

The application of more deletion rules on a two-colour quad mesh yields another two-colour quad mesh. In Fig. 27, deleting strips 1 yields two-colour quad mesh at a distance of one, therefore deleting strips 1 and 3 as well but at a distance of two. Such a combination of rules is redundant because it yields another two-colour design in the same direction but at a greater distance in the two-colour subspace.

Nevertheless, combining multiple deletion rules allows finding two-colour quad meshes at greater distances but in different directions. In Fig. 28, deleting strips 3 and 6 separately does not yield two-colour quad meshes but combining them yields a two-colour quad mesh at a distance of two. Deleting strip 1, for instance, provides a two-colour quad mesh at a distance of

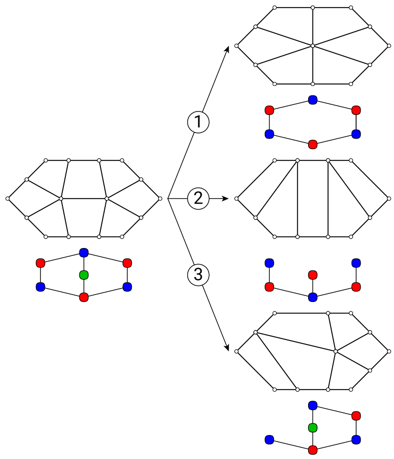



**Fig. 26.** Deleting one strip in a quad mesh that is not two-coloured may yield two-colour quad meshes or not.

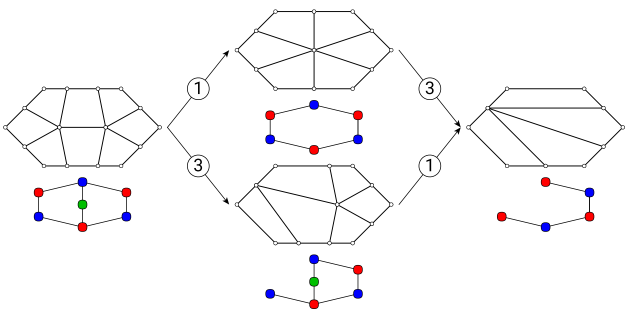



**Fig. 27.** Deleting a second strip of a two-colour quad mesh is redundant because it yields another two-colour quad mesh, but at a greater distance and in the same direction.

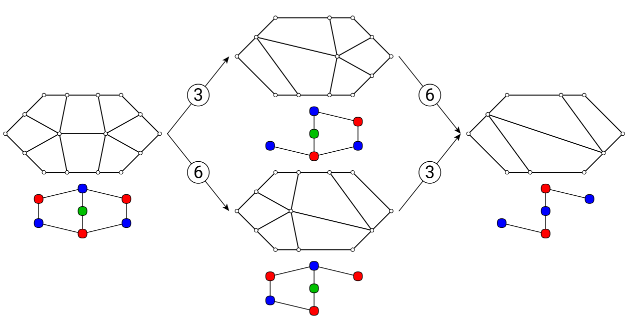



**Fig. 28.** Deleting two strips separately may not yield two-colour quad meshes, but can yield a two-colour quad mesh at a greater distance in a new direction.

one, but in another direction. Hence, the interest in such farther two-colour quad meshes to provide an alternative to the designer.

Fig. 29 illustrates the principle of the algorithm for finding the closest designs in the different design directions in the search through the $2^3 = 8$ combinations for the deletion of three strips The result from strip deletions in pink are two-coloured, and the ones in black are not. Finding a two-colour design removes all of the downstream designs in the same direction from the search algorithm, as highlighted by dashed circles and lines. The output is the closest two-colour design in their respective directions, represented by full pink circles. Deleting strips (1) yields a two-colour design, so the search does not investigate deleting strips (1,2), (1,3) and (1,2,3). Indeed, these combinations are part of the same direction, even though they are two-coloured. Even though deleting strips (2) and (3) does not yield two-colour

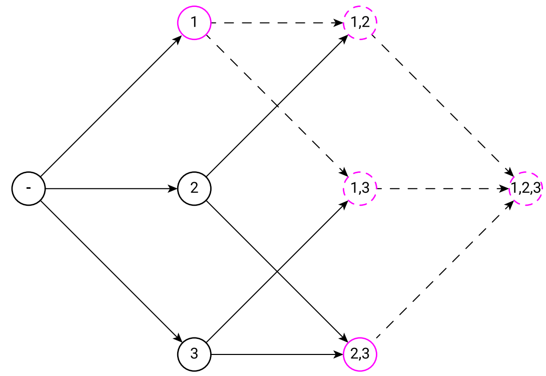



**Fig. 29.** Principle of the search algorithm for two-colour quad meshes. The search yields only the closest two-colour quad meshes in independent design directions. The two-colour quad meshes are highlighted in pink, and the discarded quad meshes downstream are marked by dashes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

designs, deleting strips (2,3) is investigated, as this combination is independent of combination (1). Deleting strips (2,3) yields a two-colour design, so the search does not investigate deleting strips (1,2,3), although the search already discarded this combination. The algorithm, therefore, reduces computation to the combinations (1), (2), (3) and (2,3) only and returns the two-colour design from the combinations (1) and (2,3).

This search approach, using an increasing number of rules, guarantees that the design obtained first is the closest one.

The implementation of the search algorithm is detailed in the next section.

### 4.2. Detailed algorithm

For an input quad mesh with $n$ strips, the search algorithm iteratively tests and discard all strip combinations:

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n. \tag{13}$$

Starting with $k = 0$, the $\binom{n}{k}$ combinations of $k$ strips among the total $n$ strips are enumerated and tested, $k$ is increased by one and the operation is repeated.

For each combination, the rule applies to a copy of the input quad mesh and its strip graph with deletion of the mesh strips and the graph nodes corresponding to the combination. The search tests the design against two criteria.

#### 4.2.1. Validity criteria

For each combination, a decision is made based on the validity of the design against two criteria:

- on pattern topology: the new graph must be two-coloured; and,
- on shape topology: the new mesh must have the same shape topology.

*Pattern topology.* The new graph must be two-coloured. This criterion is more decisive for low numbers of deleted strips. Indeed, deleting strips deletes nodes in the graph, and subgraphs of two-colour graphs are two-coloured.

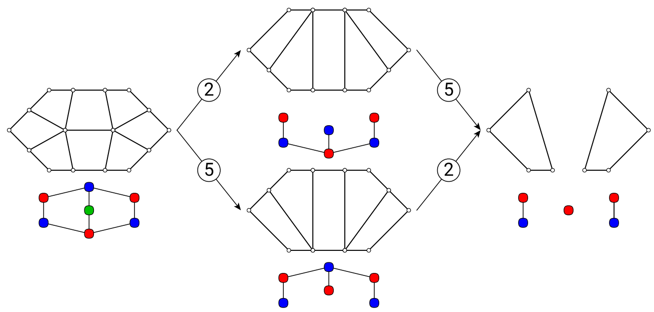



**Fig. 30.** Deleting some strips can yield a different shape topology, although yielding a two-colour quad mesh.

*Shape topology.* The new mesh must have the same shape topology. The mesh must be manifold, have the same number of handles and boundaries. This criterion is more decisive for high numbers of deleted strips. Indeed, deleting too many strips can result in a different shape topology. In Fig. 30, deleting two strips results in a different shape topology, splitting the boundary in two. This combination is therefore not valid, although yielding a two-colour design.

This example highlights an isolated graph node resulting from a collateral strip deletion. The rule deletions do not apply to this strip, but all of its faces are part of the deleted strips. This combination of two rules creates a distance of three. The combination yields a two-colour quad meshes too far, too early. The search must check the other combinations deleting fewer strips before. Therefore, the search discards such combinations.

The validity of the two criteria provides a reduction strategy on the search over the $2^n$ combinations.

#### 4.2.2. Search reduction

The search does not seek all two-colour designs, only the closest ones for independent design direction. The number of tests on downstream designs can, therefore, be reduced based on the results against the two criteria of the upstream designs. The downstream designs from a design are the ones at a greater distance in the same direction. A combination of strips $X$ is downstream another combination of strips $Y$ if $X$ is a subset of $Y$:

$$X \subset Y. \tag{14}$$

If a combination of strips yields a two-colour quad mesh, the downstream combinations are discarded during the search, as they are farther in the same direction. If a combination of strips yields a different shape topology, the search discards the downstream combinations, as applying more strip deletions will not restore the shape topology.

Thereby, this scheme reduces the iterative enumeration and avoids testing all combinations.

#### 4.2.3. Search termination

Termination does not occur after $2^n$ tests but earlier. Indeed, iterative enumeration terminates before $k$ reaches $n$, when the reduced search discards all of the downstream combinations. The designer can also specify custom termination criteria, like a maximum distance as $k_{max}$ or a minimum number of yielded designs. More generally, the designer can terminate the search when pleased with the current pool of design and does not wish to search further.

```
get a quad mesh with n strips
generate the strip graph of the mesh
start an empty set for the two-colour topologies with the same shape topology
start an empty set for the discarding sub-combinations
initiate k = −1
while k < n do
    k+ = 1
    for each combination of k strips among the n strips do
        if the combination causes collateral strip deletions then
            discard combination
        end
        if one of the discarding sub-combinations is a subset of the combination then
            discard combination
        end
        copy the original mesh
        delete the k strips of the copy mesh
        if the shape topology is different then
            add the combination to the set of discarding sub-combinations
        end
        else
            copy the original graph
            delete the corresponding k nodes of the copy graph
            if the graph is two-coloured then
                add the copy mesh to the set of two-colour topologies with the same shape topology
                add the combination to the set of discarding sub-combinations
            end
        end
    end
end
return the successful topologies
```

**Algorithm 1:** Projection of a quad mesh to the closest two-colourable quad meshes in independent directions.

Algorithm 1 provides the pseudo-code of the two-colouring search.

The complexity of the search algorithm is exponential $O(2^n)$. However, thanks to the search reduction scheme and termination criteria, the actual number of tests is significantly below $2^n$. Analytical evaluation of this number of tests is challenging but numerical results are provided in the following section, which applies this algorithm on several test cases.

## 5. Applications

The following examples validate the two-colour projection search algorithm and illustrate its application on a design scenario as a tool for two-colour topology-finding.

### 5.1. Validation tests

A set of examples validate this algorithm and provide some numerical values on the influence of the reduction scheme, the size of the two-colour subspace and a heuristic termination rule.

The examples are shown in Figs. 31 to 34, with the found two-colour designs grouped by distance from the initial non-two-colour design. No user-based termination criterion is set. Deleting strips coarsens the mesh and can cause faces to overlap. Nevertheless, the topology is correct and the overlaps removed during geometrical processing.

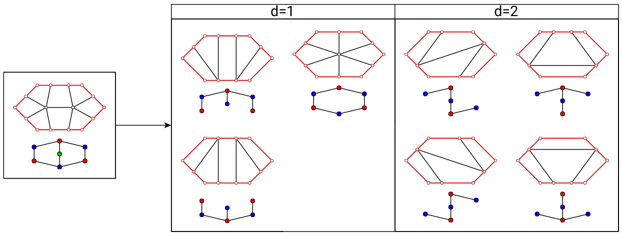



**Fig. 31.** Two-colour projection applied to a seven-strip quad mesh with a hexagonal shape.

The initial design for the hexagonal shape in Fig. 31 is not two-coloured because of the two five-valent singularities. Two-colour projection yields seven two-colour topologies, three are at a distance of one and four at a distance of two. Two-colouring becomes possible thanks to either merging the two five-valent singularities into a six-valent one or moving them to the boundary.

The initial designs for the pentagonal shape in Fig. 32 are not two-coloured. They differ by one strip only, which was added from the design in Fig. 32(a) to the one in Fig. 32(b). The projection of the five-strip topology yields five two-colour topologies, all at a distance of one, while the projection of the six-strip topology yields eight two-colour topologies, all at a distance of two. Adding a strip resulted in having more but farther two-colour topologies.

In Figs. 33 and 34, the initial designs are based on a skeleton-based decomposition of the shape, presented in [33].

The initial topology for the rectangular shape with one opening in Fig. 33 yields nine two-colour topologies, only one at a distance of one, and eight at a distance of two.

The initial topology for the rectangular shape with two openings in Fig. 34 yields 31 two-colour topologies.

Fig. 35 shows one of the two-colour designs from the example in Figs. 33 and 34 after densification while fulfilling the density requirement for face two-colouring to produce checkerboard patterns.

The detailed results for each example are in Tables 4 to 8 in the Appendix. Each table contains the number of potential combinations, the number of tested combinations due to the search reduction and the results against the two validity criteria: the two-colour topologies with the right shape topology (O), the non-two-colour topologies with the right shape topology (-) and the topologies with a wrong shape topology (X). The total number of yielded topologies is highlighted in green and the termination criterion in red.

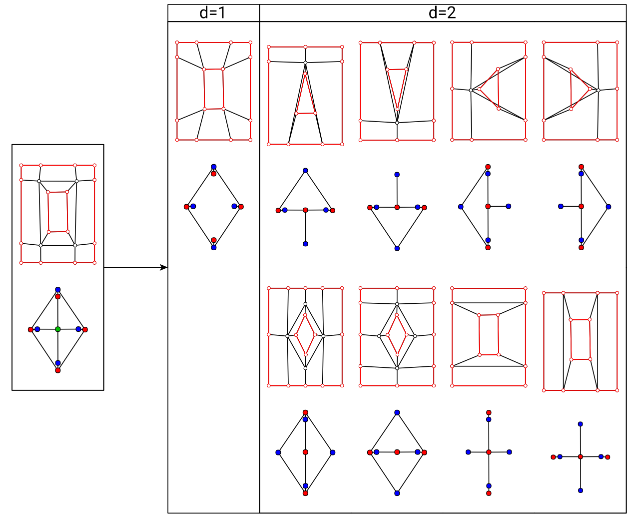



**Fig. 33.** Two-colour projection applied to a nine-strip quad mesh with a rectangular shape with one opening.

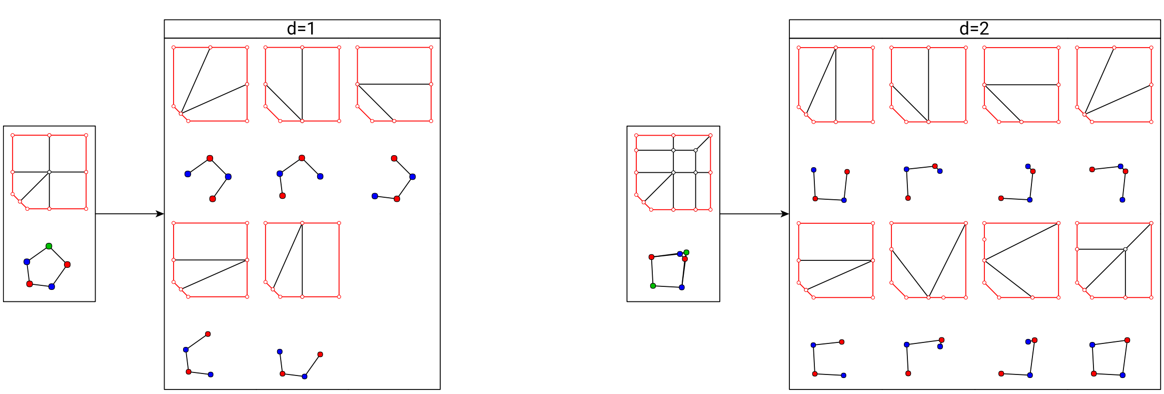
**Table 3**
Summary results of the validation examples for two-colour projection search.

| Figure | 31 | 32(a) | 32(b) | 33 | 34 |
|---|---|---|---|---|---|
| n | 7 | 5 | 6 | 9 | 14 |
| $2^n$ | 128 | 32 | 64 | 512 | 16,384 |
| % tested | 10.2 | 15.6 | 37.5 | 11.1 | 6.1 |
| % yielded | 5.5 | 15.6 | 12.5 | 1.8 | 0.2 |
| Time [s] | 0.33 | 0.078 | 0.62 | 4.4 | 790 |

Table 3 shows the summary results per example: the number of strips $n$, the total number of combinations for strip deletions $2^n$, the percentage of tested combinations and the percentage of yielded topologies.

These results highlight:

- the significant decrease of tested combinations thanks to the search reduction, especially for the topology with many strips, with 6.1% for the 14-strip topology;
- the small part of combinations to yield the closest two-colour topologies in the different design directions, especially for the topology with many strips, with 0.2% for the 14-strip topology;
- the fast computation time for low numbers of strips for efficient user–machine interactivity; the drastic increase in computation time for high numbers of strips, stressing the




(a) Five-strip quad mesh

(b) Six-strip quad mesh

**Fig. 32.** Two-colour projection applied to quad meshes with a pentagonal shape.

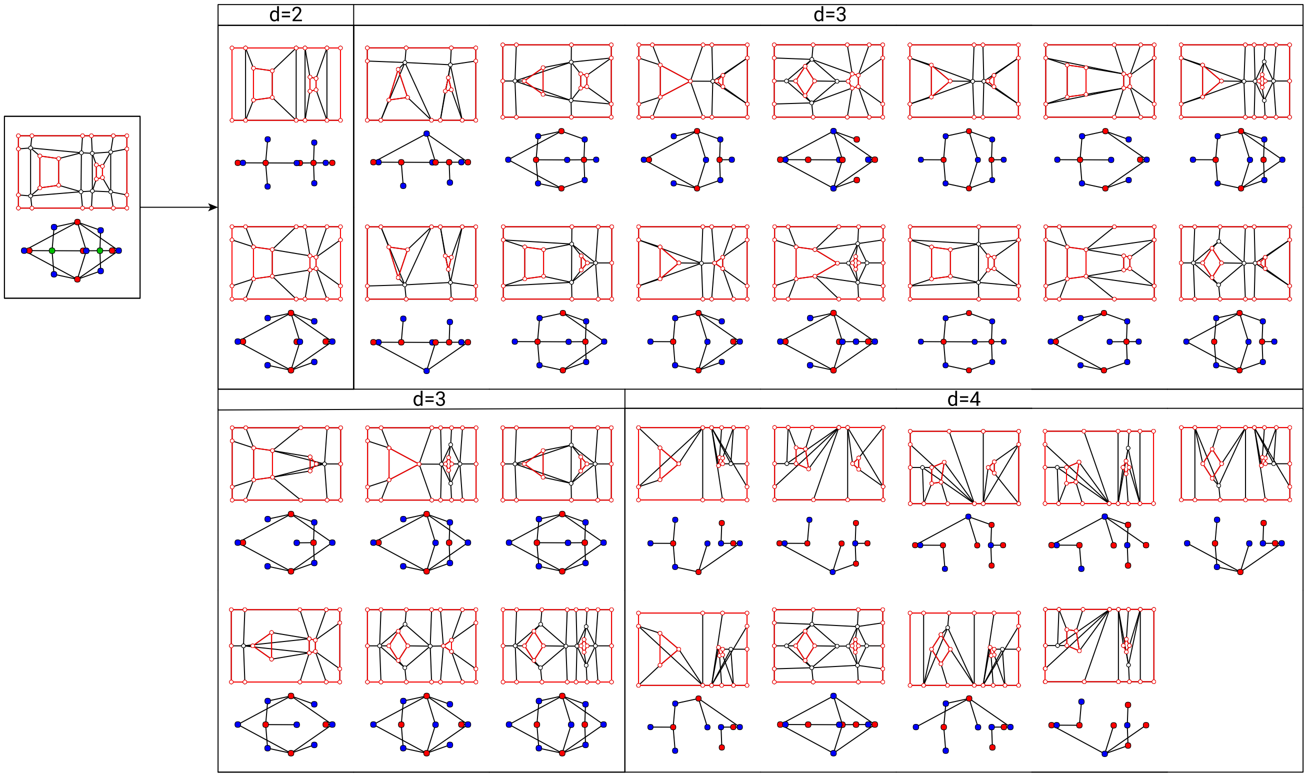



**Fig. 34.** Two-colour projection applied to a 14-strip quad mesh with a rectangular shape with two openings.

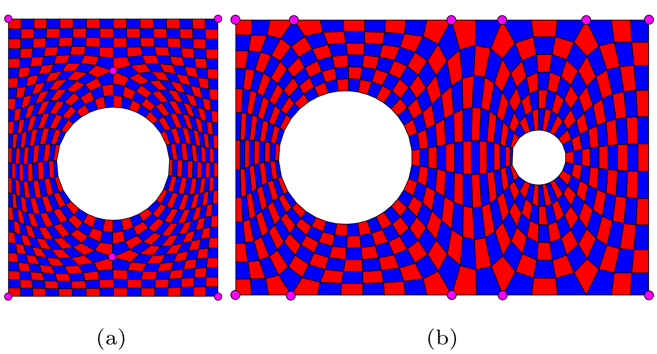



**Fig. 35.** Chequerboard patterns obtained after fulfilment of the density and singularity requirements for face two-colouring using the two-colour projection.

importance of the coarse mesh as a computation means and the need for efficient implementations, with parallel computing being a suitable means to run the multiple independent tests.

Termination occurs when there is no combination to be tested or when there is no non-two-colour topology with the right shape topology at a value $k$. These results intuit an earlier practical termination criterion: stopping the procedure when increasing $k$ stops yielding two-colour topologies with the right shape topology. This criterion reduces by 34% and 27% the number of tests for the 9-strip and the 14-strip topology, respectively.

Comparing the results for the two topologies that differ by one strip in Fig. 32 illustrates the challenge in including strip additions in two-colour projection. Although adding strips cannot make a topology two-coloured, as nodes and edges are added in the graph, which cannot reduce its chromatic number, the projection then yields more results but at a greater distance. The fundamental limitation is the infinite combinatorial richness in adding strips. Nevertheless, a stochastic search using the index-based characterisation could integrate the addition rules.

This search algorithm for two-colour quad meshes can apply to two-colour topology finding of patterns for structural design.

### 5.2. Design application

The elastic gridshell is an example of a structural system that benefits from a structural pattern extracted from a quad-mesh with two-colour polyedges. These polyedges are partitioned into two sets of continuous beams for the top and the bottom layers of the gridshell to avoid having continuous beams weaving between the two layers, reducing the bending pre-stress and easing the layout process [3].

Fig. 36 shows the shape intent, similar to a trimmed torus with a protrusion. To map this shape with a quad-mesh pattern suitable for an elastic gridshell, polyedge two-colouring, and therefore the singularity requirement, must be fulfilled.

Applying a skeleton-based decomposition [33] on this input shape yields the coarse quad mesh in Fig. 37. However, this quad mesh does not comply with the singularity requirement, because of the five-valent singularities, evidenced by the three-coloured strip graph and the index-based characterisation.

Applying the two-colour projection yields the four coarse quad meshes with two-colour strips in Fig. 38. They all result from one strip deletion and are therefore at a distance of one from the initial coarse quad mesh. The topology fulfils the singularity requirement for polyedge two-colouring thanks to merging the initial five-valent singularities or moving them to the boundary. The resulting quad meshes partition the continuous beams into two groups, the top layer in red and the bottom layer in blue.

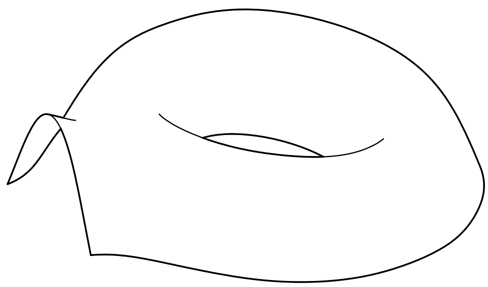


**Fig. 36.** Shape intent to map a two-colour quad mesh for the structural pattern of an elastic gridshell.

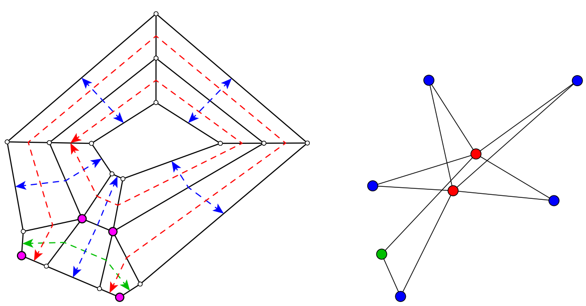



**Fig. 37.** A coarse quad-mesh that does not comply with the two-colouring requirements. The five-valent singularities do not allow fulfilling the singularity requirement, as evidenced by the three-colour strip graph.

Among the four designs, the best one can be chosen based on a topological argument, like minimising the number of singularities on the boundary for aesthetics, or a geometrical one, like minimum curvature in the polyedges to reduce the bending in the beams. Both arguments, specific to this application, hint at the third design in Fig. 38.

Thanks to two-colour topology finding, a set of suitable quad-mesh topologies are found, which can be further processed freely with the form-finding and structural-assessment means chosen by the designer.

## 6. Conclusion

This paper introduced *two-colour topology finding* of quad-mesh patterns, tackling the two-colour topological requirement for many structural patterns in a design exploration manner.

The proposed search algorithm provides not a unique solution but several design propositions, the most similar ones to the input design for different design directions.

The different types of two-colouring for quad meshes, along with the corresponding requirements and means for characterisation were identified. A topological grammar for the exploration of the design space of quad meshes and its corresponding grammar-based distance were formalised. A projection search algorithm was detailed to find the two-colour design subspace, validated on test examples and applied to the topological design of patterns for an elastic gridshell.

The algorithm simplifies the approach by constraining the combinatorial exploration to a finite design space using only strip deletion rules, discarding the addition rules. Nevertheless, considering strip addition can provide more design options, as demonstrated in the validation example in Fig. 32. To tackle such unbounded exploration, a stochastic search, as opposed to an enumerative search, using a metric like the index-based characterisation in Eq. (5) can be investigated.

The range of applications of two-colour quad-mesh patterns includes structural as well as architectural and decorative applications. A stronger requirement for quad-mesh patterns preserves the orientation between the strips of the same colour, constraining singularity design to 4k-valent, like eight-valent singularities, instead of 2k-valent singularities. A three-colouring requirement also applies to triangulated-mesh patterns, constraining the singularities to 3k-valent vertices, like nine-valent singularities, for the design of three-layered patterns for instance. The presented approach can be extended to tackle these similar problems.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

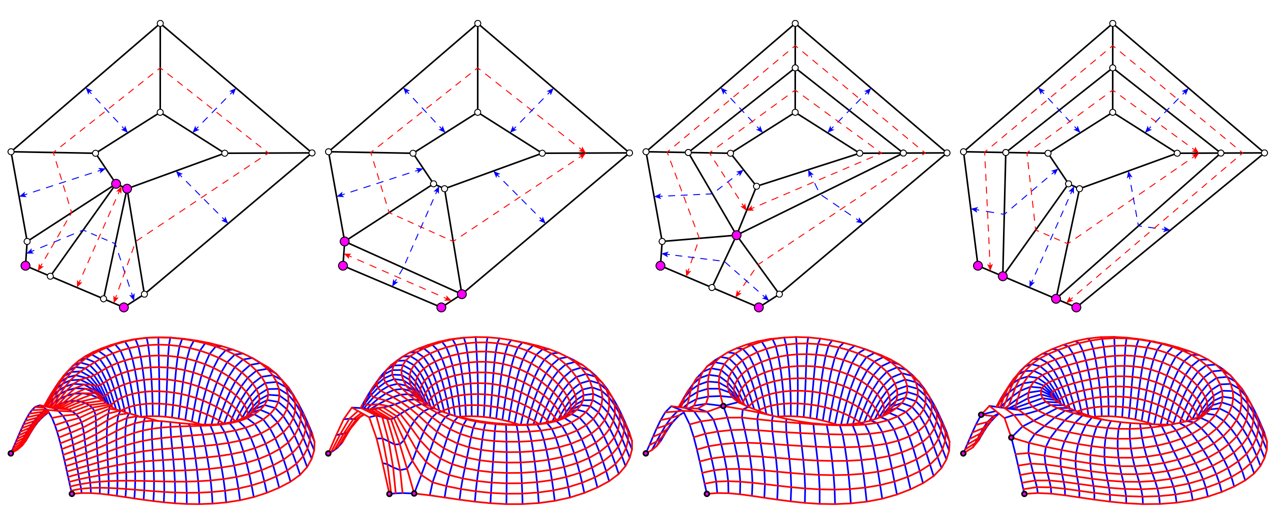


**Fig. 38.** Two-colour coarse quad meshes resulting from two-colour topology finding and their corresponding quad-mesh patterns with two-colour polyedges representing continuous beams for elastic gridshells.

**Table 4**
Detailed results of the two-colour projection for the hexagonal shape (Fig. 31).

| k | $\binom{n}{k}$ | ? | O | - | X |
|---|---|---|---|---|---|
| 1 | 7 | 7 | 3 | 4 | 0 |
| 2 | 21 | 6 | 4 | 2 | 0 |
| 3 | 35 | 0 | 0 | 0 | 0 |
| Σ | 63 | 13 | 7 | 6 | 0 |

**Table 5**
Detailed results of the two-colour projection for the pentagonal shape (Fig. 32(a)).

| k | $\binom{n}{k}$ | ? | O | - | X |
|---|---|---|---|---|---|
| 1 | 5 | 5 | 5 | 0 | 0 |
| Σ | 5 | 5 | 5 | 0 | 0 |

**Table 6**
Detailed results of the two-colour projection for the pentagonal shape bis (Fig. 32(b)).

| k | $\binom{n}{k}$ | ? | O | - | X |
|---|---|---|---|---|---|
| 1 | 6 | 6 | 0 | 6 | 0 |
| 2 | 15 | 15 | 8 | 7 | 0 |
| 3 | 20 | 3 | 0 | 3 | 0 |
| 4 | 15 | 0 | 0 | 0 | 0 |
| Σ | 56 | 24 | 8 | 16 | 0 |

**Table 7**
Detailed results of the two-colour projection for the rectangular shape with one opening (Fig. 33).

| k | $\binom{n}{k}$ | ? | O | - | X |
|---|---|---|---|---|---|
| 1 | 9 | 9 | 1 | 8 | 0 |
| 2 | 36 | 28 | 8 | 20 | 0 |
| 3 | 84 | 16 | 0 | 16 | 0 |
| 4 | 126 | 4 | 0 | 4 | 0 |
| 5 | 126 | 0 | 0 | 0 | 0 |
| Σ | 381 | 57 | 9 | 48 | 0 |

**Table 8**
Detailed results of the two-colour projection for the rectangular shape with two openings (Fig. 34).

| k | $\binom{n}{k}$ | ? | O | - | X |
|---|---|---|---|---|---|
| 1 | 14 | 14 | 0 | 14 | 0 |
| 2 | 91 | 91 | 2 | 83 | 6 |
| 3 | 364 | 282 | 20 | 236 | 26 |
| 4 | 1001 | 332 | 9 | 319 | 4 |
| 5 | 2002 | 202 | 0 | 202 | 0 |
| 6 | 3003 | 64 | 0 | 64 | 0 |
| 7 | 3432 | 10 | 0 | 10 | 0 |
| 8 | 3003 | 0 | 0 | 0 | 0 |
| Σ | 12910 | 995 | 31 | 928 | 36 |

## Appendix

Tables 4 to 8 in the appendix contains the detailed results for each validation example in Section 5.1, shown in Figs. 31 to 34. Each table contains the number of potential combinations, the number of tested combinations due to the search reduction and the results against the two validity criteria: the two-colour topologies with the right shape topology (O), the non-two-colour topologies with the right shape topology (-) and the topologies with a wrong shape topology (X). The total number of two-colour designs with the right shape topology is highlighted in green and the termination criterion in red.

## References

[1] Mesnil Romain, Douthe Cyril, Baverel Olivier, Gobin Tristan. Form finding of nexorades using the translations method. Autom Constr 2018;95:142–54.
[2] Hull Thomas. On the mathematics of flat origamis. Congr Numer 1994;215–24.
[3] Du Peloux Lionel, Tayeb Frédéric, Caron Jean-François, Baverel Olivier. 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.
[4] Baverel Olivier. Nexorades: a family of interwoven space structures [Ph.D. thesis], University of Surrey Guildford; 2000.
[5] Brocato Maurizio, Mondardini Lucia. Geometric methods and computational mechanics for the design of stone domes based on abeille's bond. In: Ceccato Cristiano, Hesselgren Lars, Pauly Mark, Pottmann Helmut, Wallner Johannes, editors. Advances in Architectural Geometry 2010. Springer; 2010, p. 149–62.
[6] Estrin Yuri, Dyskin Arcady V, Pasternak Elena. Topological interlocking as a material design concept. Mater Sci Eng C 2011;31(6):1189–94.
[7] Tachi Tomohiro. Generalization of rigid-foldable quadrilateral-mesh origami. J Int Assoc Shell Spatial Struct 2009;50(3):173–9.
[8] Lebée Arthur. From folds to structures, a review. Int J Space Struct 2015;30(2):55–74.
[9] Stitic Andrea, Weinand Yves. Timber folded plate structures–topological and structural considerations. Int J Space Struct 2015;30(2):169–77.
[10] Mesnil Romain, Baverel Olivier, Douthe Cyril, Caron Jean-François, Léger Bruno. Structural morphology and performance of plated structures with planar quadrilateral facets. J Int Assoc Shell Spatial Struct 2017;58(1).
[11] Otto Frei, Schauer Eda, Hennicke Jürgen, Hasegawa Takeshi. IL 10 Gitterschalen/Grid Shells: Bericht über das japanisch-deutsche Forschungsprojekt STI, durchgeführt von Mai 1971 bis Mai 1973 am Institut fur Leichte Flächentragwerke. Seibu Construction Company / Institut für leichte Flächentragwerke / Krämer; 1974.
[12] Douthe Cyril, Mesnil Romain, Orts Hugo, Baverel Olivier. Isoradial meshes: Covering elastic gridshells with planar facets. Autom Constr 2017;83:222–36.
[13] Marquis Pierre, Cerosimo Amandine, Douthe Cyril. Building elastic gridshells from patches. In: Proceedings of the Annual Symposium of the International Association for Shell and Spatial Structures 2017. 2017.
[14] Avelino Ricardo, Baverel Olivier. Structural analysis of gridshells designed from singularities. In: Proceedings of the Annual Symposium of the International Association for Shell and Spatial Structures 2017. 2017.
[15] Baverel Olivier, Popovic Larsen Olga. A review of woven structures with focus on reciprocal systems-nexorades. Int J Space Struct 2011;26(4):281–8.
[16] Popescu Mariana, Rippmann Matthias, Van Mele Tom, Block Philippe. Automated generation of knit patterns for non-developable surfaces. In: Humanizing digital reality: design modelling symposium Paris 2017. Springer; 2017, p. 271–84.
[17] Ayres Phil, Martin Alison Grace, Zwierzycki Mateusz. Beyond the basket case: a principled approach to the modelling of kagome weave patterns for the fabrication of interlaced lattice structures using straight strips. In: Hesselgren Lars, Kilian Axel, Sorkine-Hornung Olga, Malek Samar, Olsson Karl-Gunnar, Williams Christopher JK, editors. Advances in Architectural Geometry 2018. Klein Publishing GmbH (Ltd.); 2018.
[18] Norman AD, Seffen Keith A, Guest Simon D. Morphing of curved corrugated shells. Int J Solids Struct 2009;46(7–8):1624–33.
[19] Malek Samar, Williams Christopher JK. The equilibrium of corrugated plates and shells. Nexus Netw J 2017.
[20] Liu Yang, Pottmann Helmut, Wallner Johannes, Yang Yong-Liang, Wang Wenping. Geometric modeling with conical meshes and developable surfaces. Assoc Comput Mach Trans Graph 2006;25(3):681–9.
[21] Schiftner Alexander, Leduc Nicolas, Bompas Philippe, Baldassini Niccolo, Eigensatz Michael. Architectural geometry from research to practice: the eiffel tower pavilions. In: Hesselgren Lars, Sharma Shrikant, Wallner Johannes, Baldassini Niccolo, Bompas Philippe, Raynaud Jacques, editors. Advances in Architectural Geometry 2012. Springer; 2012, p. 213–28.
[22] Liu Yang, Wang Wenping. On vertex offsets of polyhedral surfaces. In: Pottman Helmut, Kilian Axel, Hofer Michael, editors. Advances in Architectural Geometry 2008. TU Wien, Vienna; 2008, p. 61–4.

[23] Bobenko Alexander I, Huhnen-Venedey Emanuel. Curvature line parametrized surfaces and orthogonal coordinate systems: discretization with Dupin cyclides. Geom Dedicata 2014;159(1):207–37.
[24] Mesnil Romain, Douthe Cyril, Baverel Olivier, Leger Bruno. Generalised cyclidic nets for shape modelling in architecture. Int J Archit Comput 2017;15(2):148–68.
[25] Gondran Michel, Minoux Michel. Graphs and algorithms. Wiley; 1984.
[26] Takezawa Masahito, Imai Takuma, Shida Kentaro, Maekawa Takashi. Fabrication of freeform objects by principal strips. Assoc Comput Mach Trans Graph 2016;35(6):225.
[27] Schiftner Alexander, Balzer Jonathan. Statics-sensitive layout of planar quadrilateral meshes. In: Ceccato Cristiano, Hesselgren Lars, Pauly Mark, Pottmann Helmut, Wallner Johannes, editors. Advances in Architectural Geometry 2010. Springer; 2010, p. 221–36.
[28] Caigui Jiang, Peng Chi-Han, Wonka Peter, Pottmann Helmut. Checkerboard patterns with black rectangles. Assoc Comput Mach Trans Graph 2019.
[29] Conway John H, Burgiel Heidi, Goodman-Strauss Chaim. The symmetries of things. CRC Press; 2016.
[30] Marinov Martin, Kobbelt Leif. A robust two-step procedure for quad-dominant remeshing. In: Computer graphics forum, vol. 25. Wiley Online Library; 2006, p. 537–46.
[31] Bommes David, Zimmer Henrik, Kobbelt Leif. Mixed-integer quadrangulation. Assoc Comput Mach Trans Graph 2009;28(3):77.
[32] Peng Chi-Han, Barton Michael, Jiang Caigui, Wonka Peter. Exploring quadrangulations. Assoc Comput Mach Trans Graph 2014;33(1):12.
[33] Oval Robin, Rippmann Matthias, Mesnil Romain, Van Mele Tom, Baverel Olivier, Block Philippe. Feature-based topology finding of patterns for shell structures. Autom Constr 2019;103:185–201.
[34] Oval Robin. compas_singular: a Python framework for topology finding of patterns for structural design. 2017, [Accessed on 19 April 2021].
[35] Van Mele Tom, Liew Andrew, Méndez Echenagucia Tomás, Rippmann Matthias, et al. compas: A framework for computational research in architecture and structures. 2017, [Accessed on 19 April 2021].
[36] Campen Marcel, Bommes David, Kobbelt Leif. Dual loops meshing: quality quad layouts on manifolds. Assoc Comput Mach Trans Graph 2012;31(4):110.
[37] Akleman Ergun, Ke Shenyao, Wu You, Kalantar Negar, Borhani AliReza, Chen Jianer. Construction with physical version of quad-edge data structures. Comput Graph 2016;58:172–83.
[38] Dailey David P. Uniqueness of colorability and colorability of planar 4-regular graphs are NP-complete. Discrete Math 1980;30(3):289–93.
[39] Welsh Dominic JA, Powell Martin B. An upper bound for the chromatic number of a graph and its application to timetabling problems. Comput J 1967;10(1):85–6.
[40] Hamming Richard W. Error detecting and error correcting codes. Bell Syst Tech J 1950;29(2):147–60.
[41] Levenshtein Vladimir I. Binary codes capable of correcting deletions, insertions, and reversals. Sov Phys Dokl 1966;10(8):707–10.