Computer-Aided Design 168 (2024) 103670

Contents lists available at ScienceDirect

Computer-Aided Design

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

Bending-Reinforced Grid Shells for Free-form Architectural Surfaces

Francesco Laccone [a,*], Nico Pietroni [b], Paolo Cignoni [a], Luigi Malomo [a]

[a] Institute of Information Science and Technologies "A. Faedo" (ISTI), National Research Council of Italy (CNR), via G. Moruzzi 1, Pisa, 56124, Italy
[b] School of Computer Science, Faculty of Engineering and Information Technology, University of Technology Sydney (UTS), 15 Broadway Ultimo, Sydney, Australia

ARTICLE INFO

Dataset link: https://doi.org/10.5281/zenodo.10391043

Keywords:
Form-finding
Structural design
Space Frame
Mesh generation
Computational model
Optimization

ABSTRACT

We introduce a new method for designing reinforcement for grid shells and improving their resistance to out-of-plane forces inducing bending. The central concept is to support the base network of elements with an additional layer of beams placed at a certain distance from the base surface. We exploit two main techniques to design these structures: first, we derive the orientation of the beam network on a given initial surface forming the grid shell to be reinforced; then, we compute the height of the additional layer that maximizes its overall structural performance. Our method includes a new formulation to derive a smooth direction field that orients the quad remeshing and a novel algorithm that iteratively optimizes the height of the additional layer to minimize the structure's compliance. We couple our optimization strategy with a set of constraints to improve buildability of the network and, simultaneously, preserve the initial surface. We showcase our method on a significant dataset of shapes to demonstrate its applicability to cases where free-form grid shells do not exhibit adequate structural performance due to their geometry.

1. Introduction

Grid shells are elegant and lightweight structures used in architecture to cover large spaces. Their lightness is due to the structural efficiency resulting from membrane actions through axial forces along a grid of beams. This behavior that relies on in-plane stiffness rather than bending can be obtained only by employing a careful shape design, i.e., form-finding and suitable grid topology. Usually, the shape of a grid shell is a product of the joint efforts of architects and engineers who closely collaborate in the early conceptual design phase. Thus, it fulfills aesthetic and functional requirements while guaranteeing structural efficiency. However, this joint effort only sometimes happens, and the shape is sometimes sculpted with artistic intents only, resulting in free-form shapes that are inefficient for given load and boundary conditions.

In these cases, bending forces inherently arise since transversal forces cannot be balanced by in-plane stress. Also, significant bending can be observed in many common cases, such as in the presence of asymmetrical load [1] or large openings [2]. The structural performance can be improved with shape or topology optimization, but these methods might significantly alter the original surface and its aesthetics or even violate some design constraints.

Some research works address bending in the continuum shell optimization [3,4], and study corrugation in shells [5] and in grid shells [6] to increase bending strength. Form-finding techniques of self-supporting [7], tension-only [8], and mixed of tension-compression structures [9] can include constraints on the final shape compared to a desired target.

Given a grid shell, bending stiffness may be achieved by increasing the beams' cross sections or the joints' stiffness, with the drawback of increasing fabrication costs and weight of the entire structure. Alternatively, another method is reinforcing the grid shell with a support structure that counterbalances out-of-plane forces. This approach has been applied to design two masterpieces of structural engineering: the Berlin Central Station roof [10] and the Waterloo Station roof [11]. Both structures are shaped as *translation surfaces* in which a transversal 2D reinforced truss is replicated along a 2D path.

Inspired by these built examples, we propose a new computational pipeline to design a reinforcement structure automatically for a predefined load condition capable of efficiently supporting out-of-plane loads given a fixed initial geometry. Our pipeline devises *reinforced grid shells* by connecting the base grid shell structure with another structure having similar connectivity but placed at a certain distance from the original grid shell, as in Fig. 1. These two interconnected structures can generate bending inertia from the spacing between the members of the two surfaces in the out-of-plane direction.

Our method consists of two main steps: first, we tessellate the initial shape (main layer) with a grid of beams to maximize the resistance to out-of-plane bending, and then we optimize the location of the additional layer using a novel energy formulation. Finally, we perform a final step to include primary fabrication constraints to improve the

* Corresponding author.
E-mail address: francesco.laccone@isti.cnr.it (F. Laccone).

https://doi.org/10.1016/j.cad.2023.103670
Received 24 December 2022; Received in revised form 15 December 2023; Accepted 17 December 2023
Available online 23 December 2023

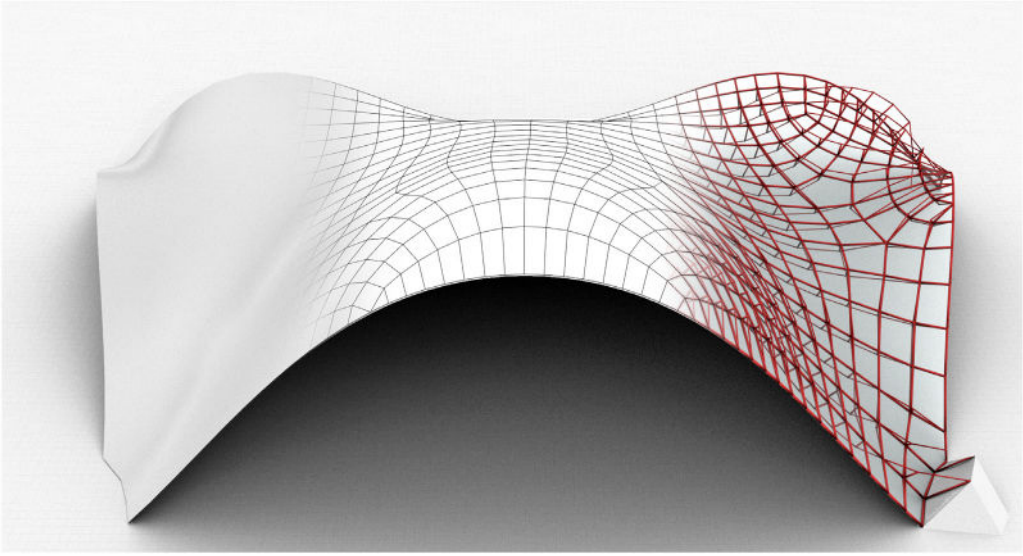

Fig. 1. Our bending-reinforced grid shell: starting from a generic continuous free-form surface, we define a tessellation to align the structural elements in a discrete setting based on the FE analysis for the continuous surface; then, we compute a network of beams with variable height from the starting tessellation, which lies on the input surface.

solution, such as solving beams' self-intersections. Our method can automatically derive full-scale architectural structures that have adequate static performances and are feasible from a fabrication viewpoint. To our knowledge, this is the first method to automatically derive reinforcement structures for fixed non-membrane shapes in a 3D discrete setting.

2. Related work

Grid shell design poses several interconnected challenges on aesthetics, mechanical behavior, tessellation, fabrication, and assembly. Therefore, it requires multi-disciplinary skills ranging from architecture, structural engineering, mathematics, and computer graphics. Their combination has recently settled into a new research field named Architectural Geometry [12].

Form-finding. Form-finding is finding a shape in equilibrium subject to specific loading and boundary conditions so that it will only experience in-plane internal forces, i.e., axial forces in grid shells. Core numerical form-finding methods exist [13], such as the Force Density [14], the Dynamic Relaxation, and the Thrust Network Method [15]. They have been combined with Airy stress functions and graphic statics for obtaining more creative design explorations [16,17]. Similar strategies can be found in general equilibrium networks of elements in tension-compression [18,19].

Structural optimization. Unlike form-finding, obtaining efficient structural grids for a given shape is an indirect problem and can be solved with optimization. Here, the tessellation plays a critical role in the static performance of the structure. In [20], the authors optimize grids of elements in response to multiple load cases using continuum homogenization. In [21], the authors use the principal stress components to tessellate the grid for a better alignment of the elements. In [22], the authors optimize quad mesh directions to obtain minimal weight. Relevant optimization strategies also characterize structures, which have different mechanical behavior, such as tensegrities [23–25], spatial structures [26,27], or bending-active structures [28–30].

Other contributions include variables of different nature in the objective function or in the constraints, such as aesthetics and buildability. In the last decades, a significant effort has been spent on rationalizing free-form structures to achieve feasible construction with reduced manufacturing costs [31–34] eventually modifying the connectivity [35]. Other papers propose optimization methods to achieve developable surfaces [36,37] or meshes composed of planar elements [38–40], including repeated elements [41] or meshes with no torsion on the nodes [42]. More general methods are designed to optimize the tessellation [43,44], additionally including curved features [45].

Our approach lays its foundations on a different concept. Instead of modifying the initial shape, we couple the main structure (laid on the initial shape) with an additional reinforcement, making the thus-formed structure capable of bearing out-of-plane forces.

Double layer and reinforcement structures. Several concepts utilize additional structural elements to enhance the structural performances of architectural structures. The approach in [46] uses accurately placed cables to design compression-only surfaces. Adding ribs and reinforcement [47–51] to align along specific paths, i.e., principal stress directions, is a well-known strategy inspired by the Michell-like optimum structures. The strain/stress field is usually computed under the assumption of planar stress (thin shell hypothesis, no stress in the normal direction). In the paper [52], a novel strategy to reinforce 3D shells with ribs is proposed. Hence, the ribs' height is proportional to the bending. The network of ribs is aligned with a cross-field that considers the principal strain directions in the equivalent continuum. The approach, however, is conceived for 3D printing or designing surfaces with geometrically complex elements that cannot be easily adapted to large architectural structures.

In architecture, creating a double layer is the standard approach to achieving resistance to out-of-plane stress. The double layer materializes bending strength and robustness from two appropriately-distanced interconnected surfaces. These surfaces can be continuous or segmented, i.e., shaped as grids or plates. The higher the distance between the layers, the larger the bending inertia. Double-layer plate structures are typically used in timber construction to realize dry-connected lightweight shells in the form of cassettes, hollow boxes, or waffle structures [53–55]. This concept can be extended even towards less stiff materials [56]. Double-layer structures, also known as space frames, are usually made of steel and are characterized by modular elements arranged in a regular unit and repeated over a predefined grid [57]. Most are flat or single-curved, but even more complex free-form geometries are feasible.

The possibility of varying the distance between the layers following the bending moment plot has inspired several design concepts for 2D structures such as truss bridges and frames. Moreover, it inspired research based on graphic statics to transform a generic free-form curve

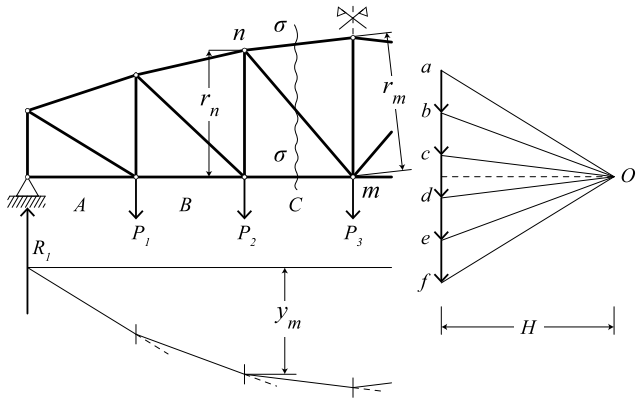

Fig. 2. Analysis of a truss: on top, Ritter method; on bottom, flexural analysis of the equivalent beam; right, force polygon.

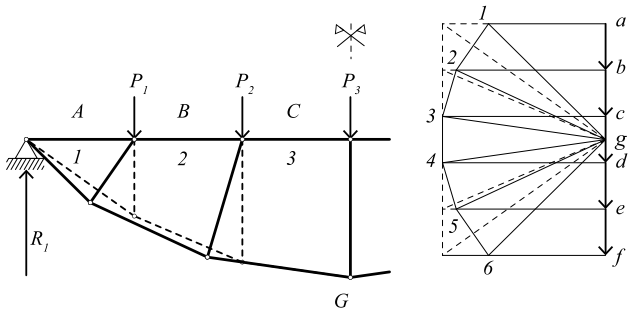

Fig. 3. Graphic static analysis of a truss: continuous lines to indicate constant-force on the (lower) added layer; dashed lines to indicate constant-force on the (upper) main layer.
Source: Adapted from [60].

into a funicular shape. In the works [58,59], the funicular of 2D and 3D shapes is achieved via post-tensioning of out-of-plane cables using graphic statics. With a similar purpose, in [60], the funicular geometry is exploited to design a tension stiffening system shaped according to the bending moment in a 2D free-form curve. Instead, our method is the first one that enables the automatic design of both the lattice and the height of the additional layer in the generic context of free-form surfaces as input.

3. Background

Apart from built examples (Berlin Central Station roof and the Waterloo Station roof), our research is inspired by analytical and graphical tools for designing and analyzing simple 2D trusses. Two key concepts are fundamental in finding an optimal reinforcement network: define the shape of the additional layer and make it uniformly stressed for efficiency.

Analysis and design of a 2D truss. We can derive the first key concept from analyzing a simply-supported truss beam subject to vertical loading. For simplicity, we assume a Pratt scheme with the hypotheses of adopting pinned joints (hinges) and nodal forces (Fig. 2). This condition allows only axial forces within the truss elements, which are easy to compute. The axial force on the upper and lower elements can be evaluated with the Ritter method, which balances the external forces on the left side of the Ritter section σ with the inner forces of the truss elements at the intersection with σ. Indeed, the equilibrium of moments at node m and n yields, respectively:

$$A_{upper} = M_m / r_m \tag{1}$$

$$A_{lower} = M_n / r_n \tag{2}$$

where M_m and M_n are the bending moments at the respective sections of the equivalent beam. A_{upper} is compressed and A_{lower} is tensioned.

These values of the bending moment can be obtained from graphic statics (Fig. 2 bottom) as $M_m = H \cdot y_m$, where y_m is the ordinate on the form polygon (or funicular) at that specific section, and H the focal distance on the force polygon (Fig. 2 right). A scale factor is applied to convert lengths into forces because, in graphic statics, force vectors are drawn in a specified scale. So, the funicular loads on a beam in equilibrium are equivalent to its bending moment plot. If the truss has a constant height r_m, per element axial force A_i is proportional to y_i. Instead, if we aim at having a constant axial force on the beams, the shape of the truss should align to the funicular of loads to have:

$$r_i = M_i / \bar{S} \tag{3}$$

in which r_i is the perpendicular distance to the main curve, $\bar{S}$ is the allowed axial force on the truss. The truss elements, which lie on the equivalent beam axis, or main curve, and are not modified, are hereinafter defined as the 'main layer'. While the other group of elements to be shaped is the 'additional layer'.

By convention, bending moments are drawn on the tensioned side of the beam, so a double layer thus formed will have an additional tensioned layer. The main layer, i.e., the equivalent beam, will be compressed in a pure bending scenario. In our method, we solve the equilibrium of the main layer, i.e., a network of beams (a grid shell), to compute the bending plot directly.

Constant-force chords on a 2D truss. The truss in Fig. 3 is shaped as the funicular or bending plot of Fig. 2 to have an additional tensioned layer. The second key concept concerns the selection of the struts that link the elements of the main (horizontal line) and the additional layer (funicular). A truss with vertical struts presents a non-constant force on the additional layer (and a constant force on the main layer), as it can be observed from the polygon of forces in Fig. 3 where the segments $g1$, $g2$... $g6$ have different lengths (dashed lines). Conversely, if these segments are forced to lie on a circle, the additional layer has a constant force (continuous lines having the same lengths). Because of the duality of form and forces, the shape of the truss must be altered to comply with the new force directions and create inclined struts.

Extending these concepts to a generic grid of beams is not straightforward because the 3D nature of the problem provides a higher number of degrees of freedom, so additional parameters are to be examined, such as the arbitrariness of loads, shape, and grid tessellation. These examples show the centrality of the concept of 'chords', as chains of truss elements on which the bending plot or the funicular varies smoothly, that we assume as the common ground of our method. Firstly, we adopt regular quad meshes since they naturally form crossing chains of edges on which chords can be built. Secondly, except for singularities, every node is traversed by two orthogonal directions that can efficiently catch bending moments. Thirdly, we can orient a quad mesh to follow a predefined cross-field over an initial surface. This strategy allows us to decouple the computation of continuous orientation on the surface from its discretization into the final network.

4. Overview

Our bending-reinforced grid shell is generated as shown in Fig. 4. The elements can be grouped into (i) a main network of beams, referred to as the main layer, and (ii) an intersecting network of beam chords linked to the main layer through struts, i.e., the additional layer. The main layer lies on the desired surface and is similar to a grid shell: it encloses a specific space and is equipped with panels. So, its primary function is supporting in-plane loads. The additional layer has a minimal visual impact and is designed to develop adequate bending strength and stiffness in collaboration with the main layer. The reinforcement has a variable height to bear out-of-plane loads efficiently. Note that

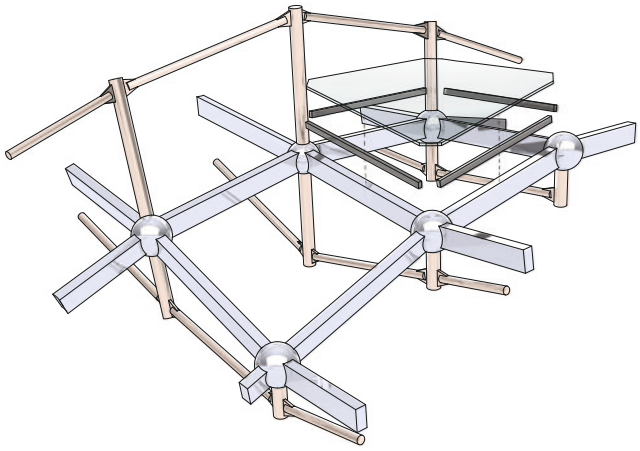

Fig. 4. Schematic closeup view of our structural system: main layer with rectangular cross sections and ball joints; additional layer with struts and solid circular 'chords' beams. The main layer's infills are equipped with panels.

the additional-layer chains can stay on both sides of the same surface in our setup. Often, one chain can pass from one side to the other to obtain, by design, always tensioned additional beams and compressed struts.

As previously discussed in Section 1, both the connectivity and the reinforcement distance from the main layer play a fundamental role in determining static performance. Ideally, both aspects should be optimized simultaneously. However, this leads to a complex combinatorial problem that we simplify, dividing it into two separate steps. First, we derive the orientation of the structural component in the continuous settings, and then we optimize the resulting discretized structure. Although shifting from a continuous surface to its beam discretization might significantly change its physical behavior, this approach has provided good results both in architectural geometry [21] and topology optimization [52]. We successfully applied a similar strategy as follows.

Continuous analysis Given a dense triangular mesh representing the input shell, we construct a volumetric representation by offsetting the mesh (see Fig. 5a). We use this solid model as input for a FEM analysis. We extract the bending directions by analyzing the variance of deformation between the top and bottom faces of the solid elements of the shell.

Meshing We use the computed bending directions to generate a smooth cross field [61], which is then used to orient and size a quadrilateral mesh (see Fig. 5b). We improve the field smoothness and define a filtering criterion that preserves the field alignment of the mesh to the relevant bending directions. The obtained mesh is converted into our main layer.

Initialization of the additional layer We initialize the additional layer by replicating the main layer and relocating its vertices to a different position, which depends on the amount and orientation of the bending forces (see Fig. 5c), recomputed from a linear static analysis.

Optimal elevation of the additional layer Having specified the connectivity, we optimize the height of the additional layer (see Fig. 5c). The distance from the main layer is modified to maximize the structural efficiency; in the optimal setup, the stress due to bending would be constant in both layers.

Post processing The final step will modify the structure to facilitate fabrication without invalidating the optimization process (see the improved additional layer in Fig. 5d with respect to the previous stage).

5. Method

5.1. Derive a field from a continuous thick shell analysis

At this stage, we consider the input surface as the mid-plane surface of a continuous elastic thick shell. The FE linear analysis on this shell returns the Cauchy stress tensor T at each vertex of the triangulation in three positions: at the mid-plane surface, on the top, and the bottom face of the shell. For a tensor, the eigenvectors and eigenvalues represent the principal directions n_p and the principal stresses σ_p, respectively. The principal directions constitute an orthonormal triplet. In particular, we are interested in the top and bottom tensors ($T^{(t)}$ and $T^{(b)}$, respectively) since they will be the place for extreme stretching/shrinking as the shell is supposed to undergo all kinds of internal forces, including bending and twisting. The mid-plane tensor represents an average value and is not sufficiently representative of our problem, as well as the bending tensor, computed under the plane section's hypothesis. Note that due to the linearity assumption, the loading magnitude and material properties affect only the amplitude of the stress field. The effect of a general uniformly-projected loading and different material properties can be obtained by multiplying the obtained stress by a scalar value.

The top and bottom principal triplets are not necessarily aligned to the tangent plane of the surface, as is the case of funicular surfaces. From these two triplets, we pinpoint the pairs of closest directions and introduce a new reference frame $\langle \bar{n_0}, \bar{n_1}, \bar{n_2} \rangle$ that is obtained by normalizing the averaged directions of the pairs. The objective is to have a common frame and to evaluate both top and bottom stresses along the same directions (Fig. 6 left).

Excluding the direction that is close to the surface normal, i.e., $\bar{n_2}$, we multiply both tensors by the directions $\bar{n_0}$ and $\bar{n_1}$ to compute the amount of force at each mesh vertex going through the surface orthogonal to these directions.

$$\sigma_p^{(t)} = T^{(t)} \cdot \bar{n_p} \quad p = 0, 1 \tag{4}$$

$$\sigma_p^{(b)} = T^{(b)} \cdot \bar{n_p} \quad p = 0, 1 \tag{5}$$

Even if the reference frame and the force along its directions are known, we can still not determine if the material is shrinking or stretching locally. So, we apply the sign of the associated eigenvalue to the stress vectors, obtaining negative values for shrink and positive values for stretch. The difference between the two values provides a measure of the curvature as the differential of deformation on the upper and lower surface of the shell. We assume that the eigenvalue sign computed in the principal frame n_p is preserved moving from a proper principal triplet to the reference frame $\bar{n}_p$. This value has been divided by Young's modulus E to obtain a strain value since the strain is $\epsilon = \sigma / E$ in the linear theory of elasticity.

$$\epsilon_p^{(t)} = sign(\sigma_p^{(t)}) \cdot \frac{1}{E} \cdot \|\sigma_p^{(t)}\| \quad p = 0, 1 \tag{6}$$

$$\epsilon_p^{(b)} = sign(\sigma_p^{(b)}) \cdot \frac{1}{E} \cdot \|\sigma_p^{(b)}\| \quad p = 0, 1 \tag{7}$$

To quantify the curvature of the deformed shell around the axis $\bar{n_0}$ and $\bar{n_1}$, we compute the relative deformation of the top and bottom surface as in Fig. 6 (right) using the strains:

$$\alpha_p = atan\left(\frac{(\epsilon_p^{(t)} + 1) - (\epsilon_p^{(b)} + 1)}{t} \right) p = 0, 1 \tag{8}$$

where t is the shell thickness. Since strain can be positive or negative, the value 1 is added to obtain only positive values. Using the angles α_i it is possible to evaluate the curvature (as the inverse of the radius) in the pth direction as:

$$K_p = \frac{1}{R} = \frac{tan(\alpha_i)}{\bar{\epsilon_i}} i = 0, 1 \tag{9}$$

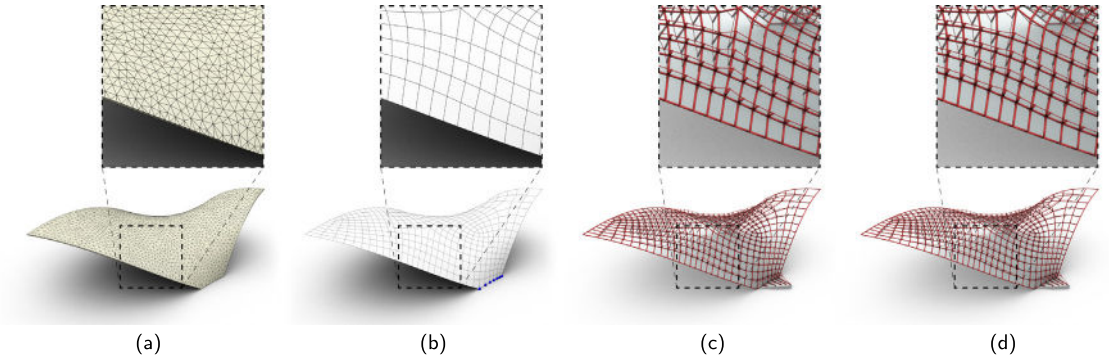

Fig. 5. Pipeline for designing reinforced grid shells: (a) thick shell (with triangle prisms); (b) cross field-aligned quad mesh; (c) initialization and optimization; (d) post-processing (e.g. smoothing of kinky lines in the additional layer).

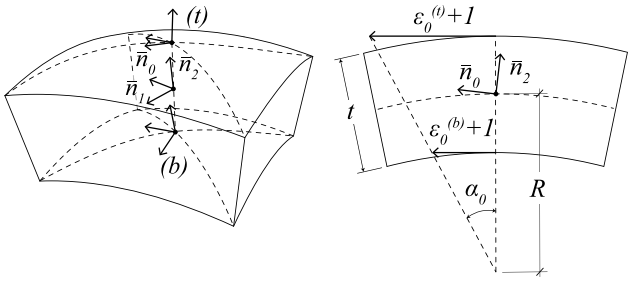

Fig. 6. Principal directions on top (t) and bottom (b) faces and the reference frame at the mid-plane used as directions of the field (left); computing the curvature as magnitudes of the field (right).

where:

$$\overline{\epsilon_p} = \frac{(\epsilon_p^{(t)} + \epsilon_p^{(b)})}{2} + 1p = 0, 1 \tag{10}$$

For each vertex of the original triangulation, the field is given by the orthogonal vectors $d_0 = \bar{n_0} \cdot K_0$ and $d_1 = \bar{n_1} \cdot K_1$ (where $K_p = |d_p|$). We denote this field as *tsf* (thick shell field). We also test our method with different input fields adopted in other state-of-the-art works: the bending field computed on the surface using the thin shell theory (*bend*) and the curvature field (*curv*).

5.2. Quad meshing and chords structure

The above procedure yields a pair of orthogonal directions d_0 and d_1 for each triangle node of the input surface, creating a cross-field. Their norms $|d_0|$ and $|d_1|$ indicate the amount of bending along such directions. We discretize the field to form a quadrilateral mesh, which will determine the structure of the main and the additional layer. In the general case, the cross-field can be somewhat irregular, and it must be smoothed to serve as a guide for generating a semi-regular quadrilateral mesh. Ideally, the smoothing process should preserve the original directions in high-intensity areas. At the same time, if the two norms $|d_0|$ and $|d_1|$ are equivalent, there is no preferred direction, i.e., the shape equally bends in all directions. Hence, there is no need to preserve any direction in those areas, and we can smooth out the field globally. We first limit the norms of $|d_0|$ and $|d_1|$ to the 10% higher percentile of the entire norms distribution. This procedure distributes the field importance evenly over the surface and reduces the sensitivity to some high peaks. Then, we quantify the importance of a field using its *anisotropy* A expressed as:

$$A = \frac{max(|d_0|, |d_1|)}{min(|d_0|, |d_1|)} \tag{11}$$

We also filter out the non-relevant direction pairs whose $max(|d_0|, |d_1|)$ is below a specified threshold (i.e., bending is negligible). To smooth the field on the surface, we use the cross-field constrained optimization of [62] implemented in the library Directional [63]. In this optimization, each direction pair is associated with an importance weight proportional to its anisotropy. Additionally, to avoid degenerate elements at the border, we constrain the field to be orthogonal to the boundary on each face having a boundary edge. Eventually, we extract a quad mesh $\mathcal{M} = [V, E]$ from the smoothed cross-field using the approach [64]. We label its vertices $v_i \in V$ and its edges $e_j \in E$. We show an example of field processing and quad mesh extraction in Fig. 7.

We group chains of consecutive edges into chords. Chords can start at a boundary vertex and terminate at another boundary or an internal irregular vertex, or they can create inner loops. The chords are the basic data structure to assemble the additional layer: each chain of the main layer is duplicated and connected vertex by vertex. We refer to the chords of the additional layer as $c_k \in C$. Each chord has vertices $v_i' \in V'$, where the apostrophe denotes the correspondence to v_i. Similarly, the edges in the additional layer are marked as $e_j' \in E'$ for a given chord. The struts $e_s \in E_s$ link a v_i vertex to one or more v_i' vertices since multiple chords can be connected to a main layer vertex.

5.3. Initialization of the grid

We shift from a continuum to a discrete domain in the meshing step. We assume the quad mesh to be invariant during the assembly of the reinforced grid shell, and we move only the vertices of the additional layer. We base the shaping of the additional layer on the linear FE analysis of the main layer (i.e., the grid shell assembled from the quad mesh).

Main layer analysis

As a first step, we model the main layer $\mathcal{M}$ as a grid shell starting from the field-aligned quad mesh. In our formulation, a grid shell is a network of beam elements (mesh edges e_j) merging into structural nodes (mesh vertices v_i). The mesh faces are employed only to compute the incident weight on the nodes. Each edge is converted into a 2-node Euler–Bernoulli beam with uniform linear elastic isotropic material and six degrees of freedom per node (three translations and three rotations). Internally, the beams are connected mutually with rigid nodes, meaning that all incident beams share the same global displacement per node and cannot deform or rotate independently. We solve for the nodal displacements vector u to find the equilibrium under the hypotheses of linear static analysis, i.e., using the small displacement theory:

$$K \cdot u = f \tag{12}$$

In which K is the global stiffness matrix, and f is the vector of nodal forces. Any load, i.e., permanent load, has to be input in f. By default, we include in f the lumped weight of the structural and non-structural components. From u, we compute the internal beam forces (six quantities per endpoint): axial A, shear S_2, S_3, bending M_2, M_3,

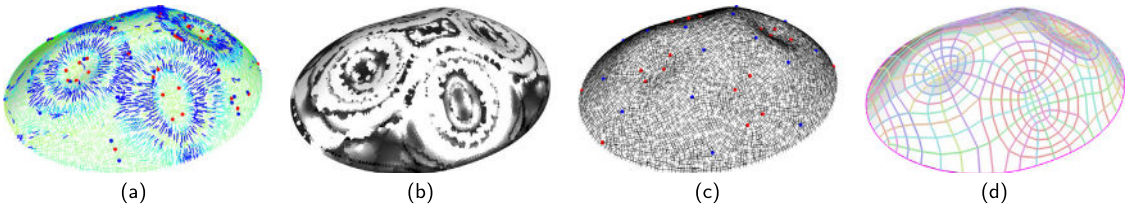

Fig. 7. Quad mesh pipeline: (a) initial cross-field (b) the anisotropy value; (c) cross-field resulting from the smoothing process; (d) final quadrangulation with chords highlighted.

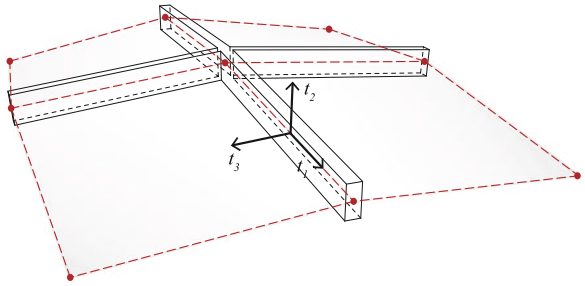

Fig. 8. Main layer's beam network with local reference system.

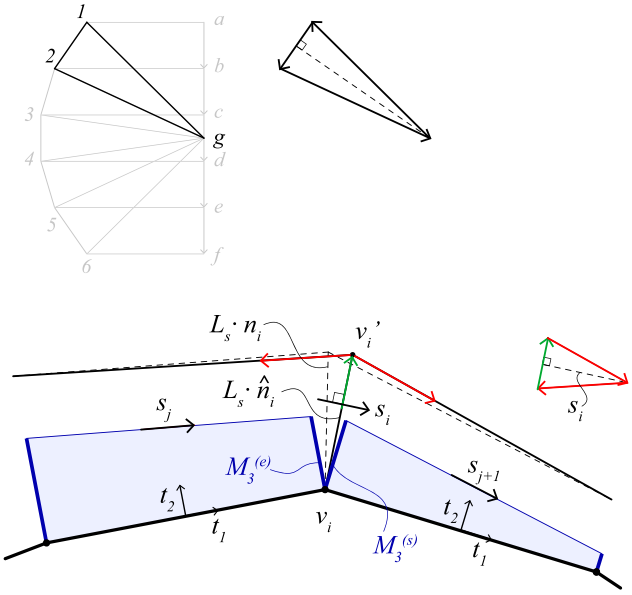

Fig. 9. Direction of the struts for a constant force on the chord: on top, node equilibrium for the first node of the truss of Fig. 3; on bottom, alignment of the strut with respect to M_3 bending moment (first component), and polygon of forces in equilibrium at node v_i'. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

torque T forces. These internal forces are referenced to the local frame $\langle t_1, t_2, t_3 \rangle$ of Fig. 8, in which t_1 is the beam axis direction, t_2 is aligned with the edge normal, and t_3 their cross product. We adopted standard sign conventions, e.g., positive bending for sagging and negative for hogging. A 2-node beam is a relatively coarse finite element; however, this discretization is accurate enough for beams submitted uniquely to nodal loads. Most importantly, we are interested only in the internal force values at the beams' endpoints, where the main layer is connected to the reinforcement.

Assembly of the starting additional layer

The reinforced grid shell $\mathcal{R} = [V \cup V', E \cup E' \cup E_s]$ is created sequentially, adding chords (vertices and edges) to the original mesh $\mathcal{M}$. The new chains of edges are obtained by duplicating the main layer beam chords. Then, each vertex of the new chord $v_i' \in V'$ is connected to its dual on the original layer $v_i \in V$ using a new edge $e_s \in E_s$ that will serve as a strut in the assembled structure. As a result, in the general case where multiple chords cross a vertex v_i, a distinct strut is created for each chord with a starting point in v_i.

This per-chord shaping of the additional layer is a paramount point in this work because, due to the complexity of the optimization and multiple degrees of freedom of the system, it allows us to restrict the number of variables. Indeed, in the optimization, we consider the chord shape as given and vary only its amplitude. Thus, a good initialization will limit the optimization problem to a smaller domain and speed up the computation.

We shape the struts adopting a per-chord logic according to two components: a first one, inspired by the 2D concepts in Section 3 and targeted on bending moment M_3, a second one, considering the joint presence of an in-plane bending M_2. In this shaping process, we keep the $\mathcal{R}$ structure fixed, except for the nodal coordinates of vertices $v_i' \in V'$, which are the only degrees of freedom.

First component. As a first step, we retrieve the bending M_3 at the endpoints of each beam of $\mathcal{M}$. For each vertex v_i on a chord, we compute the target distance of the two layers, namely the length of the strut $e'' = (v_i, v_i')$, similarly to Eq. (3), as:

$$L_s = \frac{M_3^i}{\bar{S}} = \frac{1}{2}\frac{M_3^{(s)} + M_3^{(e)}}{\bar{S}} \tag{13}$$

In which the bending M_3^i at node v_i is computed as the average of the endpoint bending of the pair of incident beams, i.e., start (s) and end (e). The allowed axial force is assumed as $\bar{S} = A_s \cdot f_k/\gamma$, where A_s is the cross-section area of the new layer, f_k is the characteristic material strength and γ is a safety factor. In our experiments we assume $\gamma = 3$.

In 2D graphic statics, a constant-force additional layer corresponds to a circular arrangement of the vectors in the force polygon (Fig. 3). The equilibrium at each node on the additional chord is the closed polygon in Fig. 9, in which, for 1 g and 2 g to be equal, the direction of the strut force is orthogonal to their difference.

On our main layer, we first compute, for each edge on a chord, the bending diagram of M_3, which we draw on the plane $t_3 = t_1 \times t_2$. For a 2-node beam type, the diagram is a line between two endpoint values, having a direction s_j (blue plot in Fig. 9).Since all s_j are unit vectors, they can represent the constant-force vectors we aim to have on the additional chord. Then, for every node on a chord v_i, we compute the normalized average direction s_i of the incident pair of s_j; and we project the per-vertex normal n_i to be orthogonal to s_i, yielding $\hat{n}_i$ and a redefinition of the struts as $L_s \cdot \hat{n}_i$. This computation takes place on the plane spanned by n_i and s_i, which represents a mean M_3^i plane. In a 3D configuration, this is a crucial step since correctly aligning the resulting force on the additional chord with the strut's directions avoids extra bending and reduces the chances of instability. Note that, if using the per-vertex normal (dashed lines in Fig. 9), the polygon of forces would not be closed (so bending forces are required for equilibrium).

Second component. In a bending-dominated grid shell, the beams undergo various 3D bending stress combining T, M_2, and M_3. In our problem, torsion T on the main layer is either negligible because

loads are applied only at the nodes or will be significantly reduced by delivering properly-shaped reinforcements in the orthogonal direction. However, considering only the out-of-plane bending M_3 in shaping the additional layer (as in) is insufficient. Therefore, for each beam endpoint, we consider a per-chord biaxial bending out of M_2 and M_3.

We place the struts relying on some basic solid mechanics ideas. In the 2D examples in Section 3, the load plane (including forces and reactions) is aligned with the principal axes of the cross-section $\langle \xi, \eta \rangle$ so that the load axis $\underline{s}$ is always orthogonal to the neutral axis of the beam $\underline{n}$ for each beam section. The load axis is the trace of the load plane, namely the intersection of the plane orthogonal to the bending moment direction and the plane through the cross-section. The neutral axis highlights a line of zero stress due to bending on the cross-section: the remaining parts of the cross-section either shrink or elongate. Consequently, the beam bends in the direction of loads, i.e., in the load plane. In this setup, if the additional layer lies on the load plane, it does not exert any other bending force.

In 3D, if an oblique bending moment $M = M_2 + M_3$ is acting (Fig. 10), the neutral axis $\underline{n}$ is rotated and does not coincide with a principal axis. Except for special cases (e.g., circular cross sections), the load axis $\underline{s}$ is no longer orthogonal to the neutral axis. Consequently, the beam does not deflect in the loading direction but along the deflection plane. In light of this 3D phenomenon, our idea is to arrange the struts on the deflection plane, thus realizing a more efficient reinforcement layer. To compute the trace of the deflection plane $\underline{f}$, we recall the property of $\underline{s}$ and $\underline{n}$ to be conjugate axes. Therefore, the following relation holds:

$$tan\beta = -\frac{M_3}{M_2} \cdot \frac{I_2}{I_3} \tag{14}$$

The angle β is evaluated between t_2 and $\underline{f}$. As defined in , the struts lie on the plane orthogonal to t_3. So, a β rotation will move them on the deflection plane. Given a chord of the main layer, we compute the β_i for each v_i vertex of the chord as the average β computed at endpoints of the pair of incident beams. Then, we rotate $\hat{n_i}$ around the average of t_1 vectors of pairs, yielding $\check{n_i}$.

Exceptions are the points that lie on the border and on which boundary conditions are applied. Hence, for design reasons, we do not adopt $\check{n_i}$ directions on those points but only $\hat{n_i}$ because it is more common for external bodies providing restraints (walls, beams, ground) to be aligned to the principal axes of the boundary beams. Therefore, for each v_i the struts evaluate:

$$-L_s \cdot \hat{n_i} \qquad \text{at the boundary nodes} \tag{15}$$

$$-L_s \cdot \check{n_i} \qquad \text{otherwise} \tag{16}$$

The entire formulation is based on the frame in Fig. 8 and standard force sign convention. Therefore, aiming to have an additional layer with tensioned chords and compressed struts, a negative sign must be added to generate the element on the right side of the surface. The new elements of the mesh (chords and struts) are converted into structural 2-node beams for the main layer.

5.4. Optimization

Given an input quad-dominant mesh $\mathcal{M}$ and the accompanying set of chords, we seek an optimal reinforced grid shell $\mathcal{R}$ that minimizes the compliance of the structure or total strain energy $\mathcal{E}$ for a given design load. In this problem, we assume the per-chord height multipliers x_k as variables, namely a vector of scalar values. Each scalar can scale the strut lengths L_s independently per chord.

For generality, some parameters $\bar{p}$ are constant throughout the routine, such as the cross section characteristics of the main and the additional layer, which the designer inputs. Thus, the problem can work for any input structure. Similarly, the struts' cross section is pre-selected for having stiff elements to deform as little as possible and withstand likely buckling phenomena. The choice of not refining by

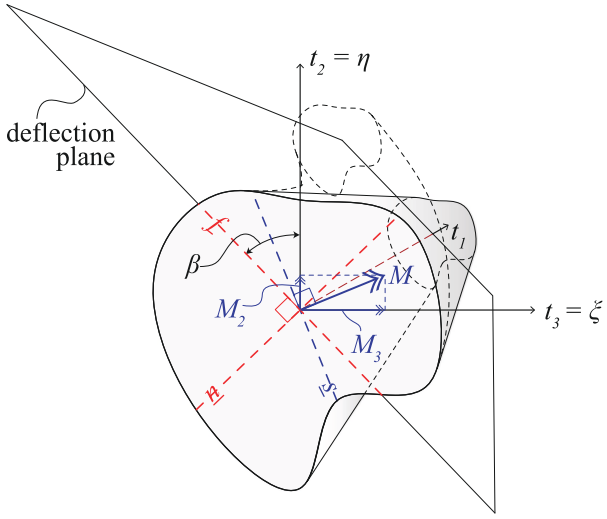

Fig. 10. Oblique bending on a generic beam: computation of the deflection plane, its trace on the cross section $\underline{f}$ and its angular distance β from the axis t_2. The struts of our additional layer are rotated to lie on the deflection plane.

sizing the structure does not prevent strength issues that it may still exhibit even after the optimization, but it reflects the usual practice in which custom actions may be undertaken to fulfill local strength demands. Once the solution is initialized as per Section 5.3, we solve the following bounded minimization problem.

$$\min_{x_k} \ \mathcal{E}(\mathcal{R}(\bar{p}), x_k) \tag{17}$$

$$\text{s.t.} \quad x_{min} \leq x_k \leq x_{max} \tag{18}$$

The equilibrium of the structure:

$$K(\bar{p}, x_k) \cdot u = f \tag{19}$$

is inherently guaranteed at each iteration as it is required to compute the strain energy:

$$\mathcal{E} = \frac{1}{2} \int_{\mathcal{L}} \frac{A^2}{EA} + \frac{S_2^2}{GA} + \frac{S_3^2}{GA} + \frac{M_3^2}{EI_3} + \frac{M_2^2}{EI_2} + \frac{T^2}{GJ} dx \tag{20}$$

$$= \frac{1}{2} \sum_j \sum_o \frac{\ell_q}{2} \left[\frac{A_{oq}^2}{EA} + \frac{S_{2,oq}^2}{GA} + \frac{S_{3,oq}^2}{GA} + \frac{M_{3,oq}^2}{EI_3} + \frac{M_{2,oq}^2}{EI_2} + \frac{T_{oq}^2}{GJ} \right] \tag{21}$$

The general formulation of the strain energy over the total length of the beams $\mathcal{L} = \sum_q \ell_q$ (Eq. (20)) can be simplified for 2-node beams in Eq. (21), in which the internal forces are computed only at the endpoints $o \in \{start, end\}$. Accordingly, for each beam q (where q includes all beams of our structures), the discretized strain energy is constant over the half-length of the beam.

In Eq. (19), the nodal force vector $f = f_{ext} + f(\bar{p}, x_k)$ is composed of a fixed component f_{ext} due to the load applied on the nodes of the main layer (i.e. cladding, snow, wind), and of a variable component that includes the own weight of the structural elements and is updated at each step based on the variables state.

We adopt filtering criteria during the optimization to integrate some basic user-defined preferences: the out-of-plane height can be constrained not to exceed (i) a maximum value $L_{s,max}$; and (ii) a minimum value $L_{s,min}$. In practice, $L_{s,max}$ defines the maximum size of the structure, considering that the struts can span on the inside, where the essential functions of the grid shell should not be hindered, or on the outside, where the surrounding context may pose other constraints. If $L_{s,max}$ is defined by the user, the x_{max} is scaled to allow the struts to reach $L_{s,max}$ length at most.

The value $L_{s,min}$ defines the size of the smallest fabricable node equipped with a strut. The strut is required to have a minimal height

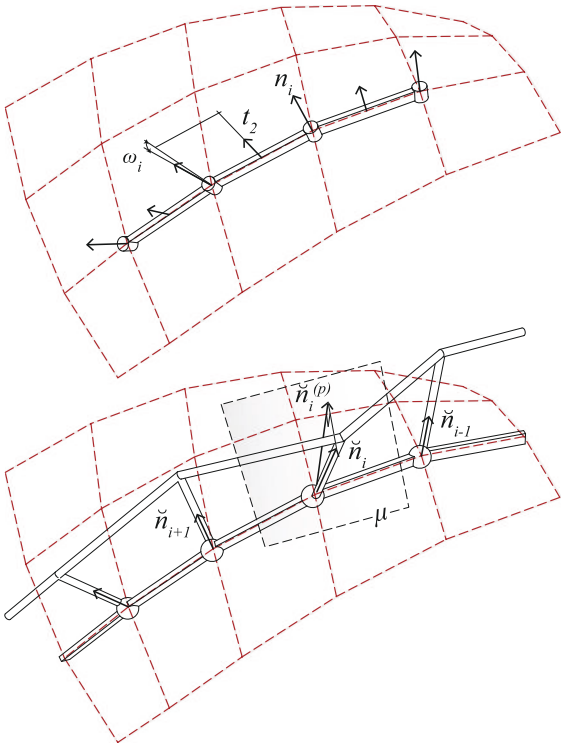

Fig. 11. Geometric torsion ω in a grid shell (on top), in our case (on bottom). Vectors have different lengths only for visualization purposes but are all unitary.

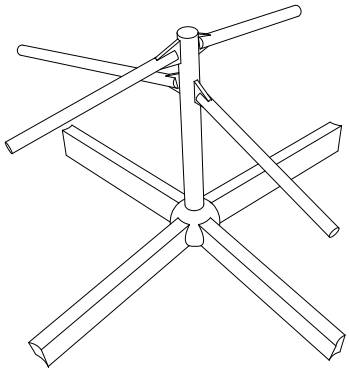

Fig. 12. Example of two crossing chords jointed on a single strut.

to avoid intersecting volumes (keep the beams separated). The value of $L_{s,min}$ is user-defined as it depends on the chosen node detail and fabrication technology. Struts shorter than $L_{s,min}$ are removed, and their endpoints are collapsed on the main layer. Beam elements of the additional layer that are delimited by two collapsed struts are removed as well, leaving only the beam of the main layer. The $L_{s,min}$ filtering applies at each optimization step.

5.5. Post-processing

Once the variables x_k have been optimized, local refinement steps are applied in a post-processing stage. The refinements are either structural to improve the flow of forces locally or practical, avoiding intersecting elements or considering their manufacturing.

Reducing geometric torsion

Beams are oriented elements, so they respond differently if the plane of loading changes. The principal axes of inertia describe this attitude.

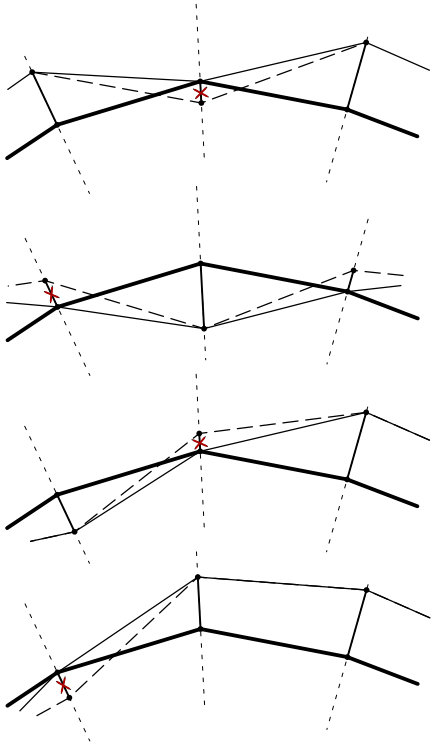

Fig. 13. Filtering cases to avoid intersections between the two layers, with the collapse of the shortest strut.

In a free-form grid shell, to obtain the maximum stiffness of the beams, it is common practice to orient the beam (principal) axes to lie on the edge frames [65], i.e. perpendicular to the shell surface (Fig. 11 top). In the general case, this configuration induces twist ω of the beams in the node, whose magnitude depends locally on the grid connectivity and size, and on the surface curvature. Consequently, beams pay extra stress, and nodes with geometric torsion result in larger and more expensive details (p.208 [66]). Torsion-free nodes ($\omega = 0$ for every incident beam) are obtained only in special cases, like in PQ meshes and conical meshes [67] or PP meshes [39]. Reducing geometric torsion positively influences aesthetics, load-bearing characteristics and costs.

In our case, we assemble a composite cross section made of the main-layer beam and the additional-layer beam. Both beams are straight and show their own geometric torsion to the mesh. However, by design, the struts can have any direction, so the composite profile could be highly twisted, becoming the primary source of torsion. It can be observed from the parallel axis theorem that, locally, the larger the strut length L_s, the more the principal axes of the composed beam tend to align to the plane through the struts. Consequently, the more the geometric torsion of the composite profile will be the main issue.

Referring to Fig. 11 (bottom), we define a local per-chord iterative optimization to move the struts directions $\check{n}_i$ to minimize the geometric torsion. For each vertex on a chord v_i (excluding boundary nodes), we compute the interpolating plane μ through the struts $\check{n}_{i-1}$ and $\check{n}_{i+1}$, on which we project the $\check{n}_i$ vector. The obtained vector $\check{n}_i^{(p)}$ is the direction that locally minimizes the geometric torsion on the chord. At each optimization step t, we derive the new strut direction as a weighted sum:

$$\check{n}_i^{(t)} = w_1 \cdot \check{n}_i + (1 - w_1) \cdot \check{n}_i^{(p)} \tag{22}$$

in which $w_1 = 1$ for $L_s \leq L_{min}$ and decrease linearly to 0 for a fixed L_{max}. Please note that these limitations differ from the $L_{s,max}$ and $L_{s,min}$ used for computing the weight w_1. The L_{min} value defines the threshold below which optimizing for the composite cross section is pointless

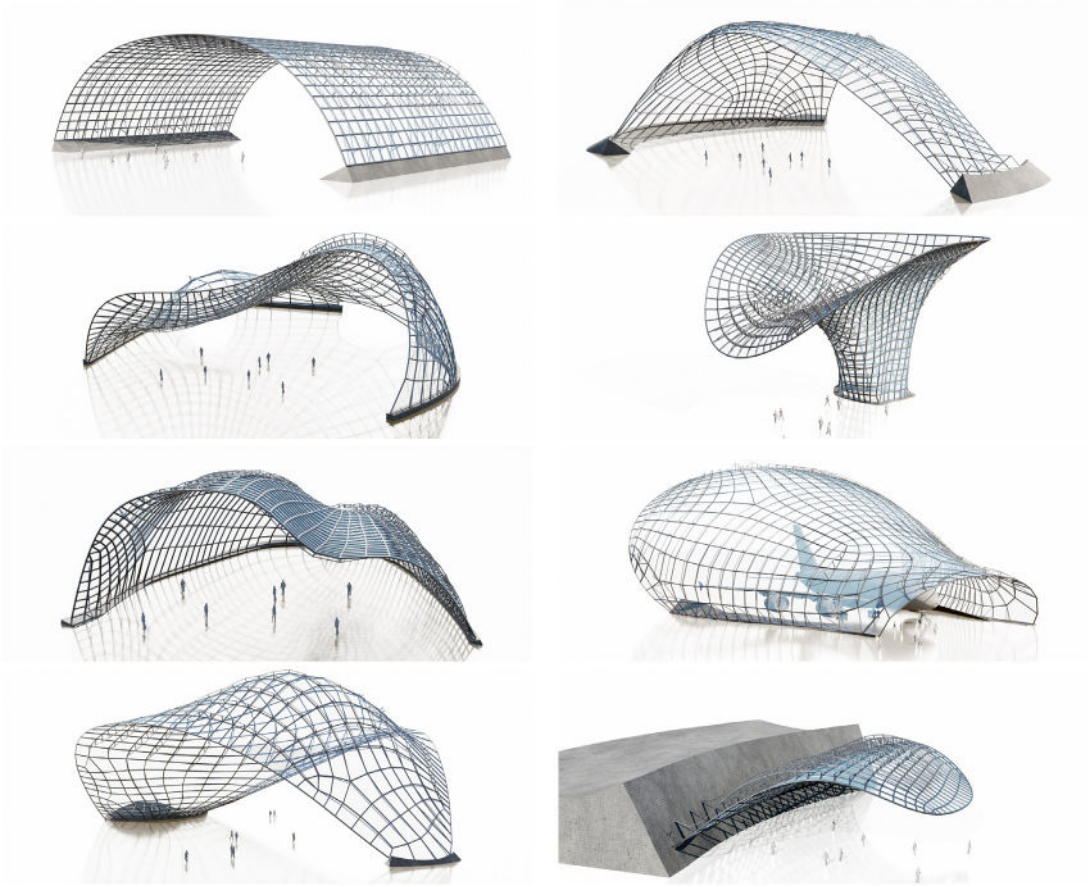

Fig. 14. Results of our method on different models with a multi-parameter setup (*tsf* field), from top left: Arch, Dolphin, Envelope, Flower, Hall, Hangar, Manta, Roof.

since the single beams have a predominant inertia contribution. The L_{max} value determines the speed of this 'smoothing' step: if reasonably high, the algorithm safely reaches convergence without distorting the grid. The optimization is stopped once the maximum move is below a threshold.

In all our examples, we adopt a solid circular cross section for the additional layer beams because of two reasons: first, it is an ideal cross section for elements such as the additional beams that are supposed to bear a prevalent axial force in tension; second, all axes in these beams are principal of inertia, so these beams merge locally into torsion-free nodes. As a consequence, the chords of the additional layer will not have geometric torsion.

Aligning per-node struts directions

A typical scenario for our vertex v_i is to be a starting node of h distinct struts for h crossing chords (mostly $h = 2$). While it might be optimal for the global statics of the structure, such nodes are practically unfeasible. The reason is that the elements have a determined size, so close struts usually have intersecting volumes. Therefore, in the post-processing stage, all struts incident to a node are aligned on a weighted average direction, in which longer struts weigh more. From a manufacturing perspective, this implies the creation of multiple joints along the strut if two or more chords lie on the same side of the original surface as in Fig. 12.

Beams' intersection filtering

The most intriguing feature of this method is the positioning of the additional layer inward or outward on the main surface. Practical issues arise where the struts at the endpoints of a general edge of the main layer have opposite directions with respect to the main surface, or both are shorter than $L_{s,min}$. In these cases, intersections or a complete overlap occur between additional and main layer beams. We identify the recurring cases shown in dashed lines in Fig. 13, in which we apply corrective actions represented as continuous lines.

6. Implementation details and dataset

We implemented a software pipeline for the proposed method. The input shell surfaces are imported into the FEM package [68] and discretized into triangular plates with uniform thickness and material properties. Each model employs a linear elastic isotropic material with the characteristics of structural steel (Young's modulus $E = 210$ GPa; Poisson ratio $\nu = 0.3$) and a sufficiently-large thickness $t = 0.2$ m. Any material could be used since, for linearity, the stress is proportional. The models can have any boundary condition, i.e., pin-supported (all translations fixed) or fully-restrained (all translations and all rotations fixed) nodes. The own weight of the structure simulates a uniformly distributed load, and its effect is again linear.

The field derivation is implemented with Matlab, while all other steps (field processing, quad meshing, grid generation, optimization and post-processing) are implemented in C++. For the quad meshing, we used the approach in [64] as it allows more precise local control of edge flow than patch-based methods [69]. The actual optimization (Section 5.4) is performed using the NLopt library [70] using the derivative-free algorithm Subplex [71]. To comply with the chosen optimization strategy, we employ a simulation-in-the-loop approach, computing a full linear static simulation of the reinforced grid shell (obtained as a function of the optimization variables) for each objective function evaluation requested by the algorithm. Given the complex relation between the optimization variables and the objective function, the optimization space is highly non-convex and presents many different local minima. To increase the chances of reaching the global

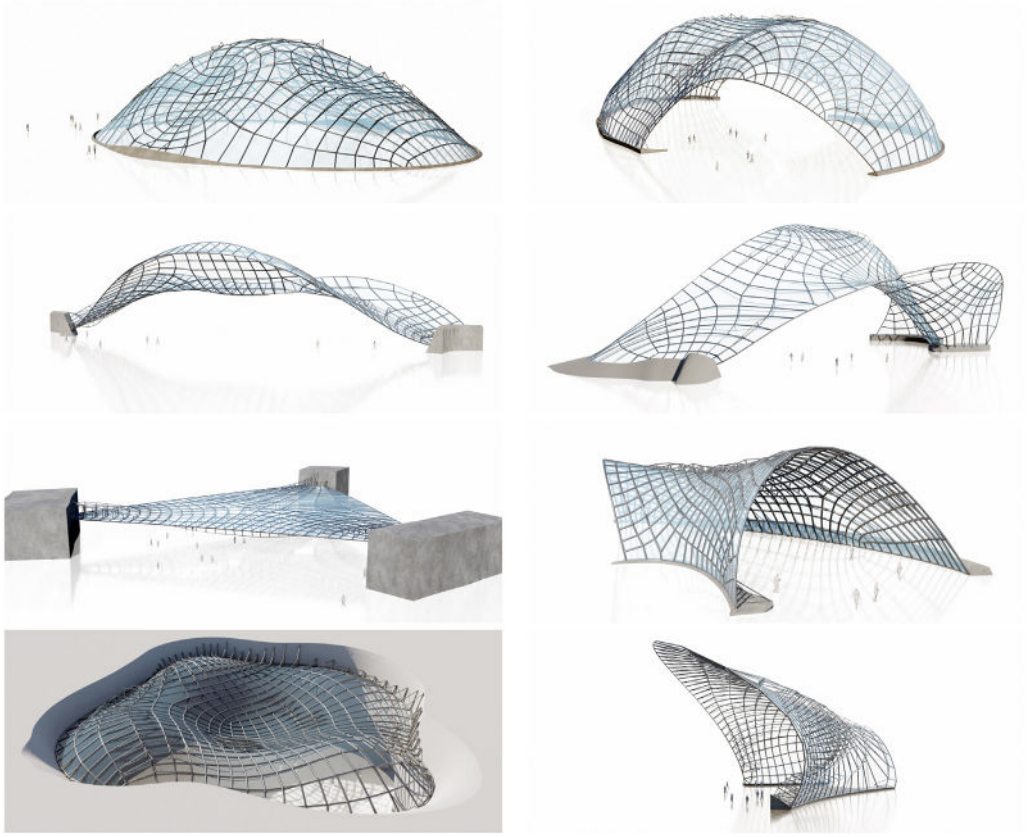

Fig. 15. Results of our method on different models with a multi-parameter setup (*tsf* field), from top left: Rust, Shell1, Shell2, Taut, TriSlab, Vault, Wave, Yarn.

optimum, we adopt a two-step optimization strategy: in the first run, we optimize for a single variable and assign it to the x_k for all chords; in the second step, we use it as a starting point the optimum found in the first run and optimize for all variables x_k independently.

User inputs are material and cross-section parameters for the main layer and the additional layer ($\bar{p}$), the load scenario, the bounds of the optimization variables, and the post-processing manufacturing limitations. By default, the upper bound of the height multiplier x_{max} is controlled by the $L_{s,max} = 5$ m, which is the value used in all our experiments.

We tested our method on a dataset of large-spanned shells (Figs. 14, 15) that exhibit different bending behavior. Some statistics are reported in Tables 1 and 2. A large subset of the tested examples is not stiff enough to support the load as grid shells with regular cross-sections. These examples are marked in red as they are unfeasible. Instead, some examples are similar to a proper funicular shell, so they are stiffer. We used as an external load f_{ext} a uniform distribution of 2 kN/m^2 in the gravity direction, representing a typical service load including non-structural load (e.g., cladding) and environmental load (e.g., snow). We distribute it among the main layer nodes in proportion to their Voronoi area. All elements are made of structural steel, their own weight is computed from a material density of $\rho = 78.5$ kN/m^3. The cross-section of the elements is circular hollow sections of 0.2 m external diameter and 0.025 m thickness for the main beams; solid circular sections of 0.1 m diameter for the added layer; solid circular sections 0.15 m diameter for the struts. For convenience, we have run our experiments in parallel on a server machine equipped with an AMD Epyc 7413 processor and 128 GB RAM. However, the performances for a single model are comparable to the ones obtained on an Apple laptop with an M1 processor. The optimization is the most demanding computation within our pipeline, which takes 5 min to 7 h, depending on the grid shell complexity. For each tested case, we compare the values of the objective function, i.e., the strain energy: (i) $\mathcal{E}_{\mathcal{M}}$ obtained from the main layer analysis as a grid shell built upon the quad mesh; (ii) $\mathcal{E}_{\mathcal{R}0}$ obtained from the initialized reinforced grid shell; and (iii) $\mathcal{E}_{\mathcal{R}opt}$ obtained from the optimized reinforced grid shell.

7. Results and discussion

We validate our method on different levels. First, we run the full algorithm in a multi-chord setup, obtaining $\mathcal{E}_{\mathcal{R}opt}$, and we discuss the advantages of our strategy in comparison with the un-reinforced grid shell. In one scenario, we adopt a grid shell with the same topology and uniform cross section, corresponding to our structures' main layer. Then, in another scenario, we submit those un-reinforced grid shells to a cross section optimization targeting a similar stiffness achieved by our structures.

Second, we test our method using different input fields: *tsf* that is specifically developed for our method; curvature field *curv*, and bending field from the thin shell theory *bend*, which have been used in other works on reinforced structures. For all these cases, the quadrangulation is performed as per Section 5.2 to have quad meshes with the same total edge length, meaning to employ the same amount of structural material on the main layer. To make the comparison fair, we optimize for a unique multiplier x_k for all chords, obtaining $\mathcal{E}'_{\mathcal{R}opt}$.

Third, we discuss the strategies adopted in our pipeline and test the accuracy of our results with those obtained with a commercial nonlinear solver.

7.1. Comparison with grid shells

We compare our reinforced grid shells and the corresponding un-reinforced grid shells. As a first experimental scenario, we adopt the same grid topology and the same uniform cross section. Table 3 shows that, in all cases, although our structures are heavier than grid shells due to the added structural material of the new layer, their performance

Table 1
Results of our method on different models in a single parameter setup: model name (see Figs. 14, 15); size of the axis aligned bounding box in meters; fields adopted to generate the quad mesh (*bend*, *curv*, *tsf*); number of vertices i; number of edges j; number of chords k; total length of the beams $\mathcal{L}_j = \sum_j \ell_j$ of the main layer $\mathcal{M}$; total strain energy of the simple grid shell $\mathcal{E}_{\mathcal{M}}$, of the reinforced grid shell before $\mathcal{E}_{\mathcal{R}0}$ and after the optimization $\mathcal{E}'_{\mathcal{R}opt}$; energy reduction; optimization time. Red values denote insufficient stiffness.

Model	Size (m × m × m)	Field	i	j	k	$\mathcal{L}_j$	$\mathcal{E}_{\mathcal{M}}$ (kJ)	$\mathcal{E}_{\mathcal{R}0}$ (kJ)	Cut (%)	$\mathcal{E}'_{\mathcal{R}opt}$ (kJ)	Cut (%)	Time (s)
Arch	59.64 × 40.00 × 15.88	bend	943	1822	60	3303.3	15 359.4	290.4	98.1	254.8	12.3	6.6
		curv	943	1822	60	3303.3	15 356.8	290.3	98.1	254.2	12.4	6.6
		tsf	943	1822	60	3303.4	15 365.0	275.9	98.2	211.2	23.5	15.5
Dolphin	75.76 × 53.24 × 25.23	bend	545	1038	49	2578.1	4192.9	768.5	81.7	120.3	84.4	8.2
		curv	561	1070	51	2620.2	3022.3	1054.4	65.1	273.7	74.0	6.5
		tsf	527	1002	67	2549.6	2829.6	2475.4	12.5	277.2	88.8	5.9
Envelope	57.06 × 55.71 × 20.24	bend	681	1306	52	2865.9	29587.1	528.7	98.2	132.0	75.0	9.9
		curv	622	1192	60	2793.1	29950.9	296.2	99.0	183.9	37.9	9.6
		tsf	610	1172	71	2743.6	38343.8	451.9	98.8	161.9	64.2	13.3
Flower	32.11 × 34.49 × 23.21	bend	947	1850	61	2438.3	3182.1	1055.9	66.8	361.2	65.8	14.9
		curv	944	1840	69	2413.7	3384.6	1244.0	63.2	225.9	81.8	15.6
		tsf	1020	1989	70	2506.0	4499.4	928.1	79.4	129.6	86.0	19.3
Hall	39.74 × 38.58 × 13.54	bend	868	1684	61	2351.8	3443.3	51.0	98.5	40.9	19.8	10.3
		curv	858	1665	56	2345.6	2938.7	66.6	97.7	34.0	49.0	9.1
		tsf	848	1645	56	2334.7	2544.0	308.8	87.9	23.1	92.5	12.5
Hangar	78.28 × 83.28 × 35.36	bend	704	1353	59	4469.9	338833.7	6768.9	98.0	1358.7	79.9	19.3
		curv	694	1341	50	4419.8	307870.3	3168.1	99.0	1828.1	42.3	9.2
		tsf	676	1297	72	4387.8	382361.2	2662.2	99.3	1812.1	31.9	12.2
Manta	68.38 × 84.92 × 28.44	bend	536	1032	39	3221.1	82330.3	5154.1	93.7	2311.6	55.2	11.3
		curv	501	965	41	3171.7	154422.4	8165.5	94.7	6317.4	22.6	7.8
		tsf	506	966	61	3147.4	60714.5	4717.9	92.2	3677.4	22.1	7.5
Roof	29.75 × 43.90 × 8.51	bend	476	910	42	1376.4	140683.1	445.8	99.7	282.6	36.6	6.0
		curv	490	941	42	1400.7	158445.5	254.2	99.8	316.9	−24.6	5.5
		tsf	468	893	44	1359.0	129656.3	173.0	99.9	97.0	43.9	6.2
Rust	80.35 × 58.08 × 19.15	bend	576	1124	36	3303.9	2367.1	1545.8	34.7	276.9	82.1	11.5
		curv	491	966	54	3105.3	1516.3	1383.9	8.7	550.8	60.2	7.0
		tsf	615	1194	69	3378.4	1218.5	1133.1	7.0	156.1	86.2	6.6
Shell1	51.11 × 81.06 × 23.56	bend	687	1324	49	3557.6	1865.7	1721.9	7.7	335.4	80.5	5.9
		curv	749	1449	62	3792.7	1829.1	1686.1	7.8	343.8	79.6	7.0
		tsf	726	1398	76	3683.0	1078.5	915.5	15.1	198.1	78.4	8.5
Shell2	90.59 × 37.08 × 18.33	bend	372	703	41	1863.0	8724.4	4422.4	49.3	2431.0	45.0	2.4
		curv	340	651	29	1828.0	11 635.8	2838.3	75.6	1589.2	44.0	4.8
		tsf	379	707	61	1892.5	8997.4	4163.9	53.7	829.2	80.1	3.9
Taut	42.38 × 89.33 × 26.73	bend	481	915	48	2544.2	4415.3	2574.1	41.7	415.9	83.8	5.8
		curv	499	954	48	2631.1	5939.9	3351.4	43.6	342.1	89.8	6.0
		tsf	492	926	69	2571.8	4059.2	2532.7	37.6	466.5	81.6	5.0
TriSlab	80.17 × 69.54 × 0.50	bend	502	962	41	2550.0	1278111.1	42 475.3	96.7	41 547.2	2.2	11.9
		curv	491	937	54	2495.0	994623.5	7094.5	99.3	4036.1	43.1	11.0
		tsf	452	853	64	2426.5	384656.6	19 886.8	94.8	3211.2	83.9	5.8
Vault	42.75 × 23.32 × 12.37	bend	625	1200	46	1469.0	65.2	65.2	0.0	6.5	90.1	7.5
		curv	575	1103	44	1410.8	66.4	66.4	0.0	7.2	89.1	4.7
		tsf	619	1184	59	1467.9	49.4	49.4	0.0	4.6	90.7	7.9
Wave	43.44 × 44.01 × 6.05	bend	556	1074	43	1777.0	1411.2	1221.3	13.5	22.0	98.2	6.2
		curv	616	1200	43	1863.6	1059.8	679.4	35.9	35.8	94.7	5.8
		tsf	589	1138	46	1814.4	552.8	541.9	2.0	17.1	96.8	7.7
Yarn	51.28 × 71.53 × 26.18	bend	401	767	43	2207.1	76743.9	2020.1	97.4	1940.2	4.0	4.8
		curv	442	841	44	2341.7	37716.4	2031.6	94.6	804.6	60.4	3.5
		tsf	431	818	43	2344.0	47115.3	1483.3	96.9	352.1	76.3	7.3

is significantly better. The data neglects the volume of nodes that most likely works against reinforced grid shells, having more nodes.

It must be pointed out that in most of the cases highlighted in red in Tables 1, and 2 the objective function $\mathcal{E}_{\mathcal{M}}$ shall not be considered accurate since the structure undergoes large displacement, and the linear solver produces non-realistic results. This phenomenon does not invalidate our pipeline since the same outcome (from grid shell linear analysis) initializes the additional layer geometry. Indeed, the linear assumption allows us to scale these quantities and obtain proper sizing. Red-highlighted structures are not stiff enough in the form of grid shells. Thus, adopting a reinforced grid shell gives maximum benefit as unfeasible shapes can be turned into robust structures (see $\mathcal{E}'_{\mathcal{R}opt}$ in Table 1 and $d'_{\mathcal{R}opt}$ in Table 2). The other cases have good shape stiffness as grid shell, yet the corresponding reinforced structure outperforms it. As expected, the strut heights are smaller on average in these cases than in the red-highlighted cases.

The efficiency of the solution lies in the better use of the axial stiffness of the structural elements: in the grid shell, most of the strain energy is caused by bending and shear, while if reinforced, the contribution of axial prevails, as evidenced in Table 2. This outcome confirms the capacity of our method to balance bending more efficiently through axial forces, e.g., in the cases of Vault, the axial contribution on $\mathcal{E}_{\mathcal{M}}$ is 39.5% while is close to 70% on $\mathcal{E}_{\mathcal{R}opt'}$; in the case of Shell1 it is 20.7% and 58.7%, respectively. A comparison of maximum displacement (reported in Table 2) provides a better understanding of the high stiffness that all examples achieve after equipping them with an additional layer. From a visual standpoint, the reinforced grid shells do not compromise the aesthetics of the main surface, and the additional layer is barely

Table 2
Results of our method on different models in a multiple parameter setup: model name (see Figs. 14, 15); fields adopted to generate the quad mesh (*bend*, *curv*, *tsf*); total strain energy on the optimized reinforced grid shell in single $\mathcal{E}'_{Ropt}$ (from Table 1) and multi parameter setup $\mathcal{E}_{Ropt}$; reduction; optimization time; percentage of axial forces (first term in Eq. (21)) in the strain energy for all setups $\mathcal{E}^{ax}/\mathcal{E}$; maximum norm of the displacements in all setups d. Red values denote insufficient stiffness.

Model	Field	$\mathcal{E}'_{Ropt}$ (kJ)	$\mathcal{E}_{Ropt}$ (kJ)	Cut (%)	Time (s)	$\mathcal{E}^{ax}_{\mathcal{M}}/\mathcal{E}_{\mathcal{M}}$ (%)	$\mathcal{E}'^{ax}_{Ropt}/\mathcal{E}'_{Ropt}$ (%)	$\mathcal{E}^{ax}_{Ropt}/\mathcal{E}_{Ropt}$ (%)	$d_{\mathcal{M}}$ (m)	$d_{Ropt'}$ (m)	d_{Ropt} (m)
Arch	bend	254.8	154.1	39.5	1847.4	0.8	60.2	55.4	3.30	0.10	0.09
	curv	254.2	163.3	35.8	2878.0	0.8	60.1	53.3	3.30	0.10	0.05
	tsf	211.2	141.6	32.9	2589.6	0.8	62.6	52.9	3.30	0.07	0.07
Dolphin	bend	120.3	94.9	21.1	5246.2	6.4	73.1	73.0	3.37	0.04	0.03
	curv	273.7	218.5	20.2	630.4	9.9	52.5	56.7	2.44	0.15	0.10
	tsf	277.2	176.1	36.5	4002.1	23.4	55.2	60.7	2.15	0.17	0.09
Envelope	bend	132.0	97.8	25.9	5289.0	2.7	47.1	52.3	15.70	0.13	0.10
	curv	183.9	133.5	27.4	3709.4	2.6	48.8	52.6	16.13	0.07	0.04
	tsf	161.9	113.3	30.0	7903.9	4.6	51.6	51.3	16.05	0.06	0.05
Flower	bend	361.2	115.8	67.9	11 316.7	9.1	36.6	41.9	2.03	0.42	0.10
	curv	225.9	96.7	57.2	12 348.1	6.5	45.3	43.4	2.60	0.34	0.14
	tsf	129.6	74.1	42.9	27 163.1	8.7	39.9	44.0	2.76	0.11	0.06
Hall	bend	40.9	19.8	51.6	1518.2	2.8	34.2	44.8	1.91	0.03	0.01
	curv	34.0	19.8	41.8	5831.8	4.3	35.3	38.7	1.66	0.02	0.02
	tsf	23.1	13.2	43.1	8600.7	9.7	43.7	48.3	1.52	0.02	0.01
Hangar	bend	1358.7	1017.1	25.1	5495.3	0.9	52.9	56.6	71.96	0.18	0.20
	curv	1828.1	1196.5	34.5	1193.7	0.5	34.8	40.6	64.79	1.01	0.31
	tsf	1812.1	1174.8	35.2	7433.0	1.4	32.3	39.5	69.88	0.34	0.27
Manta	bend	2311.6	1418.4	38.6	458.1	3.6	45.5	43.4	18.07	1.13	0.64
	curv	6317.4	4764.0	24.6	306.8	4.1	46.5	43.6	32.42	3.63	2.50
	tsf	3677.4	2099.8	42.9	1484.7	9.2	43.0	38.1	13.97	1.78	1.07
Roof	bend	282.6	177.6	37.1	662.5	3.4	27.8	25.1	62.52	0.47	0.28
	curv	316.9	254.1	19.8	351.1	2.9	31.9	34.2	65.49	0.50	0.39
	tsf	97.0	88.6	8.7	394.9	2.9	32.7	30.2	60.00	0.13	0.13
Rust	bend	276.9	124.3	55.1	2055.1	7.6	66.1	65.9	0.64	0.09	0.04
	curv	550.8	328.7	40.3	3777.4	41.9	64.6	62.7	0.32	0.15	0.14
	tsf	156.1	82.3	47.3	5037.0	40.1	64.3	67.1	0.29	0.06	0.02
Shell1	bend	335.4	253.9	24.3	3569.8	8.3	39.3	42.5	1.07	0.14	0.09
	curv	343.8	232.2	32.5	13 440.4	22.0	50.5	52.2	0.57	0.08	0.06
	tsf	198.1	140.5	29.1	9430.5	20.7	58.7	66.3	0.68	0.06	0.05
Shell2	bend	2431.0	2030.7	16.5	301.4	6.6	34.6	36.7	2.61	1.50	1.25
	curv	1589.2	1096.7	31.0	196.2	8.7	47.0	41.0	3.30	0.94	1.15
	tsf	829.2	554.0	33.2	1327.2	9.0	42.6	42.5	2.52	0.43	0.25
Taut	bend	415.9	295.6	28.9	541.0	7.5	52.0	51.9	3.89	0.49	0.31
	curv	342.1	297.6	13.0	864.8	6.5	57.7	56.4	4.37	0.26	0.22
	tsf	466.5	326.5	30.0	3048.3	16.8	52.0	52.1	3.42	0.27	0.14
TriSlab	bend	41 547.2	20 517.6	50.6	376.6	40.6	44.0	28.1	118.46	6.37	4.41
	curv	4036.1	2975.9	26.3	812.2	36.8	29.4	31.7	99.88	1.27	1.25
	tsf	3211.2	2422.3	24.6	441.7	76.1	27.5	31.0	39.36	1.41	1.10
Vault	bend	6.5	5.0	23.2	2005.1	12.6	53.4	60.4	0.07	0.01	0.01
	curv	7.2	5.5	23.5	505.3	11.9	56.6	68.0	0.08	0.01	0.01
	tsf	4.6	4.1	11.3	5365.3	39.6	69.7	68.7	0.06	0.01	0.01
Wave	bend	22.0	13.4	38.9	4290.1	9.0	48.1	55.9	0.49	0.01	0.01
	curv	35.8	23.1	35.3	2943.6	21.7	40.6	48.5	0.38	0.02	0.01
	tsf	17.1	11.2	34.8	1461.9	51.8	54.4	53.6	0.16	0.01	0.00
Yarn	bend	1940.2	1127.2	41.9	1822.4	3.8	28.8	33.7	22.60	1.45	0.38
	curv	804.6	314.8	60.9	2132.3	3.7	33.1	36.4	13.90	0.69	0.15
	tsf	352.1	273.9	22.2	797.1	3.9	53.6	47.2	18.82	0.35	0.17

visible from a distance. While from a closer perspective, it creates nice effects going smoothly in and out of the surface (Fig. 16).

In the second experimental scenario, we optimize the cross sections of the grid shell adopted for comparison using the section optimizer tool of Karamba3D [72]. In this process, we target the maximum displacement of our structures (last column of Table 2) and use the entire set of commercial EN10210-2 cross sections (from CHS 21.3 × 2.3 mm to 1219.0 × 25.0 mm). In cases with high bending (like Dolphin, Envelope, Hall, Roof) the optimized grid shell requires large cross sections with a consequent higher volume of steel with respect to our solution $V_{\mathcal{M}opt}$ (in Table 3). In such instances, the displacement goal is not achieved because it would necessitate cross sections even bigger than the maximum in the set. In some cases (like Vault, Wave), the optimized grid shells and our reinforced grid shells have a comparable volume. Elsewhere, the optimized grid shell is more efficient.

In the optimization results provided by Karamba3D, the resulting cross sections are highly variable in many cases. All other factors being equal, large joints are required to accommodate the connection for diverse beams incident to a node. Furthermore, for diverse beams around a face, the face (which is already non-planar beforehand) would be additionally distorted, causing the loss of visual smoothness of the surface at the seams or would require variable-height panel supports. A key disadvantage of variating the cross section with respect to adding a reinforcement layer is that the resulting volume of the optimized cross sections obstructs the faces locally, limiting the transparency as in Fig. 17. This effect can be visualized even in the most efficient grid

Table 3
Volume of steel: reinforced grid shells with single $V_{\mathcal{R}opt'}$ and multi parameter $V_{\mathcal{R}opt}$ setup (after post processing), corresponding grid shell having constant cross section $V_{\mathcal{M}}$ (i.e. main layer) and optimized cross section $V_{\mathcal{M}opt}$ (using [72]).

Model	Field	$V_{\mathcal{M}}$ (m^3)	$V_{\mathcal{M}opt}$ (m^3)	$V_{\mathcal{R}opt'}$ (m^3)	$V_{\mathcal{R}opt}$ (m^3)
Arch	bend	45.40	75.74	63.33	65.64
	curv	45.40	93.01	63.33	65.36
	tsf	45.40	65.64	67.30	67.21
Dolphin	bend	35.43	90.76	58.60	57.37
	curv	36.01	63.69	45.74	46.92
	tsf	35.04	46.52	52.11	52.63
Envelope	bend	39.39	183.88	61.82	61.50
	curv	38.39	261.92	54.68	55.29
	tsf	37.71	95.73	62.64	60.28
Flower	bend	33.51	31.46	41.80	52.97
	curv	33.18	22.45	43.70	45.28
	tsf	34.44	39.34	51.20	54.35
Hall	bend	32.32	220.54	38.15	38.64
	curv	32.24	125.96	38.85	38.86
	tsf	32.09	37.45	46.17	46.30
Hangar	bend	61.44	173.92	115.36	109.63
	curv	60.75	84.51	113.43	104.83
	tsf	60.31	84.28	98.98	95.85
Manta	bend	44.27	75.29	69.61	69.75
	curv	43.59	64.31	68.55	68.30
	tsf	43.26	62.08	63.91	63.70
Roof	bend	18.92	129.08	25.60	26.71
	curv	19.25	104.21	28.37	28.90
	tsf	18.68	127.43	37.62	34.92
Rust	bend	45.41	35.13	70.32	71.94
	curv	42.68	29.24	57.93	58.28
	tsf	46.43	32.66	66.65	66.10
Shell1	bend	48.90	27.23	58.82	58.87
	curv	52.13	22.48	73.82	74.97
	tsf	50.62	16.94	70.26	70.01
Shell2	bend	25.61	26.48	32.74	34.21
	curv	25.13	22.66	38.19	40.13
	tsf	26.01	25.66	37.30	38.55
Taut	bend	34.97	25.97	48.54	49.37
	curv	36.16	26.95	59.84	59.10
	tsf	35.35	40.41	52.81	53.01
TriSlab	bend	35.05	64.32	75.40	61.94
	curv	34.29	63.22	70.84	69.09
	tsf	33.35	65.30	62.47	58.12
Vault	bend	20.19	13.21	29.69	28.67
	curv	19.39	24.65	25.54	25.43
	tsf	20.18	21.50	30.77	30.21
Wave	bend	24.42	31.17	34.59	34.47
	curv	25.61	16.43	33.33	33.37
	tsf	24.94	35.17	42.34	42.16
Yarn	bend	30.34	27.91	46.06	46.75
	curv	32.19	35.37	50.27	47.52
	tsf	32.22	29.62	57.09	58.87

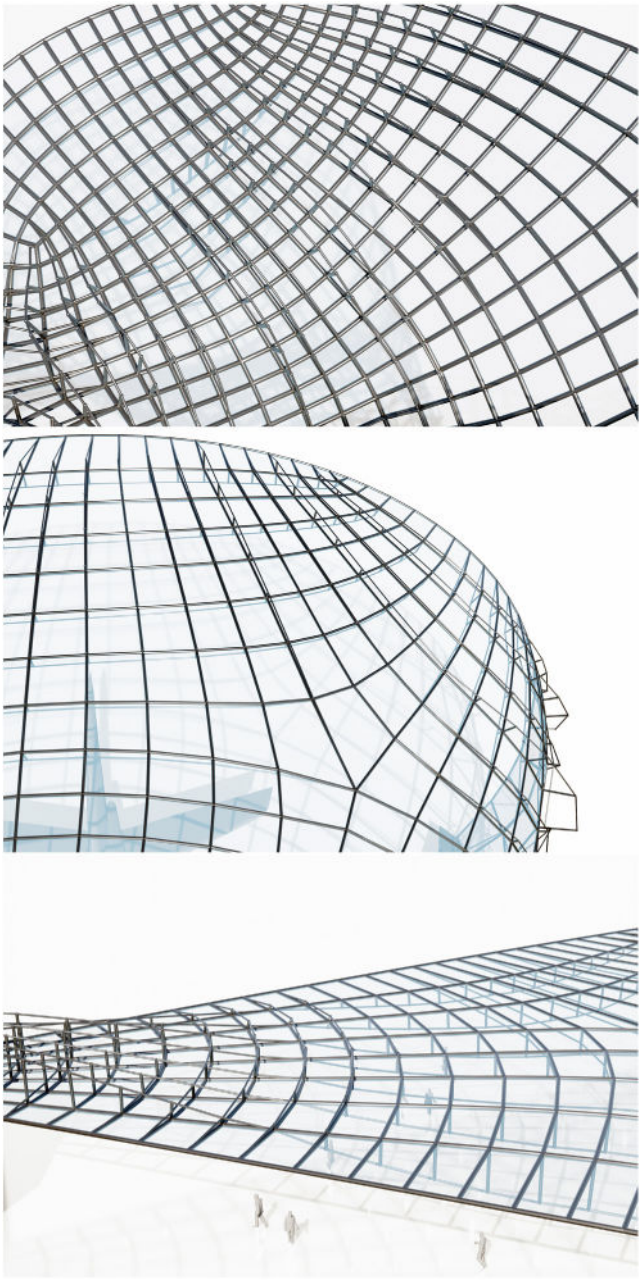

Fig. 16. Closeup view on reinforced grid shells: from top, Flower, Hangar, TriSlab.

shells (Fig. 17a–c). Conversely, if the set of commercial cross sections is bounded to a smaller maximum diameter, the optimized grid shell becomes less efficient and/or cannot reach the desired stiffness.

7.2. Comparison of different input fields

The selection of the main grid is the first factor influencing the reinforced solution since our formulation operates a discrete combinatorial problem. The shape of the main grid is invariant in our optimization and determines the organization and connectivity of the additional layer, as provided by input chords. Overall, our method can consistently produce suitable structures for any input field (Fig. 18): in every case, the performance of our structures is significantly better than their corresponding grid shells, as shown by comparing $\mathcal{E}_{\mathcal{M}}$ and $\mathcal{E}'_{\mathcal{R}opt}$ values. Moreover, quad meshes produced as in Section 5.2 led to similar reinforced structures regardless of the adopted field.

From a closer look, our field *tsf* provides better results as a minimal $\mathcal{E}'_{\mathcal{R}opt}$ is attained in most cases, while is not far from objective values obtained with other fields. Our *tsf* field produces a much better solution for bending-dominant surfaces akin to thick shells, such as some of the red-highlighted rows of Table 1.

Sometimes, the solution is governed by local mesh distortions, i.e., singularities and strong chord alterations, which often come with stress concentration and high strain energy. When working on nearly-developable shapes like Dolphin, Envelope, Hangar, Manta, the *curv* field provides a smooth quadrangulation and better reinforced grid shell results, favored by an orderly grid rather than a grid aligned to a structural-aware field. Moreover, this *curv* field has the additional

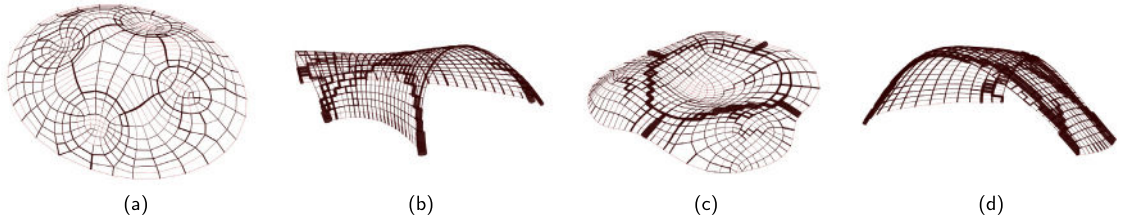

Fig. 17. Examples of grid shells with optimized cross sections used for the comparison in Table 3: (a) Rust *curv*; (b) Vault *bend*; (c) Wave *curv*; (d) Dolphin *bend*.

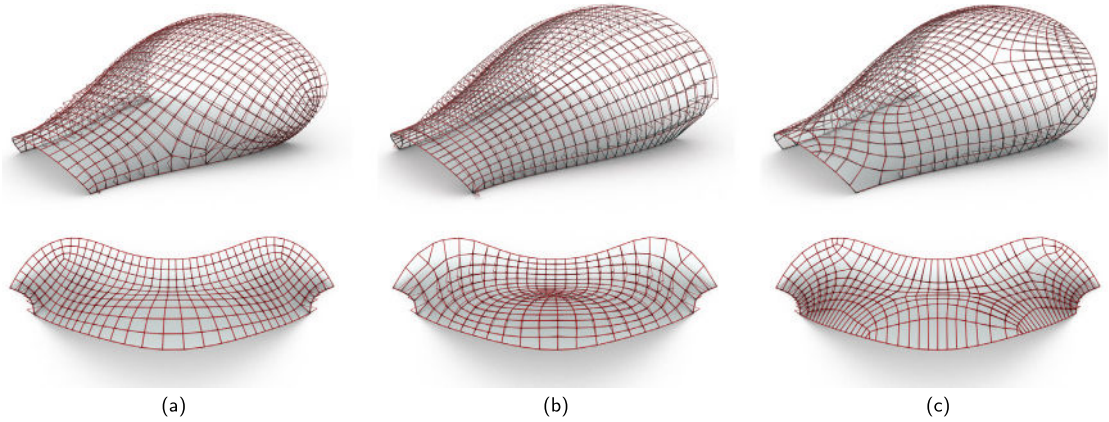

Fig. 18. Comparison of reinforced grid shells obtained on the models Hangar (top) and Shell2 (bottom) with a single-parameter setup from different input fields: (a) *bend*; (b) *curv*; (c) *tsf*.

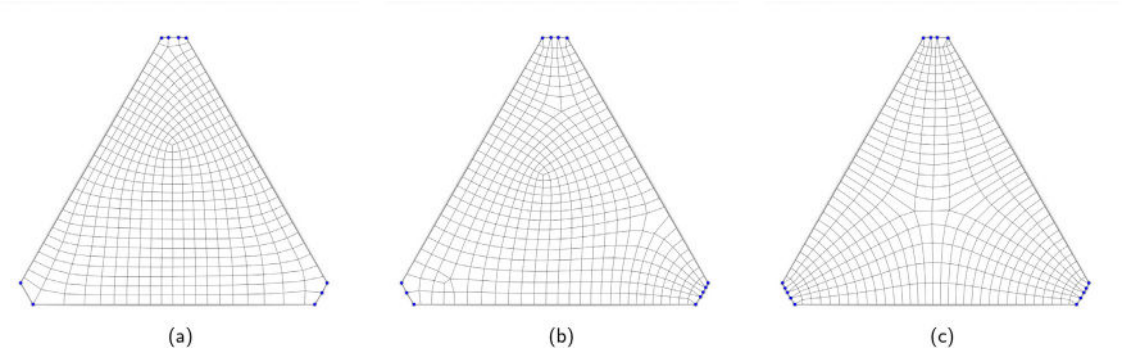

Fig. 19. Quad mesh obtained on the TriSlab model for different input fields: (a) *bend*; (b) *curv*; (c) *tsf*. Blue spheres at vertices indicate fixed supports of the structure. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

advantage of being derived from geometry only, so it saves time avoiding the initial shell analysis. Unfavorably, *curv* fails in the cases of (almost-) flat shapes (TriSlab case) or spherical curvature (Fig. 19).

For input surface behavior akin to a thin shell, like in Taut, the *bend* field provides better results. Even though this field is inaccurate in the presence of high bending, likewise *curv* it leads to well-shaped quads, which avoid localized strain energy peaks.

The starting mesh influences this problem, but the necessity of finding the optimal mesh is largely reduced by the strategy we adopt for the reinforcement initialization based on oblique bending. Thus, we can catch even bending forces not aligned with the mesh frames. As a drawback, non-optimal fields might have highly distorted strut directions, but the reinforced grid shell still outperforms the grid shell. We demonstrate the general validity of the method by running the example Duck in Fig. 20, based on a generic mesh. The performance of our structures is remarkable, but it generates struts that intersect the main layer, posing severe fabrication issues.

7.3. Sensitivity and accuracy

The optimization time is a function of the number of beams in the structure since the equilibrium and the number of optimization variables are computed at each step. Operating in a single parameter setup (results in Table 1) speeds up the optimization and delivers satisfactory results. Moreover, the shape of chords is uniformly proportional to their initialization, so the final solution generally preserves all chords. This feature leads to a consistent reinforcement design, which may be desirable for aesthetic reasons. On the other hand, in a multiple parameter setup, the objective function can reach a value up to 67.9% smaller (Table 2), but spending much longer time. In this setup, the x_k variable may be set to 0 independently, causing the collapse of the k chord on the main layer. Consequently, structural material and manufacturing costs are saved, but appearance is also affected, as shown in Fig. 21.

The post-processing impacts the objective function, as shown in Table 4, as it alters the number of beams, the chord path, and the strain energy consequently. However, in cases where the grid is not well

Table 4
Influence of the post processing on the results in the case of single and multiple parameter setup: model name (see Fig. 14, 15); fields adopted to generate the quad mesh (*bend*, *curv*, *tsf*); total strain energy on the optimized reinforced grid shell in single parameter $\mathcal{E}'_{Ropt}$ (from Table 1) and its corresponding value after post processing $\mathcal{E}'^{pp}_{Ropt}$; reduction (negative means worse performance); maximum norm of the displacements in both setups d; same quantities in a multi parameter setup.

Model	Field	$\mathcal{E}'_{Ropt}$ (kJ)	$\mathcal{E}'^{pp}_{Ropt}$ (kJ)	Cut (%)	$d_{Ropt'}$ (m)	$d^{pp}_{Ropt'}$ (m)	$\mathcal{E}_{Ropt}$ (kJ)	$\mathcal{E}^{pp}_{Ropt}$ (kJ)	Cut (%)	d_{Ropt} (m)	d^{pp}_{Ropt} (m)
Arch	bend	254.8	254.75	0.00	0.10	0.10	154.1	217.1	−40.87	0.09	0.09
	curv	254.2	254.22	0.00	0.10	0.10	163.3	167.9	−2.80	0.05	0.05
	tsf	211.2	211.18	0.00	0.07	0.07	141.6	222.0	−56.74	0.07	0.09
Dolphin	bend	120.3	126.15	−4.90	0.04	0.04	94.9	132.0	−39.12	0.03	0.05
	curv	273.7	328.88	−20.16	0.15	0.16	218.5	238.8	−9.30	0.10	0.10
	tsf	277.2	508.66	−83.49	0.17	0.30	176.1	495.9	−181.65	0.09	0.25
Envelope	bend	132.0	189.33	−43.42	0.13	0.10	97.8	222.9	−127.94	0.10	0.14
	curv	183.9	183.10	0.45	0.07	0.07	133.5	225.3	−68.70	0.04	0.09
	tsf	161.9	219.26	−35.39	0.06	0.08	113.3	278.3	−145.62	0.05	0.17
Flower	bend	361.2	360.83	0.10	0.42	0.42	115.8	177.0	−52.83	0.10	0.14
	curv	225.9	223.70	0.98	0.34	0.34	96.7	198.5	−105.37	0.14	0.21
	tsf	129.6	240.48	−85.50	0.11	0.27	74.1	133.5	−80.28	0.06	0.13
Hall	bend	40.9	40.92	0.03	0.03	0.03	19.8	19.8	0.08	0.01	0.01
	curv	34.0	34.01	−0.12	0.02	0.02	19.8	19.8	0.03	0.02	0.02
	tsf	23.1	31.42	−35.83	0.02	0.02	13.2	34.6	−163.17	0.01	0.04
Hangar	bend	1358.7	2129.76	−56.75	0.18	0.65	1017.1	1703.0	−67.43	0.20	0.60
	curv	1828.1	5294.18	−189.59	1.01	2.43	1196.5	2338.4	−95.44	0.31	0.98
	tsf	1812.1	3936.58	−117.24	0.34	1.15	1174.8	4600.1	−291.55	0.27	1.11
Manta	bend	2311.6	3842.53	−66.23	1.13	1.33	1418.4	4289.3	−202.39	0.64	1.47
	curv	6317.4	8900.57	−40.89	3.63	4.24	4764.0	8925.2	−87.35	2.50	4.28
	tsf	3677.4	7469.39	−103.12	1.78	2.18	2099.8	7770.1	−270.04	1.07	2.26
Roof	bend	282.6	282.21	0.12	0.47	0.47	177.6	177.0	0.36	0.28	0.28
	curv	316.9	316.67	0.06	0.50	0.50	254.1	253.5	0.21	0.39	0.39
	tsf	97.0	95.23	1.86	0.13	0.13	88.6	88.1	0.59	0.13	0.13
Rust	bend	276.9	332.47	−20.09	0.09	0.16	124.3	234.1	−88.39	0.04	0.08
	curv	550.8	790.60	−43.53	0.15	0.17	328.7	765.3	−132.81	0.14	0.18
	tsf	156.1	352.68	−125.95	0.06	0.11	82.3	237.9	−189.04	0.02	0.07
Shell1	bend	335.4	439.82	−31.15	0.14	0.16	253.9	539.4	−112.44	0.09	0.18
	curv	343.8	405.45	−17.92	0.08	0.12	232.2	514.7	−121.70	0.06	0.20
	tsf	198.1	343.53	−73.40	0.06	0.14	140.5	443.4	−215.72	0.05	0.22
Shell2	bend	2431.0	3391.10	−39.49	1.50	1.69	2030.7	2617.7	−28.91	1.25	1.50
	curv	1589.2	1574.68	0.91	0.94	1.18	1096.7	2503.4	−128.26	1.15	1.07
	tsf	829.2	1074.47	−29.57	0.43	0.59	554.0	1258.7	−127.20	0.25	0.86
Taut	bend	415.9	655.10	−57.50	0.49	0.62	295.6	535.7	−81.24	0.31	0.56
	curv	342.1	859.37	−151.22	0.26	0.53	297.6	859.4	−188.75	0.22	0.59
	tsf	466.5	817.26	−75.18	0.27	0.40	326.5	724.6	−121.93	0.14	0.39
TriSlab	bend	41 547.2	44 857.24	−7.97	6.37	6.59	20 517.6	51 995.8	−153.42	4.41	10.05
	curv	4036.1	6425.46	−59.20	1.27	1.58	2975.9	9603.3	−222.71	1.25	2.53
	tsf	3211.2	2960.59	7.80	1.41	1.04	2422.3	3572.4	−47.48	1.10	0.95
Vault	bend	6.5	7.23	−11.48	0.01	0.01	5.0	6.2	−24.17	0.01	0.01
	curv	7.2	7.22	0.00	0.01	0.01	5.5	5.5	0.00	0.01	0.01
	tsf	4.6	5.36	−16.35	0.01	0.01	4.1	5.7	−39.02	0.01	0.01
Wave	bend	22.0	27.59	−25.49	0.01	0.02	13.4	33.6	−150.45	0.01	0.02
	curv	35.8	93.61	−161.53	0.02	0.07	23.1	80.1	−246.23	0.01	0.03
	tsf	17.1	55.57	−224.61	0.01	0.04	11.2	32.5	−191.23	0.00	0.01
Yarn	bend	1940.2	4673.74	−140.89	1.45	2.23	1127.2	4819.0	−327.51	0.38	2.70
	curv	804.6	2101.32	−161.16	0.69	2.16	314.8	2956.2	−839.10	0.15	2.51
	tsf	352.1	449.11	−27.55	0.35	0.21	273.9	1125.3	−310.80	0.17	1.62

aligned to the flow of forces, the directions of the struts on different chords might result too far, or the chord paths kinky. Hence, the regularity of the main layer mesh has a relevant role, so the post-process effect is hardly case-dependent. Other limitations for functionality and fabrication are embedded in the initialization and optimization of the additional layer. For instance, in Table 1, the *curv* Roof case has a negative cut because the solution $\mathcal{E}_{\mathcal{R}_0}$ is computed at the initialization of the additional layer, before the limitation on $L_{s,max}$ is applied.

To show the sensitivity of our method in creating appropriate solutions, we solve the case of a HemiSphere example, having a polar symmetric mesh as in Fig. 22. The objective is to prove our method against a known scenario, in which it could be intuitive to guess an optimal solution since the radial bending moments have a known shape in the continuum. We use hollow sections of 0.15 m of external diameter and 0.015 m thickness for the main beams and solid circular sections of 0.08 m diameter for the reinforcement. The results show that for case (a), in which we adopt the same load as in other experiments, no additional layer is included since the bending moments can be supported by the main grid alone. It would not have been efficient to add other elements. This result also occurs when inputting a membrane mesh. If the load doubles in case (b), additional symmetric layers are created radially on the lower quad hoop. However, the expected shape of the radial bending appears on the condition that the post-processing filters are removed in case (c). Bending, from negative to positive, originates from the compatibility of the internal forces on the surface with the fixed supports. However, this edge effect is damped out rapidly. In theory, additional layer elements are required to cross the main layer, but the shortest strut is collapsed due to filtering.

To confirm the results of our method, we process them with a commercial FE software [73], with which we remove the linearity

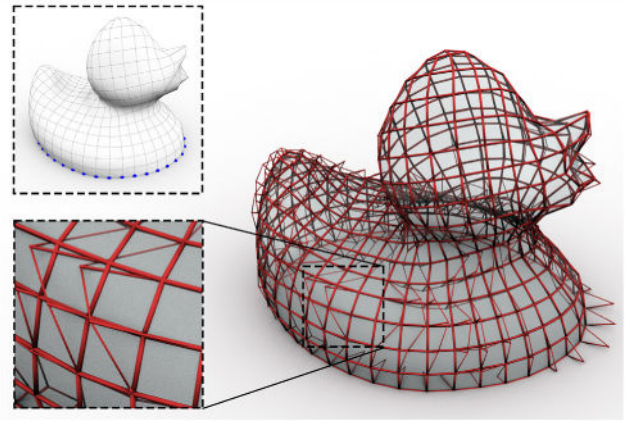

Fig. 20. Reinforced grid shells obtained from a generic input mesh Duck: initial mesh with blue spheres at vertices as supports, conceptual view, closeup showing conflicts. Size: 35.31 m × 41.54 m × 32.9 m, i = 488, j = 954, k = 37, $\mathcal{E}_{\mathcal{M}}$ = 2514.6 kJ, $\mathcal{E}'_{\mathcal{R}}$ = 68.16 kJ. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

assumption and use a more complex beam model. We adopt the same boundary and loading conditions, material, and structural elements. We run a geometrically nonlinear analysis in which the load is applied gradually, and the structure's stiffness matrix is updated at each load step. In Fig. 23, we report a comparison of two models having extreme behavior: Roof, which is poorly supported and highly prone to large displacement due to bending, and Rust, which is continuously supported on the boundary and has a shell-like behavior. We compare our deformed shape obtained by solving Eq. (19) and the deformed shape obtained from the nonlinear analysis. Fig. 23 shows the histogram of the distances between the nodes of the two deformed shapes. For the Roof case, the maximum distance is 0.561 m at the tip of the structure ($d_{\mathcal{R}opt}$ = 0.063 m vs d_{nonlin} = 0.624 m), and the error accumulates moving from the supports to the tip, as expected. For the Rust case, the maximum distance is 0.043 m ($d_{\mathcal{R}opt}$ = 0.009 m vs d_{nonlin} = 0.052 m), and the error is very low. The deformed shapes are qualitatively similar, but the reliability of the analysis results increases with the stiffness of the initial grid shell.

8. Conclusions

In this paper, we introduce a novel method to design automatically a new type of structure, reinforced grid shells, built on general free-form architectural surfaces. We decouple this problem into two phases. First, we derive a quad-dominant mesh aligned to a field obtained from the FE analysis of the continuous shell. Then, we optimize the structure to capture out-of-plane load. This approach has the great advantage of separating the discretization from the optimization in a discrete setup, which makes it exceptionally flexible as these structures can be built from different input fields, using our remeshing with anisotropy, or directly from any input mesh. The optimization is fast and achieves solutions that are more efficient than the corresponding grid shell in the case of non-membrane shapes.

The most exciting feature is that this method provides no shape alteration of the initial surface, which is desirable from the architectural perspective. The main layer has a prevailing stiffness and can safely withstand the in-plane forces coming from the cladding placed on it. The additional layer is visually lighter and can be located inward and outward, crossing the main grid only at nodes.

The main limitation is that we do not expressly optimize our mesh for the membrane (in-plane) behavior. Nevertheless, we can manage grids unaligned with in-plane forces since we consider oblique bending

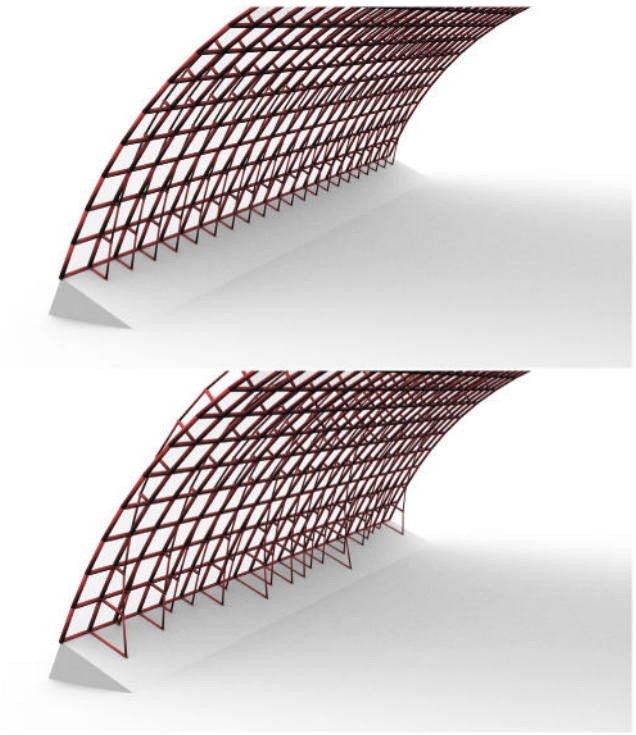

Fig. 21. Visual comparison of reinforced grid shells obtained on the Arch model in a single (top) and multi parameter setup (bottom): closeup on one support.

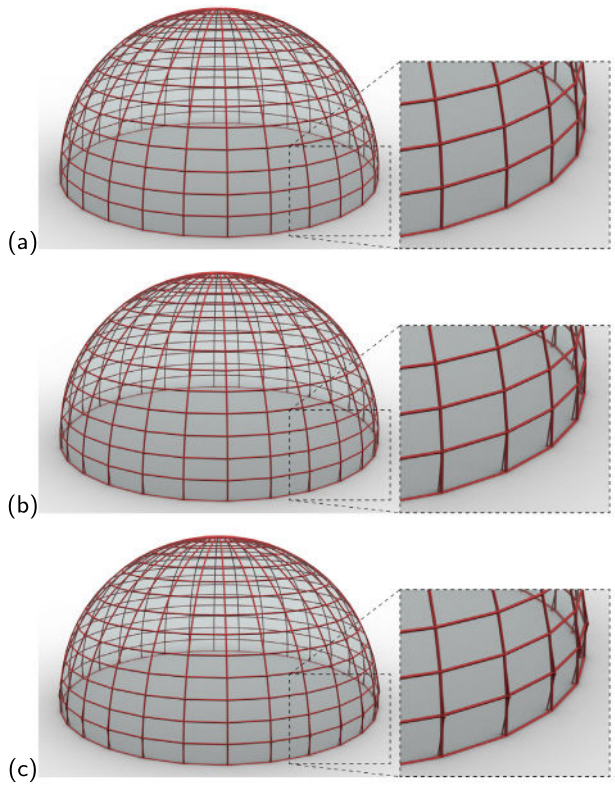

Fig. 22. Conceptual view of reinforced grid shells obtained from a HemiSphere model with closeups at the supports: results for (a) 2 kN/m^2 load, (b) a 2× load, (c) a 2× load without post-processing. Size: 40 × 40 × 20 m.

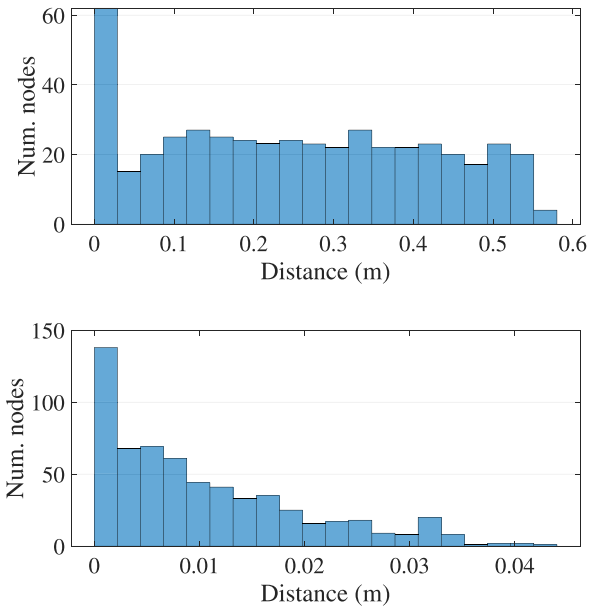

Fig. 23. Nonlinear analysis of our structures: distances between the deformed shapes computed with our method and with a commercial software; Roof (on top), Rust (on bottom).

in the formulation of the additional layer. Our solutions are generally as efficient as the input surfaces are bending-dominated. On the other hand, it must be considered that these structures use more elements than a standard grid shell, impacting fabrication and assembly, and have different aesthetics. However, they can be a practical choice in free-form shapes for which grid shells lack stiffness.

Dealing with free-form shapes, our method naturally produces several unique curved panel geometries and non-congruent nodes, which in some cases may represent a significant limitation to buildability, for instance, due to the costs of heat-bent glass panels of different curvature. Future work shall focus on embedding fabrication constraints and advancing the practical application of this kind of structure, making fabrication affordable, i.e. by adopting planar panels or modular nodes.

Another limitation is the optimization for a single load case. In the future, multiple loading scenarios shall be included to reinforce any input grid shells for asymmetric load conditions. The most interesting future direction is turning the reinforcement into a post-tensioned cable, which can impose compression on the main grid. Thus, compression-only materials could also be safely employed for the main layer.

CRediT authorship contribution statement

Francesco Laccone: Conceptualization, Data curation, Formal analysis, Funding acquisition, Methodology, Project administration, Validation, Visualization, Writing – original draft, Writing – review & editing, Investigation, Software. **Nico Pietroni:** Conceptualization, Methodology, Software, Supervision, Writing – original draft, Data curation, Visualization, Writing – review & editing. **Paolo Cignoni:** Conceptualization, Funding acquisition, Methodology, Supervision, Project administration, Resources. **Luigi Malomo:** Conceptualization, Data curation, Methodology, Project administration, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

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.

Data availability

The geometric models and the quad meshes used in this paper are available at: https://doi.org/10.5281/zenodo.10391043.

Acknowledgments

The authors wish to thank the anonymous Reviewers and the Editor for their helpful comments, and Prof. Maurizio Froli for the early discussions around the topic of the paper.

Funding

This research was partially developed during the Short Term Mobility (STM) of Francesco Laccone, hosted by Nico Pietroni at the UTS. The STM grant was founded by the National Research Council of Italy (CNR).

References

[1] Adriaenssens S, Block P, Veenendaal D, Williams C. Shell structures for architecture: form finding and optimization. Routledge; 2014.
[2] Venuti F, Bruno L. Influence of in-plane and out-of-plane stiffness on the stability of free-edge gridshells: A parametric analysis. 131, 2018, p. 755–68. http://dx.doi.org/10.1016/j.tws.2018.07.019,
[3] Bletzinger K-U, Ramm E. Structural optimization and form finding of light weight structures. 79, (22–25):Elsevier; 2001, p. 2053–62. http://dx.doi.org/10.1016/S0045-7949(01)00052-9,
[4] Ramm E, Wall W. Shell structures—a sensitive interrelation between physics and numerics. 60, (1):Wiley Online Library; 2004, p. 381–427. http://dx.doi.org/10.1002/nme.967,
[5] Malek S, Williams C. The equilibrium of corrugated plates and shells. 19, Springer; 2017, p. 619–27. http://dx.doi.org/10.1007/s00004-017-0347-7,
[6] Malek SR. The effect of geometry and topology on the mechanics of grid shells [Ph.D. thesis], Massachusetts Institute of Technology; 2012.
[7] Vouga E, Höbinger M, Wallner J, Pottmann H. Design of self-supporting surfaces. 31, (4):New York, NY, USA: ACM; 2012, p. 87:1–87:11. http://dx.doi.org/10.1145/2185520.2185583,
[8] Van Mele T, Block P. A novel form finding method for fabric formwork for concrete shells. 52, (4):International Association for Shell and Spatial Structures (IASS); 2011, p. 217–24,
[9] Miki M, Adiels E, Baker W, Mitchell T, Sehlström A, Williams CJ. Form-finding of shells containing both tension and compression using the airy stress function. 37, (4):SAGE Publications Sage UK: London, England; 2022, p. 261–82. http://dx.doi.org/10.1177/09560599221102618,
[10] Gugeler J, Havemann K, Schober H. Lehrter Bahnhof Berlin: Das Nord-Süd-Dach. Stahlbau 2006;75(3):194–202.
[11] Hunt AJ, Jones AC, Otlet M, Dexter DI. Waterloo International rail terminal trainshed roof structure. Struct Eng 1994;72:123.
[12] Pottmann H, Eigensatz M, Vaxman A, Wallner J. Architectural geometry. 47, 2015, p. 145–64. http://dx.doi.org/10.1016/j.cag.2014.11.002,
[13] Veenendaal D, Block P. An overview and comparison of structural form finding methods for general networks. 49, (26):2012, p. 3741–53. http://dx.doi.org/10.1016/j.ijsolstr.2012.08.008,
[14] Schek H-J. The force density method for form finding and computation of general networks. 3, (1):Elsevier; 1974, p. 115–34,
[15] Block P. Thrust network analysis: exploring three-dimensional equilibrium [Ph.D. thesis], Massachusetts Institute of Technology; 2009, URL http://hdl.handle.net/1721.1/49539.
[16] Millar C, Mitchell T, Mazurek A, Chhabra A, Beghini A, Clelland JN, et al. On designing plane-faced funicular gridshells. SAGE Publications Sage UK: London, England; 2021, 09560599221126656. http://dx.doi.org/10.1177/09560599221126656,
[17] Konstantatou M, Baker W, Nugent T, McRobie A. Grid-shell design and analysis via reciprocal discrete airy stress functions. 37, (2):SAGE Publications Sage UK: London, England; 2022, p. 150–64. http://dx.doi.org/10.1177/09560599221081004,
[18] Mitchell T, Baker W, McRobie A, Mazurek A. Mechanisms and states of self-stress of planar trusses using graphic statics, part I: The fundamental theorem of linear algebra and the airy stress function. 31, (2–4):SAGE Publications Sage UK: London, England; 2016, p. 85–101. http://dx.doi.org/10.1177/0266351116660790,
[19] Ohlbrock PO, D'Acunto P. A computer-aided approach to equilibrium design based on graphic statics and combinatorial variations. 121, 2020, 102802. http://dx.doi.org/10.1016/j.cad.2019.102802,

[20] Winslow P, Pellegrino S, Sharma S. Multi-objective optimization of free-form grid structures. 40, Springer; 2010, p. 257–69. http://dx.doi.org/10.1007/s00158-009-0358-4,

[21] Pietroni N, Tonelli D, Puppo E, Froli M, Scopigno R, Cignoni P. Statics aware grid shells. 34, (2):2015, p. 627–41. http://dx.doi.org/10.1111/cgf.12590,

[22] Kilian M, Pellis D, Wallner J, Pottmann H. Material-minimizing forms and structures. 36, (6). New York, NY, USA: Association for Computing Machinery; 2017, http://dx.doi.org/10.1145/3130800.3130827,

[23] Tibert A, Pellegrino S. Review of form-finding methods for tensegrity structures. 18, (4):2003, p. 209–23. http://dx.doi.org/10.1260/026635103322987940,

[24] Pietroni N, Tarini M, Vaxman A, Panozzo D, Cignoni P. Position-based tensegrity design. 36, (6):New York, NY, USA: ACM; 2017, p. 172:1–172:14. http://dx.doi.org/10.1145/3130800.3130809,

[25] Palmieri M, Giannetti I, Micheletti A. Floating-bending tensile-integrity structures. 8, (1):De Gruyter; 2021, p. 89–95. http://dx.doi.org/10.1515/cls-2021-0008,

[26] Jiang C, Tang C, Seidel H-P, Wonka P. Design and volume optimization of space structures. 36, (4):New York, NY, USA: ACM; 2017, p. 159:1–159:14. http://dx.doi.org/10.1145/3072959.3073619,

[27] Jiang C, Tang C, peter Seidel H, Chen R, Wonka P. Computational design of lightweight trusses. 2021, 103076. http://dx.doi.org/10.1016/j.cad.2021.103076,

[28] Lienhard J, Alpermann H, Gengnagel C, Knippers J. Active bending, a review on structures where bending is used as a self-formation process. 28, (3–4):SAGE Publications Sage UK: London, England; 2013, p. 187–96. http://dx.doi.org/10.1260/0266-3511.28.3-4.187,

[29] Panetta J, Konaković-Luković M, Isvoranu F, Bouleau E, Pauly M. X-Shells: A new class of deployable beam structures. 38, (4). New York, NY, USA: Association for Computing Machinery; 2019, http://dx.doi.org/10.1145/3306346.3323040,

[30] Laccone F, Malomo L, Pietroni N, Cignoni P, Schork T. Integrated computational framework for the design and fabrication of bending-active structures made from flat sheet material. 34, 2021, p. 979–94. http://dx.doi.org/10.1016/j.istruc.2021.08.004,

[31] Fu C-W, Lai C-F, He Y, Cohen-Or D. K-set tilable surfaces. ACM Trans Graph 2010;29(4):44:1–6.

[32] Singh M, Schaefer S. Triangle surfaces with discrete equivalence classes. ACM Trans Graph 2010;29(4):46:1–7.

[33] Austern G, Capeluto IG, Grobman YJ. Rationalization methods in computer aided fabrication: A critical review. 90, 2018, p. 281–93. http://dx.doi.org/10.1016/j.autcon.2017.12.027,

[34] Wang H, Pellis D, Rist F, Pottmann H, Müller C. Discrete geodesic parallel coordinates. 38, (6). New York, NY, USA: Association for Computing Machinery; 2019, http://dx.doi.org/10.1145/3355089.3356541,

[35] Liu Z-Y, Zhang Z, Zhang D, Ye C, Liu L, Fu X-M. Modeling and fabrication with specified discrete equivalence classes. 40, (4). New York, NY, USA: Association for Computing Machinery; 2021, http://dx.doi.org/10.1145/3450626.3459843,

[36] Liu Y, Pottmann H, Wallner J, Yang Y-L, Wang W. Geometric modeling with conical meshes and developable surfaces. ACM Trans Graph 2006;25(3):681–9.

[37] Verhoeven F, Vaxman A, Hoffmann T, Sorkine-Hornung O. Dev2PQ: Planar quadrilateral strip remeshing of developable surfaces. 41, (3). New York, NY, USA: Association for Computing Machinery; 2022, http://dx.doi.org/10.1145/3510002,

[38] Liu Y, Xu W, Wang J, Zhu L, Guo B, Chen F, et al. General planar quadrilateral mesh design using conjugate direction field. ACM Trans Graph 2011;30(6):140:1–140:10.

[39] Jiang C, Wang C, Tellier X, Wallner J, Pottmann H. Planar panels and planar supporting beams in architectural structures. Association for Computing Machinery (ACM); 2022, http://dx.doi.org/10.1145/3561050, URL http://hdl.handle.net/10754/680820,

[40] Wang C, Jiang C, Wang H, Tellier X, Pottmann H. Architectural structures from quad meshes with planar parameter lines. 156, 2023, 103463. http://dx.doi.org/10.1016/j.cad.2022.103463,

[41] Huard M, Eigensatz M, Bompas P. Planar panelization with extreme repetition. 2015, p. 259–79. http://dx.doi.org/10.1007/978-3-319-11418-7_17.

[42] Pottmann H, Liu Y, Wallner J, Bobenko A, Wang W. Geometry of multi-layer freeform structures for architecture. ACM Trans Graph 2007;26(3). http://dx.doi.org/10.1145/1276377.1276458.

[43] sheng Wang Q, Ye J, Wu H, qing Gao B, Shepherd P. A triangular grid generation and optimization framework for the design of free-form gridshells. 113, 2019, p. 96–113. http://dx.doi.org/10.1016/j.cad.2019.04.005,

[44] Grande E, Imbimbo M, Tomei V. Role of global buckling in the optimization process of grid shells: Design strategies. 156, 2018, p. 260–70. http://dx.doi.org/10.1016/j.engstruct.2017.11.049,

[45] Oval R, Rippmann M, Mesnil R, Van Mele T, Baverel O, Block P. Feature-based topology finding of patterns for shell structures. 103, 2019, p. 185–201. http://dx.doi.org/10.1016/j.autcon.2019.02.008,

[46] Laccone F, Malomo L, Froli M, Cignoni P, Pietroni N. Automatic design of cable-tensioned glass shells. 39, (1):2020, p. 260–73. http://dx.doi.org/10.1111/cgf.13801,

[47] Li W, Zheng A, You L, Yang X, Zhang J, Liu L. Rib-reinforced shell structure. 36, (7):2017, p. 15–27. http://dx.doi.org/10.1111/cgf.13268,

[48] Tam K-MM, Mueller CT. Additive Manufacturing Along Principal Stress Lines. 4, (2):2017, p. 63–81. http://dx.doi.org/10.1089/3dp.2017.0001,

[49] Halpern AB, Billington DP, Adriaenssens S. The ribbed floor slab systems of pier luigi nervi. In: Proceedings of IASS annual symposia, vol. 2013, no. 23. International Association for Shell and Spatial Structures (IASS); 2013, p. 1–7.

[50] Burger J, Huber T, Lloret-Fritschi E, Mata-Falcón J, Gramazio F, Kohler M. Design and fabrication of optimised ribbed concrete floor slabs using large scale 3D printed formwork. 144, 2022, 104599. http://dx.doi.org/10.1016/j.autcon.2022.104599, URL https://www.sciencedirect.com/science/article/pii/S0926580522004691,

[51] Ma J, Gomaa M, Bao DW, Javan AR, Xie YM. Printnervi–design and construction of a ribbed floor system in the digital era. 63, (4):International Association for Shell and Spatial Structures (IASS); 2022, p. 241–51. http://dx.doi.org/10.20898/j.iass.2022.017,

[52] Gil-Ureta F, Pietroni N, Zorin D. Reinforcement of general shell structures. 39, (5). New York, NY, USA: Association for Computing Machinery; 2020, http://dx.doi.org/10.1145/3375677,

[53] Robeller C, Konaković M, Dedijer M, Pauly M, Weinand Y. Double-layered timber plate shell. 32, (3–4):2017, p. 160–75. http://dx.doi.org/10.1177/0266351117742853,

[54] Rezaei Rad A, Burton H, Rogeau N, Vestartas P, Weinand Y. A framework to automate the design of digitally-fabricated timber plate structures. 244, 2021, 106456. http://dx.doi.org/10.1016/j.compstruc.2020.106456,

[55] Wagner HJ, Alvarez M, Groenewolt A, Menges A. Towards digital automation flexibility in large-scale timber construction: integrative robotic prefabrication and co-design of the BUGA Wood Pavilion. 4, (3):Springer; 2020, p. 187–204. http://dx.doi.org/10.1007/s41693-020-00038-5,

[56] Laccone F, Manolas I, Malomo L, Cignoni P. Exploratory study on a segmented shell made of recycled-hdpe plastic. In: Proceedings of IASS2020/21-SURREY. 2021.

[57] Chilton J. Space grid structures. Routledge; 2007, http://dx.doi.org/10.4324/9780080498188.

[58] Todisco L, Peiretti HC, Mueller C. Funicularity through external posttensioning: design philosophy and computational tool. J Struct Eng 2016;142(2):04015141. http://dx.doi.org/10.1061/(ASCE)ST.1943-541X.0001416.

[59] Todisco L, Corres-Peiretti H, Mueller C, et al. Exploration of externally post-tensioned spatial structures using 3D graphic statics. In: Proceedings of IASS annual symposia, vol. 2018, no. 16. International Association for Shell and Spatial Structures (IASS); 2018, p. 1–8.

[60] Van Mele T, Lachauer L, Rippmann M, Block P. Geometry-based understanding of structures. 53, (4):International Association for Shell and Spatial Structures (IASS); 2012, p. 285–95,

[61] Vaxman A, Campen M, Diamanti O, Panozzo D, Bommes D, Hildebrandt K, et al. Directional field synthesis, design, and processing. 35, 2016, p. 545–72. http://dx.doi.org/10.1111/cgf.12864,

[62] Diamanti O, Vaxman A, Panozzo D, Sorkine-Hornung O. Designing N-PolyVector fields with complex polynomials. 33, (5):Chichester, GBR: The Eurographs Association & John Wiley & Sons, Ltd.; 2014, p. 1–11,

[63] Vaxman A, et al. Directional: A library for Directional Field Synthesis, Design, and Processing, http://dx.doi.org/10.5281/zenodo.3338174.

[64] Bommes D, Zimmer H, Kobbelt L. Mixed-integer quadrangulation. ACM Trans Graph 2009;28(3):1.

[65] Stephan S, Sánchez-Alvarez J, Knebel K. Reticulated structures on free-form surfaces. 73, (8):2004, p. 562–72. http://dx.doi.org/10.1002/stab.200490149,

[66] Schober H. Transparent shells: form, topology, structure. John Wiley & Sons; 2015.

[67] Liu Y, Pottmann H, Wallner J, Yang Y-L, Wang W. Geometric modeling with conical meshes and developable surfaces. In: ACM SIGGRAPH 2006 papers. New York, NY, USA: Association for Computing Machinery; 2006, p. 681–9. http://dx.doi.org/10.1145/1179352.1141941.

[68] ANSYS. Academic research mechanical release 18.0. 2019.

[69] Pietroni N, Nuvoli S, Alderighi T, Cignoni P, Tarini M. Reliable feature-line driven quad-remeshing. 40, (4). New York, NY, USA: Association for Computing Machinery; 2021, http://dx.doi.org/10.1145/3450626.3459941,

[70] Johnson SG. The nlopt nonlinear-optimization package. 2021, URL https://github.com/stevengj/nlopt.

[71] Rowan TH. Functional stability analysis of numerical algorithms [Ph.D. thesis], USA: University of Texas at Austin; 1990, UMI Order No. GAX90-31702.

[72] Preisinger C, Heimrath M. Karamba - a toolkit for parametric structural design. 24, (2):Taylor & Francis; 2014, p. 217–21. http://dx.doi.org/10.2749/101686614X13830790993483,

[73] G+D Computing. Straus7. Sydney, Australia: G+ D Computing; 2005.