Unlocking Euclidean Problems with Isotropic Initialization

Khusrav Yorov[a,1,*], Bolun Wang[b,c,1], Mikhail Skopenkov[a], Helmut Pottmann[d], Caigui Jiang[e]

[a] *King Abdullah University of Science and Technology (KAUST), Thuwal 23955, Saudi Arabia*
[b] *MAIS, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*
[c] *RWTH Aachen University, 52062 Aachen, Germany*
[d] *TU Wien, 1040 Vienna, Austria*
[e] *Xi'an Jiaotong University, Xi'an, China*

Abstract

Many problems in Euclidean geometry, arising in computational design and fabrication, amount to a system of constraints, which is challenging to solve. We suggest a new general approach to the solution, which is to start with analogous problems in isotropic geometry. Isotropic geometry can be viewed as a structure-preserving simplification of Euclidean geometry. The solutions found in the isotropic case give insight and can initialize optimization algorithms to solve the original Euclidean problems. We illustrate this general approach with three examples: quad-mesh mechanisms, composite asymptotic-geodesic gridshells, and asymptotic gridshells with constant node angle.

Keywords: discrete differential geometry, isotropic geometry, computational design, initialization of optimization, quad mesh mechanism, asymptotic-geodesic gridshell

1. Introduction

Many problems in computational design and fabrication amount to the solution of a system of constraints which is solved by numerical optimization. Despite the presence of constraints, the solution space or design space may still be sufficiently large to be explored by a designer. However, it may be difficult to access this design space. This is equivalent to finding an initial guess for the underlying optimization problem. Especially difficult cases are those where one has very few, or even no explicit solution. An example for the latter situation is provided by hybrid asymptotic-geodesic gridshells. See Figs. 1–2.

There are various ways to handle such difficult situations. One may initially omit some constraints, possibly find a solution then, and use that to initialize optimization. One may also initialize with a nearly trivial special case, with the drawback that this may not open a path towards interesting solutions.

*Corresponding author.
Email addresses: khusrav.yorov@kaust.edu.sa (Khusrav Yorov), bolun.wang@rwth-aachen.de (Bolun Wang), mikhail.skopenkov@gmail.com (Mikhail Skopenkov), helmut.pottmann@gmail.com (Helmut Pottmann), cgjiang@xjtu.edu.cn (Caigui Jiang)
[1] Joint first authors.

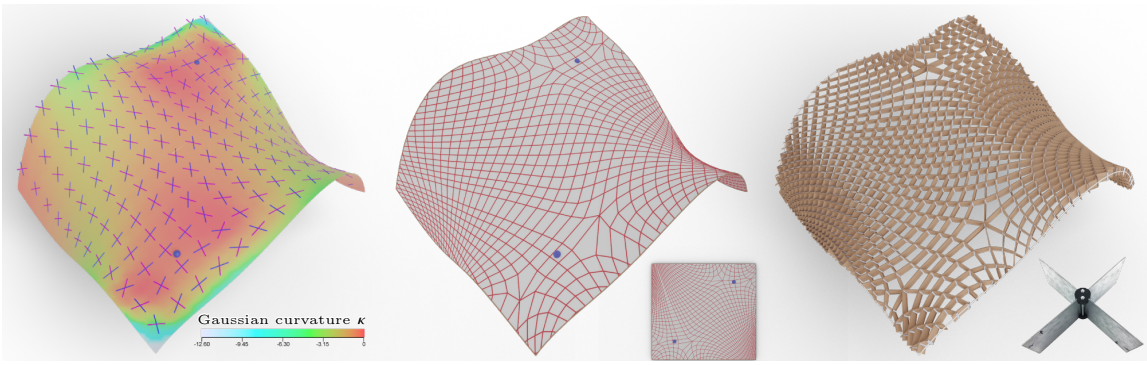


Figure 1: Design of a so-called asymptotic gridshell with prescribed constant node angle and positions of combinatorial singularities by isotropic initialization. An *asymptotic gridshell* (right) is obtained by bending originally straight lamellas and arranging them in a quadrilateral grid so that they are orthogonal to a reference surface. This forces the lamellas to follow the asymptotic directions of the surface. A constant intersection angle of lamellas simplifies fabrication, as it allows the use of identical nodes throughout the structure (inset). Left: A surface with a nearly constant *isotropic* angle between asymptotic directions (crosses) and flat points near prescribed positions (blue dots). Color shows the variation of Euclidean Gaussian curvature. Such a surface is given by a simple analytic expression in isotropic geometry, but not in Euclidean geometry. Middle: The remeshed surface and its top view (in the inset). Singularities appear around the blue points where the flat points are prescribed. Right: An asymptotic gridshell with a constant *Euclidean* angle 72° between the lamellas obtained by optimization initialized by the mesh in the middle.

In the present paper, we suggest the following general two-step approach to certain hard problems in Euclidean geometry, allowing us to overcome this obstacle.

In the first step, we solve the same problem in isotropic geometry. Isotropic geometry is one of the classical non-Euclidean geometries, and can be viewed as a structure-preserving simplification of Euclidean geometry. It is obtained by replacing the Euclidean norm $\sqrt{x^2+y^2+z^2}$ in space with the simpler isotropic semi-norm $\sqrt{x^2+y^2}$. That alone would be too degenerate. However, isotropic geometry is also based on a 6-dimensional group of isotropic congruence transformations, which leads to a wealth of results that are very similar to their Euclidean counterparts [1].

The second step is to use the isotropic solution to initialize an optimization algorithm for a numerical solution to the original Euclidean problem. Last but not least, the isotropic solution may provide insight into how the Euclidean problem can be solved.

This approach can be traced back to classical works. An example is the Plateau problem on finding a minimal surface with a given boundary. This problem was solved in a fairly general setting by Müntz [2]. He constructed minimal surfaces by deformation of the graphs of harmonic functions. The latter are minimal surfaces in isotropic geometry. So, in this case, the optimization led to the whole existence proof and a construction method. The suggested approach works surprisingly well for various practical geometric problems.

We illustrate it with examples, which are nearly inaccessible by the methods available before: construction

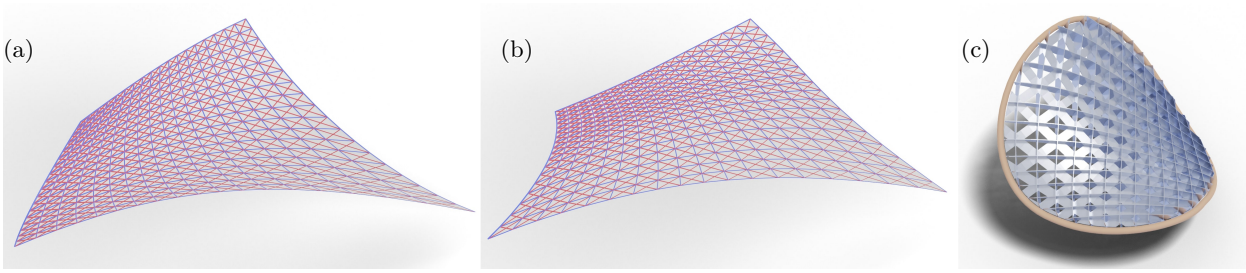


Figure 2: Design of a so-called *composite asymptotic-geodesic gridshell* by isotropic initialization. In particular, *an AGAG gridshell* (c) is obtained by bending originally flat lamellas and arranging them so that some are orthogonal and some are tangent to a reference surface, and four lamellas meet at each node. This forces the lamellas to follow the asymptotic (red) and geodesic (blue) curves on the surface respectively, organized in a so-called AGAG web. We construct an isotropic AGAG web (a) and then optimize it to a Euclidean one (b). The meshes were downsampled by removing some polylines from the red and blue families to reduce density while keeping their overall shape. Cutting along a closed curve leads to a gridshell (c). Isotropic AGAG webs are completely classified, but it is unknown if a single nontrivial smooth Euclidean AGAG web exists.

of quad-mesh mechanisms (see Fig. 3), composite asymptotic-geodesic gridshells (see Fig. 2), and asymptotic gridshells with constant node angle (see Fig. 1 and Fig. 4).

The presented method is not a general-purpose simple trick to solve all kinds of hard problems in Euclidean geometry. It naturally has certain limitations. Most importantly, one must be able to come up with a meaningful isotropic initial guess that is not too degenerated to be transformed into a Euclidean solution. Our examples are interesting in various ways. This may not be straightforward, as is illustrated in some of our examples, especially on AGAG webs and quad mesh mechanisms. The latter required a new non-degenerate definition of isometric surfaces in isotropic 3-space [3].

1.1. Related work

Fundamental contributions to isotropic geometry [4, 5, 6] and the monograph [1] are written in German. Thus, we point to further contributions that contain a summary of key concepts, namely [7, 8, 9, 10].

In the following, we first concentrate on known applications of isotropic geometry and then turn to prior work on the applications treated in this paper. For those, the use of isotropic geometry is new.

Structural design. K. Strubecker [11] pointed out that the so-called Airy stress function, which is associated with a planar equilibrium state, is best understood when viewing its graph within isotropic geometry. There is a complete correspondence between mechanical properties, e.g. stresses, and properties of the Airy stress surface in isotropic geometry, e.g. isotropic curvatures. Following up on that, various contributions to structural design, in particular when considering only vertical loads, exploited the geometry of the stress surface for design [12, 13, 14, 15, 16, 17]. With constant vertical loads, stress surface and design surface may agree up to vertical scaling [15] and then represent surfaces of constant isotropic Gaussian curvature, for which elegant geometric constructions are known [5, 18, 19]. These self-Airy surfaces provide many design

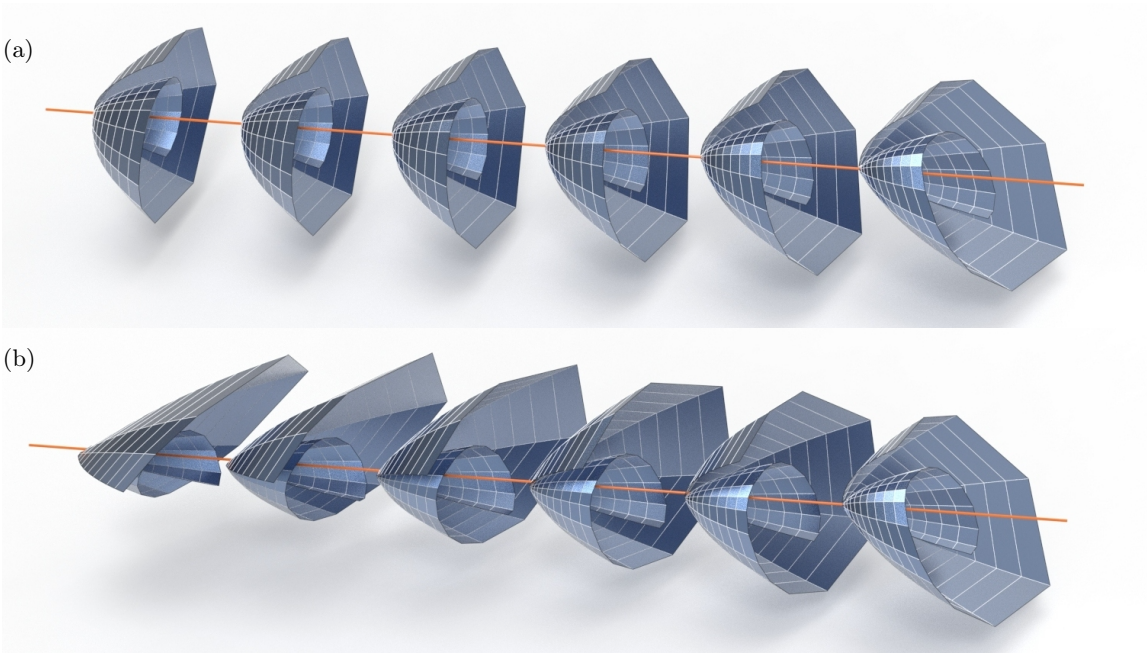


Figure 3: Design of mechanisms with rigid faces and rotational joints in edges by isotropic initialization. We construct an isotropic mechanism (a) and then optimize it to a Euclidean one (b). A few positions of each mechanism are shown. The line depicts the isotropic direction. While the initial (rightmost) positions of two mechanisms are visually the same, the flexion leads to different shapes. Isotropic mechanisms are completely classified, whereas the construction of Euclidean ones is a widely open problem. The particular mechanism in (a) belongs to class (i) introduced in Section 3.

options for the construction of polyhedral surfaces in equilibrium with forces only in edges, as needed for steel-glass structures [15]. Even if such approaches are based on very rough approximations of real loads, these designs can later be changed via optimization towards more realistic load cases. Such an approach is in the spirit of the present paper.

Beyond applications in connection with the Airy stress function, isotropic geometry has been employed for geometric approaches to functions defined by 2D images [20] or functions defined on surfaces [7].

Visual appearance of polyhedral surfaces. Static equilibrium of a planar truss is related to a stress surface with planar faces. Stress in the members is equivalent to the isotropic dihedral angles of the polyhedral stress surface. This implies that material-minimizing trusses have the smoothest polyhedral surfaces in isotropic space as stress surfaces [21]. They are discrete versions of isotropic principal parameterizations. Motivated by this result, which follows naturally from the pioneering work of [22], Pellis et al. [23] studied the visual smoothness of polyhedral surfaces in Euclidean geometry. With an appropriate smoothness measure, principal meshes yield highest visual smoothness. As in the isotropic case, surfaces of minimal total absolute curvature $\int(|\kappa_1|+|\kappa_2|)dA$ (κ_i being principal curvatures) yield the optimal shapes in terms of visual smoothness (see [24]). Isotropic geometry also played a key role in the design of polyhedral surfaces

with controlled roughness [10].

Laguerre sphere geometry. Laguerre geometry is one of the classical sphere geometries. It comes with several models, one of which plays in (appropriately extended) isotropic 3-space. There it appears as isotropic counterpart of Euclidean Möbius geometry. For example, a conical mesh is an object of Laguerre geometry and appears in the isotropic model as (isotropic) circular mesh [8]. Some problems, such as the study of rational curves and surfaces with rational offsets, are greatly simplified when using the isotropic model of Laguerre geometry [25]. Laguerre minimal surfaces become more easily accessible in the isotropic model, where they appear as graphs of biharmonic functions [9, 26]. Envelopes of moving cones or cylinders are well treated in Laguerre geometry and the isotropic model, leading to an application in CNC machining [27].

We now turn to a short outline of prior work on those applications which we later introduce to demonstrate the use of isotropic geometry for the initialization of difficult Euclidean optimization problems.

Quad mesh mechanisms. Studies of flexible polyhedral surfaces have a long history, and recently received growing interest as examples of transformable designs. We address here quad meshes, mostly with planar faces, which are mechanisms with rigid faces and rotational joints in the edges. The problem is algebraic in nature, but difficult. So far, only mechanisms with 3×3 planar faces have been classified [28], and understanding of individual classes is still in progress [29]. The composition to larger meshes is largely unclear, except for very special cases (see e.g. [30, 31, 32]). We develop a recent approach based on optimization [33], which contains a much more complete overview of prior work and a variety of strategies for initialization. In the present paper, we propose another such strategy; see Fig. 3 and Section 3.

Gridshells from straight lamellas. Eike Schling and his coworkers designed and fabricated gridshells by bending straight flat lamellas and arranging them in a quadrilateral grid such that the lamellas are orthogonal to an underlying reference surface S [34, 35]. Thus, in their final position the lamellas follow asymptotic curves on S. If lamellas meet at right node angles, S is a minimal surface. For another constant node angle γ, the surface S is a surface with constant ratio of principal curvatures $\kappa_1/\kappa_2 = \tan(\gamma/2) < 0$ (*CRPC surface*). This is equivalent to the constant ratio $H^2/K = -\cot^2\gamma$ of squared mean curvature and Gaussian curvature. One of the key advantages of such a gridshell is that they can be actuated to realize highly controlled transformable structures [36]. Only very special cases of such natural generalizations of minimal surfaces are explicitly known [37], but there is work on the computational design via optimization [38, 39]. We propose another strategy; see Fig. 4.

Many more open questions arise if one uses three or four families of lamellas and arranges them in form of a 3-web or 4-web, respectively. For such structures, some families of lamellas are tangential to S and thus follow geodesics on S. This leads to various types of these asymptotic-geodesic webs. Computational models based on discrete differential geometry have been proposed in [40, 41, 42], but an understanding of potential shapes and the initialization of optimization is still a challenge. We are not aware of any explicit

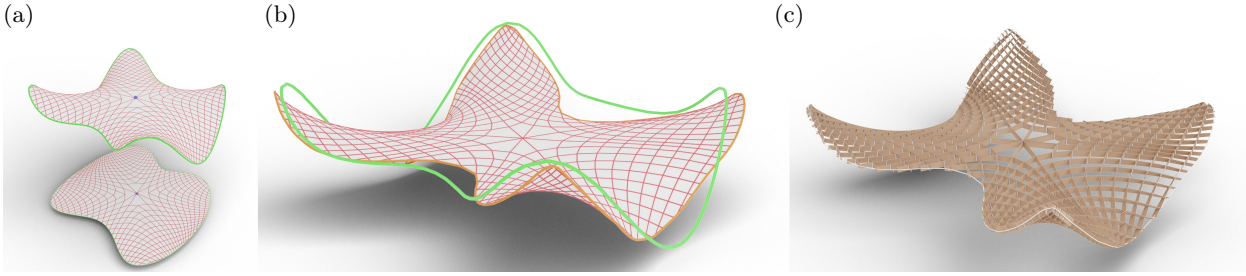


Figure 4: Design of an asymptotic gridshell with prescribed constant node angle (60°), boundary curve (green), and flat point (blue dot) by isotropic initialization. (a) The approximate isotropic CRPC surface constructed using Algorithm 4 and its top view. (b) The Euclidean CRPC surface with a constant angle 60° between the asymptotic polylines obtained by optimization initialized by the mesh in (a). (c) An asymptotic gridshell extracted from (b).

example of a surface which carries a non-trivial 4-web of type AGAG (see Figs. 2(b) and 10(b, f)), making even the computation via optimization a challenge due to a lack of reasonable initial guesses. Since discrete AGAG-webs have been classified in isotropic geometry [43], they will be used for that purpose (Section 4).

1.2. Overview and contributions

We discuss the initialization of optimization algorithms for the numerical solution of Euclidean geometric problems based on solutions of the isotropic counterparts. While the latter are expected to be easier to deal with, the solutions in isotropic geometry are not straightforward in the cases which we outline in the present paper.

After a short summary of some essential background from isotropic geometry (Section 2), we turn to quad mesh mechanisms in Section 3. Very recent work by [3] on a fruitful definition of isometric maps between surfaces in isotropic geometry allows one to define and fully characterize flexible quad meshes with planar faces in isotropic geometry [44]. The resulting isotropic mechanisms are already surprisingly close to the Euclidean ones and provide a good basis for computational design, also because many isotropic mechanisms fall into a class of meshes that has been studied previously for architectural applications [45].

In Section 4, we turn to gridshells that are formed by bending straight flat lamellas and geometrically require the computation of webs from geodesic and asymptotic curves [40]. 3-webs of geodesics can be easily computed in isotropic geometry based on a theorem by Graf and Sauer [46].

However, others are harder to deal with. We construct isotropic counterparts of 3-webs formed by two families of asymptotic and one family of geodesics. Fortunately, for AGAG webs, i.e., 4-webs formed by two geodesic and two asymptotic curve families, we can use the recent complete classification of their isotropic counterparts [43]. The AGAG case is the most challenging. To our knowledge, no smooth Euclidean AGAG webs are known. As for discrete AGAG webs, only a few numerical examples are shown by Schling et al. [40]; however, those examples are mostly flat and symmetric due to the lack of proper initialization. By our approach, we obtain complex Euclidean AGAG shapes (see Figs. 2, 10, 17).

Our final application deals with the computation of asymptotic gridshells with constant node angle, in particular those that approximate a given boundary curve; see Fig. 4 and Section 5. Geometrically, this amounts to the computation of a surface with a constant ratio of principal curvatures. In isotropic geometry, the problem is simplified by a novel approximation of such surfaces using complex analysis. Using this setup, we construct these surfaces with given flat points, either with a given boundary or without (see Figs. 1, 4, 13, 14). Here flat points are those points where more than two asymptotic lines can meet.

Details on the optimization are provided in Section 6 and their performance, choice of parameters, and a discussion of results and future work form the content of Section 7.

2. Preliminaries

2.1. Isotropic geometry

In *isotropic geometry*, the Euclidean norm $\|(x, y, z)\| = \sqrt{x^2 + y^2 + z^2}$ in space with the coordinates x, y, z is replaced with the *isotropic semi-norm* $\|(x, y, z)\|_i := \sqrt{x^2 + y^2}$. The *isotropic congruence transformations* are special affine transformations preserving the isotropic semi-norm:

$$\mathbf{x}' = A \cdot \mathbf{x} + \mathbf{b}, \quad A = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ c_1 & c_2 & 1 \end{pmatrix}$$

for some values of the parameters $\mathbf{b} \in \mathbb{R}^3$ and $\phi, c_1, c_2 \in \mathbb{R}$.

These transformations appear as Euclidean congruences in the projection onto the plane $z = 0$, which we call *top view*. The top view of a point P is denoted by $\overline{P}$. For instance, two planar quadrilaterals with the same top view (but not in one isotropic plane) are isotropically congruent. *Isotropic distances* between points, *isotropic angles* between lines, and *isotropic areas* of polygons are expressed through the isotropic norm in the usual way and appear in the top view as Euclidean distances, angles, and areas. Points P and Q with the same top view are called *parallel*. The isotropic distance $\|\overrightarrow{PQ}\|_i$ between them vanishes. Lines and planes parallel to the z-axis are called *isotropic*. See [47] for an elementary introduction.

One cannot define the isotropic angle between (non-isotropic) planes in the same way because their top views are just the same. Whenever a direct analog of Euclidean definition degenerates, one needs to find a replacing, nondegenerate one. Here the right definition of the *isotropic angle* is the difference of the slopes of the two planes. An isotropic plane is viewed as *orthogonal* to any non-isotorpic plane.

An isotropic sphere is the set of points at a fixed isotropic distance from a given point. It looks like a cylinder, hence called *isotropic sphere of cylindrical type*. There is another type: the surface of constant normal curvature $A \neq 0$, given by the equation $2z = A(x^2 + y^2) + Bx + Cz + D$ for some B, C, D. In a Euclidean interpretation, it is a paraboloid of revolution, hence called *isotropic sphere of parabolic type*.

In contrast to Euclidean geometry, isotropic geometry possesses a metric duality. It is defined as the polarity with respect to the *unit isotropic sphere* $2z = x^2 + y^2$, which maps a point (x^*, y^*, z^*) to the plane with the equation $z + z^* = xx^* + yy^*$, and vice versa. The point dual to a non-isotropic plane p is denoted by p^*, and vice versa. The isotropic angle between two non-isotropic planes equals the isotropic distance between the points dual to them.

Isotropic lines and planes play a special role and are usually excluded as tangent spaces in differential geometry. A surface is *admissible* if all its tangent planes are non-isotropic. An admissible surface can be locally represented as the graph of a function $z = f(x, y)$.

Since isotropic distances are measured in the top view, the shortest paths on the surface, *isotropic geodesics*, have straight top views. Thus isotropic geodesics are sections of the surface by isotropic planes.

The *isotropic Gauss map* takes a point on the surface to the point on the isotropic unit sphere such that the tangent planes at the two points are parallel. If this map is 1–1, then the oriented isotropic area of the image is called the *total isotropic Gaussian curvature* of the surface. The latter area equals the oriented isotropic area of the *metric dual surface* consisting of the points metric dual to the tangent planes of the original surface. The total curvature can also be computed as $\int K\,dxdy$, where $K := f_{xx}f_{yy} - f_{xy}^2$ is the *isotropic Gaussian curvature* of the surface $z = f(x, y)$.

2.2. Discrete differential geometry

We also use *discrete differential geometry* studying discrete analogs of curves and surfaces [48]. Now we introduce basic notions and elaborate on them in subsequent sections.

By a *discrete curve* we mean a collection of $m+1$ distinct points $f_i \in \mathbb{R}^3$ indexed by an integer $0 \le i \le m$. A discrete curve determines a polyline with the vertices f_i and edges (sides) $f_i f_{i+1}$.

By an $m \times n$ *net* (*discrete surface*) we mean a collection of $(m+1)(n+1)$ distinct points $f_{ij} \in \mathbb{R}^3$ indexed by two integers $0 \le i \le m$ and $0 \le j \le n$. An $m \times n$ net determines a quad mesh with regular combinatorics: points f_{ij} are *vertices*, segments of the form $f_{ij}f_{i+1,j}$ and $f_{ij}f_{i,j+1}$ are *edges*, and polylines of the form $f_{ij}f_{i+1,j}f_{i+1,j+1}f_{i,j+1}$ are *faces*. A vertex f_{ij} with $i \in \{0, m\}$ or $j \in \{0, n\}$ is a *boundary* vertex. The polylines $f_{i0} \dots f_{in}$ for $0 \le i \le m$ and $f_{0j} \dots f_{mj}$ for $0 \le j \le n$ are called *discrete parameter lines* (namely, *i-lines* and *j-lines* respectively).

In general, faces do not need to be planar. If all the faces are planar (but none is contained in a line), then the $m \times n$ net is called a *Q-net*, and the faces are referred to as *quadrilaterals* (although they can have self-intersections). See Fig. 5.

3. Flexible polyhedral nets

The first application of our method is the construction of quad-mesh mechanisms, also known as flexible nets. We are mostly interested in flexible polyhedral nets, i.e. flexible Q-nets. We build on the classification

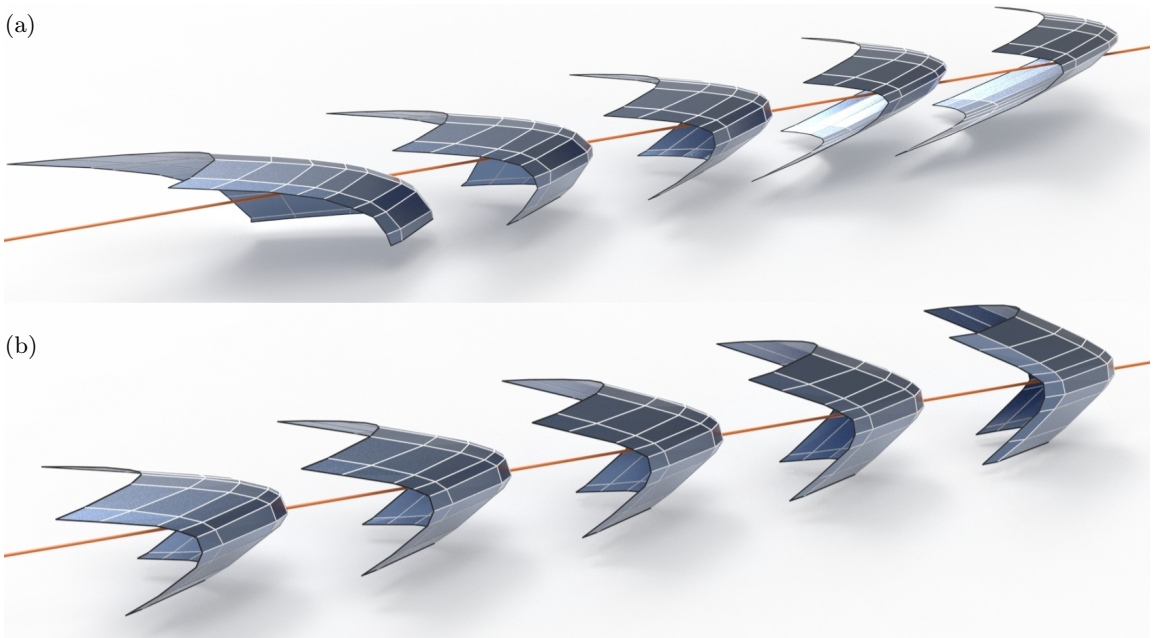


Figure 5: (a) A Q-net that is flexible in isotropic geometry. A few positions of the isotropic mechanism are shown. The line depicts the isotropic direction. The net belongs to class (ii) described in Section 3. A single position of the isotropic mechanism (a) is used to initialize optimization, leading to the Euclidean mechanism (b). Although the optimization does not change the shape of this position much, the flexion leads to different shapes.

of flexible Q-nets in isotropic geometry from [44] (see Figs. 3 and 5). Let us briefly state this classification.

First, we need to define what *is* a flexible net in isotropic geometry. Even this step is non-trivial and has only recently been done [3]. Only requiring rigid faces (*face condition*) and rotational joints in the edges is not enough in isotropic geometry; otherwise, *any* net would be flexible. The insight comes from the famous Gauss Theorema Egregium, saying that flexion of smooth surfaces preserves the Gaussian curvature. So it is natural to additionally require preservation of curvature at the vertices (*vertex condition*).

This curvature is defined as follows. For an $m \times n$ Q-net f_{ij}, denote by p_{ij} the plane of the face $f_{ij}f_{i+1,j}f_{i+1,j+1}f_{i,j+1}$. Consider the planes $p_1 := p_{i-1,j-1}, p_2 := p_{i,j-1}, p_3 := p_{ij}, p_4 := p_{i-1,j}$ of consecutive faces around a particular non-boundary vertex f_{ij} (see Fig. 6(a)). The top views of the points metric dual to those planes form a quadrilateral $\overline{p_1^*}\,\overline{p_2^*}\,\overline{p_3^*}\,\overline{p_4^*}$ called the *top view of the face metric dual to the vertex* f_{ij} (see Fig. 6(b)). The oriented area of this quadrilateral is called the *discrete isotropic Gaussian curvature*, or simply the *curvature at* f_{ij}:

$$\Omega(f_{ij}) := \text{Area}\left(\overline{p_1^*}\,\overline{p_2^*}\,\overline{p_3^*}\,\overline{p_4^*}\right) = \frac{1}{2}\sum_{j=1}^{4} \det\left(\overline{p_j^*}, \overline{p_{j+1}^*}\right), \quad \overline{p_5^*} := \overline{p_1^*}. \tag{1}$$

For instance, if the i-line through f_{ij} is straight (and does not contain $f_{i\pm1,j}$; see Fig. 7(a)), then $\overline{p_1^*} = \overline{p_4^*}$, $\overline{p_2^*} = \overline{p_3^*}$, and hence $\Omega(f_{ij}) = 0$.

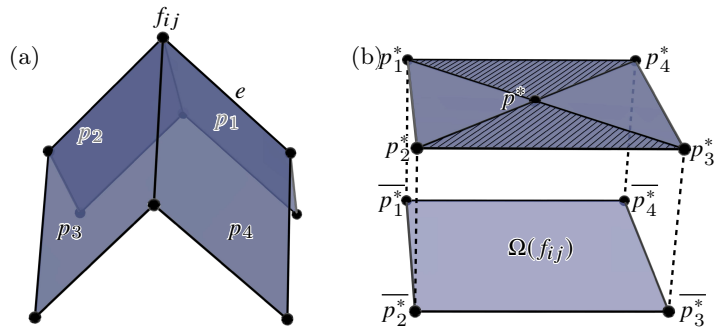


Figure 6: (a) Four consecutive faces around a vertex f_{ij} of an $m \times n$ Q-net and their planes p_1, p_2, p_3, p_4. (b) Their metric duals form a quadrilateral $p_1^* p_2^* p_3^* p_4^*$ (a face of so-called *metric dual net*). The curvature $\Omega(f_{ij})$ at f_{ij} is the oriented area of the top view $\overline{p_1^*}\,\overline{p_2^*}\,\overline{p_3^*}\,\overline{p_4^*}$. The *opposite ratio* of f_{ij} with respect to the edge e in (a) is the ratio of the areas of the shaded triangles $p^* p_1^* p_4^*$ and $p^* p_2^* p_3^*$ in (b). Here, the edge e determines the order of faces to define the opposite ratio.

An *isotropic isometric deformation* of an $m \times n$ Q-net f_{ij} is a continuous family of $m \times n$ Q-nets $f_{ij}(t)$, where $t \in [0, 1]$ and $f_{ij}(0) = f_{ij}$, satisfying

face condition: corresponding faces of all $m \times n$ Q-nets $f_{ij}(t)$ are isotropically congruent; and

vertex condition: corresponding non-boundary vertices of all $m \times n$ Q-nets $f_{ij}(t)$ have the same curvatures.

An isotropic isometric deformation is *trivial* if for each $t \in [0, 1]$ there is an isotropic congruence C_t such that $f_{ij}(t) = C_t(f_{ij})$ for each $0 \le i \le m, 0 \le j \le n$. The Q-net f_{ij} is an *isotropic flexible net* if it has a nontrivial isotropic isometric deformation; see Fig. 7(a).

All isotropic flexible nets were classified in the series of papers [49, 44]. Namely, under minor convexity assumptions, an $m \times n$ Q-net is an isotropic flexible net if and only if one of the following conditions holds:

(i) n lines $p_{k,0} \cap p_{k+2,0}, \ldots, p_{k,n-1} \cap p_{k+2,n-1}$ lie in one isotropic plane for each $0 \le k \le m-3$ or m lines $p_{0,l} \cap p_{0,l+2}, \ldots, p_{m-1,l} \cap p_{m-1,l+2}$ lie in one isotropic plane for each $0 \le l \le n-3$;

(ii) any two non-boundary vertices joined by an edge have equal opposite ratios with respect to the edge.

Here the *opposite ratio* is defined in Fig. 6 and condition (i) on face planes p_{ij} is illustrated in Fig. 7(b). The nets in resulting classes (i) and (ii) can be efficiently constructed; see examples in Figs. 3, 5, and 8.

This classification has sharp contrast to Euclidean geometry, where even for 3×3 flexible nets, there is a huge number of classes.

An example of an isotropic flexible net is a *generalized T-net*. It is defined by the condition that the parameter lines are planar, and one family of parameter lines lies in isotropic planes. In other words, i-lines are planar, and j-lines are (discrete) isotropic geodesics. Planar parameter lines are a favorite in architecture, and isotropic planes have advantages for supporting structures [45]. Generalized T-nets belong to class (i);

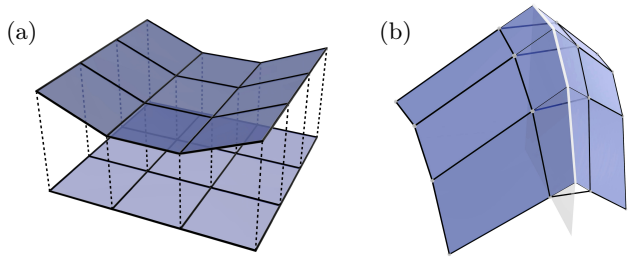


Figure 7: (a) A planar 3×3 net with square faces is flexible in isotropic geometry. Another position of the same isotropic mechanism is shown above. The top views of the vertices remain fixed (implying face condition), and i-lines remain straight (implying vertex condition). (b) An isotropic flexible net of class (i). The intersections (white lines) of the planes of every other face lie in the same isotropic plane (gray). Under some convexity assumptions, this condition implies that the extensions (thin lines) of every other edge on j-lines intersect. Extending every other face until they meet, we construct a generalized T-net. The initial net is obtained from the resulting one by "truncation of edges".

hence they are isotropic flexible nets. Conversely, most nets of class (i) can be obtained from a generalized T-nets by truncation of edges (the white edges in Fig. 7(b)).

Generalized T-nets include *T-nets*, a class of flexible nets in Euclidean geometry introduced by Graf and Sauer [30] and now actively studied [50], as a particular case, hence the name. Thus, T-nets are flexible in both Euclidean and isotropic geometries (after an appropriate Euclidean rotation). But the class of generalized T-nets is much larger: in particular, their faces are not necessarily trapezoids.

Another particular case of generalized T-nets is *isotropic Voss nets*, i.e., Q-nets with all parameter lines being isotropic geodesics (see the special case in Fig. 7(a)). They are an isotropic analog of *Voss nets*, a class of Euclidean flexible nets introduced by Voss [30].

Generalized T-nets are particularly simple to design.

They are known to be metric dual to *cone-cylinder nets* given by the equation

$$P_{ij} = a_i + \sigma_i b_j, \qquad 0 \le i \le m+1 \text{ and } 0 \le j \le n+1, \tag{2}$$

for some vectors $a_0, \dots, a_{m+1}, b_0, \dots, b_{n+1} \in \mathbb{R}^3$ and real numbers $\sigma_0, \dots, \sigma_{m+1} \neq 0$. One can choose these parameters arbitrarily and apply metric duality to get a generalized T-net [44]

$$f_{ij} = -\frac{1}{\det(e_3, b_{j+1} - b_j, \Delta_{ij})} \begin{pmatrix} \det(e_1, b_{j+1} - b_j, \Delta_{ij}) \\ \det(e_2, b_{j+1} - b_j, \Delta_{ij}) \\ \det(a_i + \sigma_i b_j, b_{j+1} - b_j, \Delta_{ij}) \end{pmatrix},$$

where $e_1 = (1,0,0)$, $e_2 = (0,1,0)$, $e_3 = (0,0,1)$, and

$$\Delta_{ij} := a_{i+1} - a_i + b_j(\sigma_{i+1} - \sigma_i).$$

The only restriction is a non-zero denominator $\det(e_3, b_{j+1} - b_j, \Delta_{ij}) \neq 0$. Alternatively, one can apply the general design method for nets with planar parameter lines from [45, Section 3.1.2].

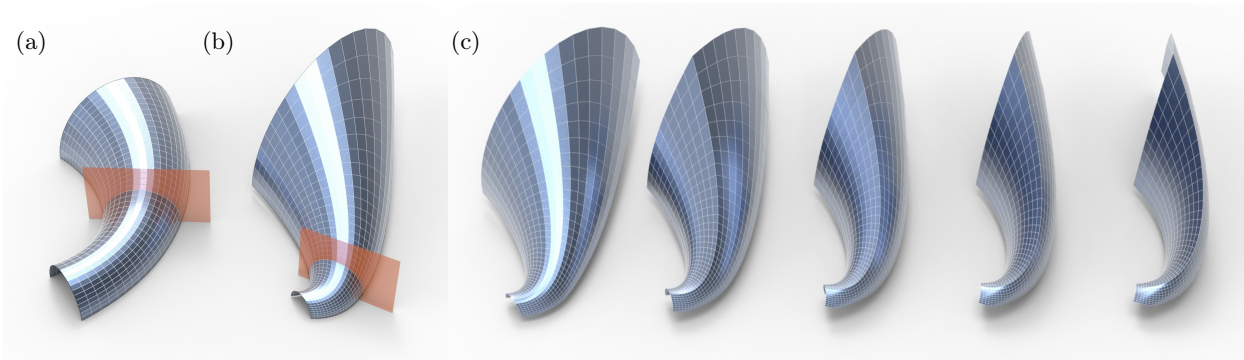


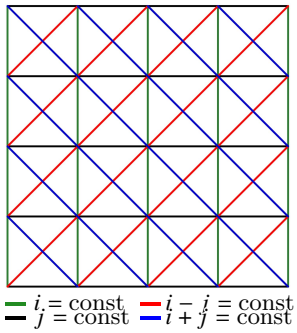
Figure 8: (a) A generalized T-net, flexible in isotropic geometry. The plane containing one of the parameter lines is in orange. This is a particular case of class (i) of isotropic flexible nets. Applying a projective transformation preserving the isotropic direction, we get another isotropic flexible net (b). Optimization leads to a Euclidean mechanism, with a few positions shown in (c). We observe a crease appearing during flexion of an originally smooth-looking mesh; cf. an analogous behavior of T-nets [50, Figure 18]. Such a crease may eventually lead to a collapse, but the mechanism is freely flexible within a large interval.

With a variety of isotropic flexible nets at hand, we optimize them towards Euclidean ones using the framework of [33]; see Fig. 3. Surprisingly, the optimization leads to very little change in the shape. The resulting isotropic and Euclidean mechanisms have almost the same initial positions, but very different isometric deformations. Every position of an isotropic flexible net can be used to initialize optimization and thus leads to a different Euclidean mechanism. We can get even more Euclidean mechanisms by applying an isotropic motion or, more generally, a projective transformation preserving the z-direction (which preserves the isotropic flexibility); see Fig. 8.

These results lead to several new observations. Although generalized T-nets, as well as all isotropic flexible nets of class (i), have a family of planar parameter lines, those do, in general, not remain planar during Euclidean flexion. Therefore, we expect no new classes of Euclidean flexible $m \times n$ nets with planar parameter lines, analogous to class (i) in isotropic geometry, which shows the advantage of our optimization approach. In the mechanism of Fig. 3(b), the originally almost planar parameter lines do not remain planar during Euclidean flexion. However, for the parameter lines originally lying in isotropic planes, the deviation from planarity during Euclidean flexion is small. Interestingly, faces stay planar with high numerical precision even if we optimize towards a mechanism with rigid, not necessarily planar faces and do not include the planarity of faces as objective in our optimization algorithm; cf. [33]. Furthermore, in the mechanisms of Figs. 3(b) and 8(c), we observe that all 3×3 sub-nets belong to one of the *equimodular* or *linear elliptic/conic coupling* types of Izmestiev's classification [28, Secs. 3.3, 3.5.4, and 3.5.5]. Since Izmestiev's classes form a complicated variety with components of different dimensions in the parameter space, the precision of the numerical results does not allow us to distinguish between the *elliptic equimodular class* and its limiting cases, such as *linear elliptic/conic coupling*. The latter seems the most probable candidate; cf. [31, Figure 8].

4. Asymptotic-geodesic webs

Now we present the second application of our method: the construction of asymptotic-geodesic webs.



Denote by $f(u, v)$, a parametric representation of a smooth surface. The net of parameter lines u = const (*u-lines*) and v = const (*v-lines*), extended by the diagonal curves $u - v$ = const ($(u - v)$-*lines*), is called a *smooth* 3-*web (hexagonal web)* on the surface [46]. Extending this 3-web by the other diagonal curves $u + v$ = const ($(u + v)$-*lines*) yields a *smooth* 4-*web* on the surface.

Consider a discrete surface f_{ij} of size $n \times n$. The discrete parameter lines i = const and j = const together with the diagonal polylines joining the points f_{ij} with $i - j$ = const (*discrete* $(i - j)$-*lines*), is called a *discrete* 3-*web*. By further extending the discrete 3-web with the other diagonal polylines joining the points f_{ij} with $i + j$ = const (*discrete* $(i + j)$-*lines*), one obtains a *discrete* 4-*web* (see the inset). Here all four constants as well as i and j are integers. The 3- or 4-webs are still denoted by f_{ij}.

Our interest is in webs that consist of asymptotic and geodesic curves on a surface, referred to as *asymptotic-geodesic webs*. We discuss three types of them, known as GGG, AAG, and AGAG webs [42, 40]. We do it first in Euclidean geometry.

Recall that a biregular curve $C(t)$ (that is, a curve such that $C'(t)$ and $C''(t)$ are linearly independent for each t) on a smooth surface is *asymptotic* if the osculating plane (spanned by $C'(t)$ and $C''(t)$) is tangent to the surface at each point of the curve. A biregular curve is a *geodesic* if the osculating plane is orthogonal to the surface at each point of the curve. A *smooth asymptotic* (respectively, *geodesic*) *net* is a regular parametrized surface $f(u, v)$ such that the parameter lines are asymptotic curves (respectively, geodesics).

The definitions in the discrete case are aligned with these properties. Consider a discrete curve C with the vertices $f_0, f_1, \cdots, f_k$ that are also vertices of a discrete surface f_{ij}. The edge $f_i f_{i+1}$ is called the *discrete tangent* to C at f_i and the plane spanned by f_{i-1}, f_i, and f_{i+1} is the *discrete osculating plane* at f_i if $i \neq 0, k$, and the three points are not collinear [30]. The unit normal $\frac{(f_i - f_{i-1}) \times (f_i - f_{i+1})}{|(f_i - f_{i-1}) \times (f_i - f_{i+1})|}$ to this plane is the *discrete binormal vector* at f_i.

A *discrete normal* vector at a non-boundary vertex f_{ij} is a unit vector orthogonal to the two discrete tangents of i-lines and j-lines at f_{ij}, and the *discrete tangent plane* is the one spanned by those edges. Hereafter, assume that those edges indeed span a plane.

The discrete curve C is a *discrete asymptotic curve* if, at each point f_i of C besides the endpoints and the boundary points of discrete surface, the three neighbors f_{i-1}, f_i, and f_{i+1} are collinear or the discrete osculating plane is the discrete tangent plane. The curve C is a *discrete geodesic* if, at each non-boundary point f_i of C besides the endpoints, the three neighbors f_{i-1}, f_i, and f_{i+1} are collinear or the osculating plane is orthogonal to the tangent plane.

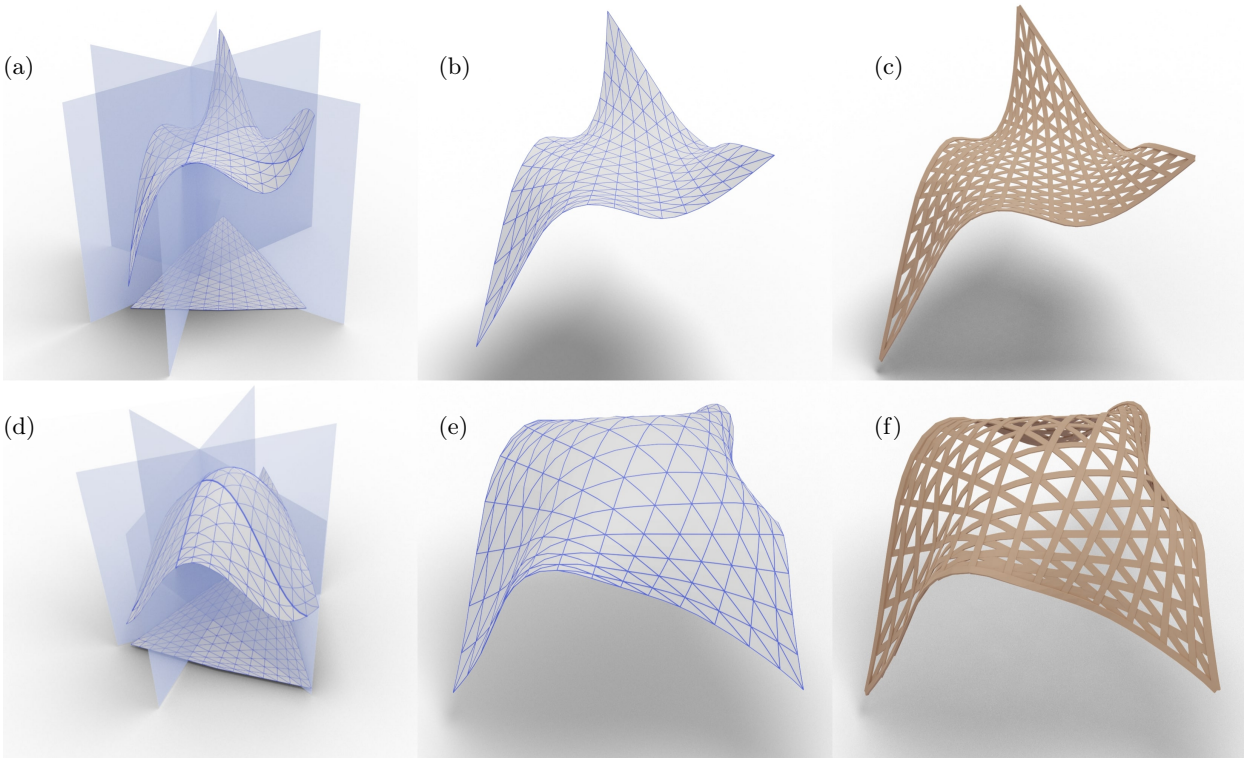


Figure 9: Design of Euclidean GGG webs by optimization of isotropic ones. Starting with a planar web formed by tangents to a class 3 curve and lifting it to different surfaces, we obtain different isotropic GGG webs (a, d). The meshes were downsampled by removing some polylines to reduce density while keeping their overall shape. Transparent planes are isotropic and demonstrate that the blue polylines are indeed isotropic geodesics. Optimization leads to Euclidean GGG webs (b, e). We get gridshells (c, f).

If the discrete parameter lines $i = \text{const}$ and $j = \text{const}$ are discrete asymptotic curves, then f_{ij} is a *discrete asymptotic net (A-net)*. Equivalently, for all $0 < i, j < n$, the five vertices $f_{ij}, f_{i-1,j}, f_{i+1,j}$, $f_{i,j-1}$, and $f_{i,j+1}$ are coplanar [48]. In the construction of A-nets below, we also impose the same condition at the boundary, i.e., for $i, j \in \{0, n\}$, with some of those five vertices omitted (although it is not required by the definition).

We consider the following types of webs (discrete or smooth):

GGG: A 3-web on a surface such that its three families of curves are geodesics (discrete or smooth).

AAG: A 3-web on a surface such that the parameter lines $i = \text{const}$ and $j = \text{const}$ are asymptotic curves, and the curves $i - j = \text{const}$ are geodesics.

AGAG: A 4-web on a surface such that the parameter lines $i = \text{const}$ and $j = \text{const}$ are geodesics, and the diagonal curves $i - j = \text{const}$ and $i + j = \text{const}$ form one (in the smooth case) or two (in the discrete case) A-nets.

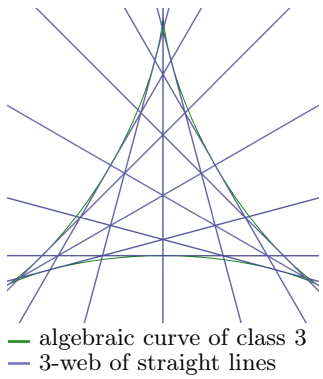
Note that in the discrete case, we get two A-nets (instead of one), consisting of points f_{ij} with $i - j$ even and

odd respectively. The resulting two collections of points are still called A-nets, although different diagonal curves consist of different numbers of points. Beware that the discrete curves $i - j = \text{const}$ and $i + j = \text{const}$ are *not* necessarily discrete asymptotic curves on the original discrete surface; they only become such after omitting the points f_{ij} with one parity of $i - j$. The assignment of curve families to parameter lines versus diagonals is motivated by the construction of the isotropic counterparts (see Sections 4.2 and 4.3).

We aim to transform webs from isotropic to Euclidean geometry through numerical optimization. Let us define the isotropic counterparts of our webs. For that, we need to restrict ourselves to *admissible* surfaces (discrete or smooth), that is, the ones having no isotropic tangent planes. Asymptotic curves are a projective geometry concept and remain the same in both Euclidean and isotropic geometry. However, geodesics are much simpler in isotropic geometry: they are just (discrete or smooth) curves on the surface whose top views are straight line segments. One defines *GGG, AAG, and AGAG webs* in isotropic geometry by replacing the Euclidean geodesics with their isotropic counterparts.

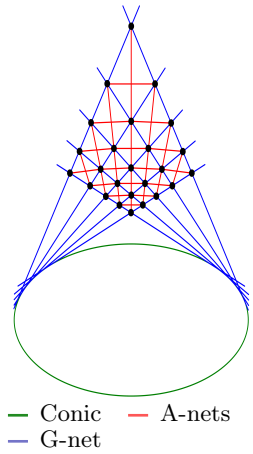
4.1. Isotropic GGG webs

Thus, an isotropic GGG (discrete or smooth) web is a 3-web on an admissible surface whose top view is a 3-web of straight lines. Graf and Sauer [46] showed that any 3-web of straight lines is formed by the tangents of an algebraic curve of class 3 (see the inset). In the same work, they also provided a way to construct a discrete 3-web of straight lines. Thus, one can obtain a discrete isotropic GGG web by projecting the 3-web of straight lines onto a surface, see Fig. 9 (a, d).



4.2. Isotropic AGAG webs

An isotropic AGAG web is a 4-web on an admissible surface such that the i- and j-lines are geodesics (and thus form a so-called *G-net*), and the two families of diagonal curves $i - j = \text{const}$ and $i + j = \text{const}$ form two A-nets. Such webs were characterized by [43]: the top view of the geodesics lie on two families of straight lines tangent to the same conic (see the inset), and the A-net is constructed by lifting the two diagonal nets as described in Theorem 8 of [43]; see Figs. 10 (a, e) and 2 (a).



4.3. Isotropic AAG Webs

A discrete isotropic AAG web f_{ij} of size $n \times n$ consists of $2n + 2$ asymptotic curves (i- and j-lines) and $2n + 1$ isotropic geodesics ($(i - j)$-lines) denoted by $D_0, \ldots, D_{2n}$ (D_0 and D_{2n} consist of a single point each). Denote by $\bar{D}_0, \ldots, \bar{D}_{2n}$ the straight lines that contain the top view of $D_0, \ldots, D_{2n}$, respectively. To design

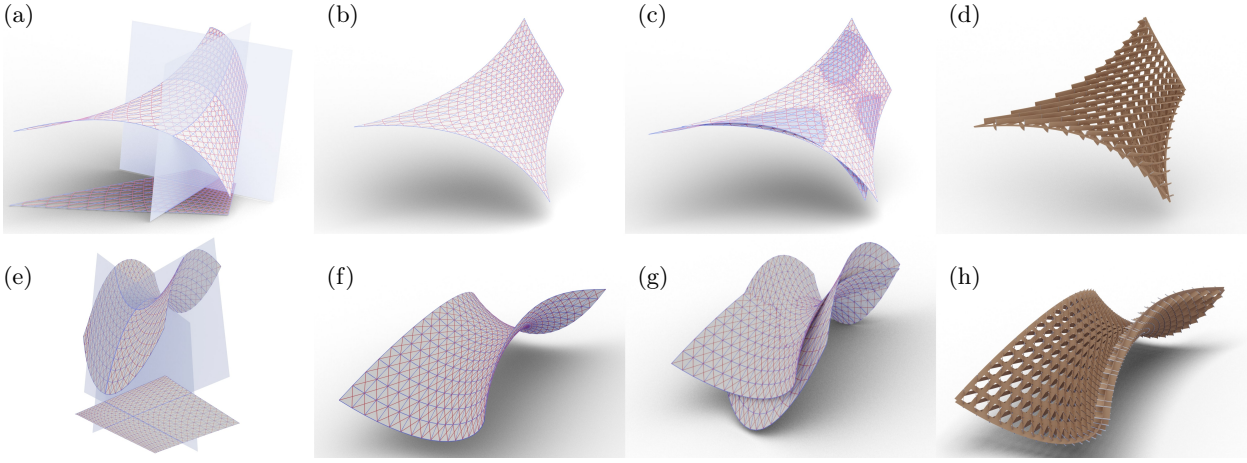
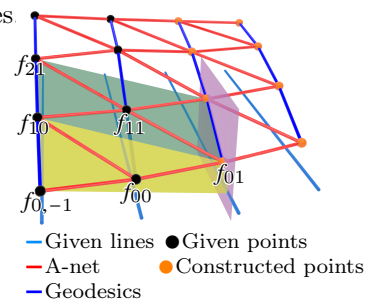


Figure 10: Design of Euclidean AGAG webs by optimization of isotropic AGAG webs. Starting with a planar web formed by conic tangents (blue) and diagonal polylines (red), we construct an isotropic AGAG web (a). The blue polylines are part of the G-net, and the red polylines belong to the diagonal A-nets. The meshes were downsampled by removing some red and blue polylines to reduce density. Transparent planes are isotropic and demonstrate that the blue polylines are indeed isotropic geodesics. Optimization leads to a Euclidean AGAG web (b). The Euclidean (solid) and isotropic (transparent) webs are compared in (c). Dropping even more polylines, we extract a gridshell (d). Applying a projective transformation preserving the isotropic direction to (a), we get another isotropic AGAG web (e). Optimization leads to a Euclidean AGAG web (f) and a gridshell (h).

the web, we propose two methods that work for a "general position" input. (We omit the proofs and the definition of "general position"; cf. [27].)

The first method is via propagation (see Algorithm 1, which is run twice: for the plus and the minus sign in all "±", respectively). We prescribe $2n+1$ lines $\bar{D}_0, \ldots, \bar{D}_{2n}$, the points $f_{00}, f_{11}, \ldots, f_{nn}$ (with the top view on $\bar{D}_n$) and $f_{0,-1}, f_{1,0}, \ldots, f_{n+1,n}$ (with the top view on $\bar{D}_{n+1}$). Here $f_{0,-1}$ and $f_{n+1,n}$ are just auxiliary points, they are not contained in the web but are only used to determine the tangent planes at f_{00} and f_{nn}. For instance, one can take two space curves with the top views on $\bar{D}_n$ and $\bar{D}_{n+1}$, respectively, and pick $n+1$ points $f_{00}, \ldots, f_{nn}$ and $n+2$ points $f_{0,-1}, f_{1,0}, \ldots, f_{n+1,n}$ uniformly on the curves.



First, we construct the vertex f_{01}. For an A-net, it must lie on the discrete tangent planes at f_{00} and f_{11}, and those planes are spanned by triples $f_{00}, f_{0,-1}, f_{1,0}$ and f_{11}, f_{10}, f_{21} respectively. The top view of f_{01} must lie on $\bar{D}_{n-1}$, hence f_{01} lies on the isotropic plane through $\bar{D}_{n-1}$. Thus f_{01} is the intersection of these three planes; see the inset.

To construct f_{01}, it suffices to compute the discrete normals at f_{00} and f_{11}; see Step 4 of Algorithm 1.

Similarly to f_{01}, we construct $f_{12}, \ldots, f_{n-1,n}$, using the normals along the diagonal D_n; see Step 6 of Algorithm 1.

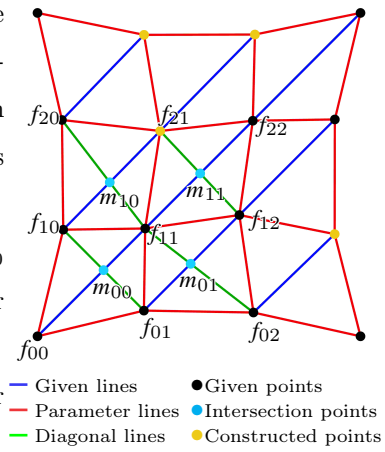
The rest of the construction follows this logic: we determine the vertices along the diagonal $D_{n\pm i}$ using the normals along the neighboring diagonals $D_{n\pm(i-1)}$. Then, we compute the normals along $D_{n\pm i}$. We repeat this process until the entire net f_{ij} is constructed. See Fig. 11.

The second construction method uses planar Kœnigs nets. Given a Q-net f_{ij}, denote by m_{ij} the intersection of the diagonals of the quadrilateral $f_{ij}f_{i,j+1}f_{i+1,j+1}f_{i+1,j}$. The net f_{ij} is called a *Kœnigs* net if there exist real numbers ν_{ij}, such that for each $0 \le i, j \le n$

$$\frac{f_{ij} - m_{ij}}{\nu_{ij}} = \frac{f_{i+1,j+1} - m_{ij}}{\nu_{i+1,j+1}} \quad \text{and} \quad \frac{f_{i,j+1} - m_{ij}}{\nu_{i,j+1}} = \frac{f_{i+1,j} - m_{ij}}{\nu_{i+1,j}}. \tag{3}$$

In Theorem 8 of [43], it is proved that for any planar Kœnigs net f_{ij}, there is an A-net whose top view is f_{ij}, and a construction of such A-net is provided. Thus, to construct an isotropic AAG web, it suffices to construct a planar Kœnigs net whose $(i-j)$-lines are straight. This is done as follows; see Algorithm 2.

We prescribe $2n+1$ diagonal lines $\bar{D}_0, \dots, \bar{D}_{2n}$ in the plane and the boundary vertices $f_{0n}, f_{0,n-1}, \dots, f_{01}, f_{00}, f_{10}, \dots, f_{n0}$ on $\bar{D}_0, \dots, \bar{D}_{2n}$, respectively (see the inset). Additionally, we prescribe vertices $f_{11}, f_{22}, \dots, f_{nn}$ on the diagonal $\bar{D}_n$ and $f_{12}, f_{23}, \dots, f_{n-1,n}$ on the diagonal $\bar{D}_{n-1}$, and numbers $\nu_{00}, \nu_{01} \neq 0$.



We construct a planar Kœnigs net f_{ij} by propagation. First, we find m_{00} as the intersection of $\bar{D}_n$ with $f_{10}f_{01}$. Then compute ν_{10}, ν_{11} from (3) for $i = j = 0$.

Next, we find m_{10} and m_{01} and compute ν_{20}, ν_{02}, and ν_{12} from (3) for $i + j = 1$.

Next, we compute the triple $(f_{21}, m_{11}, \nu_{21})$ as follows: we have two equations from (3) for $(i, j) = (1, 0)$ and $(1, 1)$, and a third equation arises from the fact that m_{11} lies on the line $\bar{D}_n$. We solve this system of equations at Step 10 of Algorithm 2. Finally, from (3) for $i = j = 1$, we get ν_{22}. To continue, assume that the numbers ν_{ij} and the $s \times s$ net, consisting of the vertices f_{ij} for $0 \le i, j \le s$, have been constructed. To extend it to a net of size $(s+1) \times (s+1)$, we find m_{s0} and m_{0s} and then compute $\nu_{s+1,0}, \nu_{0,s+1}$ from (3) for $(i, j) = (0, s)$ and $(s, 0)$.

Similarly to the triple $(f_{21}, m_{11}, \nu_{21})$ we compute

$$(f_{s+1,1}, m_{s,1}, \nu_{s+1,1}), \dots, (f_{s+1,s-1}, m_{s,s-1}, \nu_{s+1,s-1})$$

and

$$(f_{1,s+1}, m_{1,s}, \nu_{1,s+1}), \dots, (f_{s-1,s+1}, m_{s-1,s}, \nu_{s-1,s+1})$$

at Steps 7 and 10 of Algorithm 2. Then, from (3) for $i = s-1$ and $j = s$, we obtain $\nu_{s,s+1}$. We compute $(f_{s+1,s}, m_{s,s}, \nu_{s+1,s})$. Finally, from (3) for $i = j = s$, we obtain $\nu_{s+1,s+1}$.

Thus, the $(s+1)\times(s+1)$ net is fully constructed. Repeating this process iteratively generates the net f_{ij}. We note that the stability of the propagation depends primarily on the configuration of the prescribed initial diagonal lines $\bar{D}_0,\ldots,\bar{D}_{2n}$ in the plane. To ensure regularity, it is essential to prevent intersections between the adjacent diagonal lines within the surface region. A small difference $|k_{l+1}-k_l|$ combined with a relatively large $|b_{l+1}-b_l|$ in Algorithm 2 yields a sufficiently large Kœnigs net for surface design. See Fig. 12

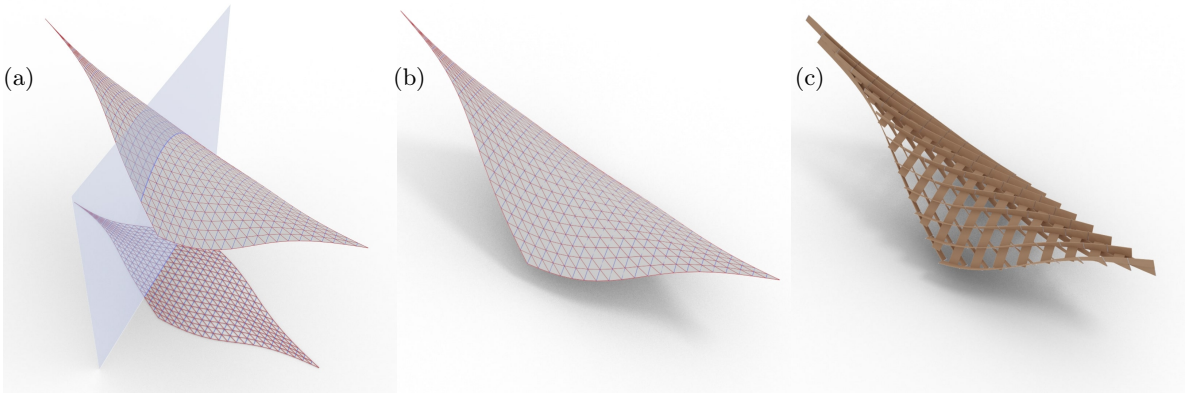


Figure 11: Design of Euclidean AAG webs by optimization of isotropic ones. We construct an isotropic AAG web (a) by propagation (see Algorithm 1). Blue and red polylines represent geodesics and asymptotic curves. Optimization leads to a Euclidean AAG web (b) and downsampling gives a gridshell (c).

Algorithm 1: Design of isotropic AAG webs by propagation

Input: $2n+1$ lines $\bar{D}_l : y = k_l x + b_l$ for $l = 0,\ldots,2n$. Points $f_{00}, f_{11},\cdots,f_{nn}$ (with the top view on $\bar{D}_n$) and $f_{0,-1}, f_{1,0},\cdots,f_{n+1,n}$ (with the top view on $\bar{D}_{n+1}$).

Output: A discrete isotropic AAG web f_{ij} of size $n\times n$ containing the given points such that the top views of $(i-j)$-lines are on the given lines $\bar{D}_0,\ldots,\bar{D}_{2n}$.

for $l \leftarrow 0$ ***to*** n **do**
  for $i \leftarrow 0$ ***to*** n **do**
    if $i \pm l \leq n$ *and* $i \pm l \geq 0$ **then**
      if $l(l\pm 1) > 0$ **then**
        Compute $f_{i,i\pm l}$ solving the system $\begin{cases} \mathbf{n}_{i,i\pm(l-1)} \cdot (f_{i,i\pm l} - f_{i,i\pm(l-1)}) = 0, \\ \mathbf{n}_{i\pm 1,i\pm l} \cdot (f_{i,i\pm l} - f_{i\pm 1,i\pm l}) = 0, \\ y_{i,i\pm l} = k_{n\mp l} x_{i,i\pm l} + b_{n\mp l}\,; \end{cases}$
      if $l \pm 1 \geq 0$ **then**
        $\mathbf{n}_{i,i\pm l} \leftarrow (f_{i,i\pm l} - f_{i,i\pm(l-1)}) \times (f_{i,i\pm l} - f_{i\pm 1,i\pm l})$;

Return the web f_{ij} ;

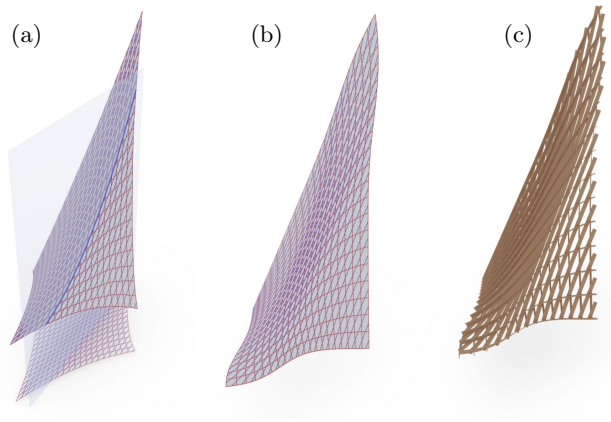


Figure 12: Another design method for Euclidean AAG webs based on Algotithm 2. Starting with a planar Kœnigs net (red) with one family of diagonals forming straight lines (blue), we construct an isotropic AAG web (a). Optimization leads to a Euclidean AAG web (b) and a gridshell (c).

Algorithm 2: Design of planar Kœnigs nets

Input: $2n+1$ lines $\bar{D}_l : y = k_l x + b_l$ for $l = 0, \ldots, 2n$. Points $f_{0,n}, f_{0,n-1}, \cdots, f_{0,0}, f_{1,0}, f_{2,0}, \cdots, f_{n,0}$ on $\bar{D}_0, \ldots, \bar{D}_{2n}$, respectively, $f_{00}, f_{11}, \cdots, f_{nn}$ on $\bar{D}_n$, and $f_{0,1}, f_{1,2}, \cdots, f_{n-1,n}$ on $\bar{D}_{n-1}$ and numbers $\nu_{00}, \nu_{01} \neq 0$.

Output: A planar Kœnigs net f_{ij} of size $n \times n$ containing the given points and such that $(i-j)$-lines are on the given lines $\bar{D}_0, \ldots, \bar{D}_{2n}$

for $s \leftarrow 0$ ***to*** $n-1$ **do**

  $m_{s,0} \leftarrow \bar{D}_{n+s} \cap f_{s+1,0} f_{s,1}$ and $m_{0,s} \leftarrow \bar{D}_{n-s} \cap f_{0,s+1} f_{1,s}$;

  $\nu_{s+1,0} \leftarrow \nu_{s,1}(f_{s+1,0} - m_{s,0})/(f_{s,1} - m_{s,0})$;

  $\nu_{0,s+1} \leftarrow \nu_{1,s}(f_{0,s+1} - m_{0,s})/(f_{1,s} - m_{0,s})$;

  for $i \leftarrow 1$ ***to*** s **do**

    if $i < s$ **then**

      Compute $f_{i,s+1}$, $m_{i,s} = (x_{i,s}, y_{i,s})$, $\nu_{i,s+1}$ from the system

$$\begin{cases} \nu_{i-1,s}(f_{i,s+1} - m_{i-1,s}) = \nu_{i,s+1}(f_{i-1,s} - m_{i-1,s}), \\ \nu_{i+1,s}(f_{i,s+1} - m_{i,s}) = \nu_{i,s+1}(f_{i+1,s} - m_{i,s}), \\ y_{i,s} = k_{n+i-s} x_{i,s} + b_{n+i-s} \; ; \end{cases}$$

    else

      $\nu_{i,s+1} \leftarrow \nu_{i-1,s}(f_{i,s+1} - m_{i-1,s})/(f_{i-1,s} - m_{i-1,s})$;

    Compute $f_{s+1,i}$, $m_{s,i} = (x_{s,i}, y_{s,i})$, $\nu_{s+1,i}$ from the system

$$\begin{cases} \nu_{s,i-1}(f_{s+1,i} - m_{s,i-1}) = \nu_{s+1,i}(f_{s,i-1} - m_{s,i-1}), \\ \nu_{s,i+1}(f_{s+1,i} - m_{s,i}) = \nu_{s+1,i}(f_{s,i+1} - m_{s,i}), \\ y_{s,i} = k_{n+s-i} x_{s,i} + b_{n+s-i} \; ; \end{cases}$$

  $\nu_{s+1,s+1} \leftarrow \nu_{s,s}(f_{s+1,s+1} - m_{s,s})/(f_{s,s} - m_{s,s})$;

Return the web f_{ij};

5. Asymptotic nets with a constant angle

Now we turn to the third application of our approach: construction of surfaces with a constant angle between asymptotic curves.

We already know what to do whenever we face a hard Euclidean problem: consider its isotropic analog, that is, the surfaces with a constant *isotropic* angle γ between asymptotic curves. This is equivalent to the constant ratio $H^2/K = -\cot^2\gamma$. See [51].

It is striking that there is a simple approximate analytic expression for such *isotropic CRPC surfaces* when the angle γ is not too far from 90° (nothing like that is available in the Euclidean case).

The formula is especially elegant if one uses the complex coordinate $w = x + iy$ in the xy-plane. Using the complex derivatives $f_w := \frac{1}{2}\left(f_x - if_y\right)$ and $f_{\bar{w}} := \frac{1}{2}\left(f_x + if_y\right)$, we write the isotropic mean and Gaussian curvatures of a surface $z = f(w)$ as $H = 2f_{w\bar{w}}$ and $K = 4f_{w\bar{w}}^2 - 4f_{ww}f_{\bar{w}\bar{w}}$. The equation $H^2/K = -\cot^2\gamma =$ const takes the form (up to an overall sign)

$$f_{w\bar{w}} = \cos\gamma \cdot \sqrt{f_{ww}f_{\bar{w}\bar{w}}}. \tag{4}$$

Let us search for a solution in the form of series in the small parameter $\varepsilon = \cos\gamma$:

$$f(w) = f^{(0)}(w) + f^{(1)}(w)\varepsilon + f^{(2)}(w)\varepsilon^2/2 + \ldots, \tag{5}$$

for some unknown functions $f^{(0)}(w), f^{(1)}(w), \ldots$. Here $f^{(0)}(w)$ satisfies (4) for $\varepsilon = \cos\gamma = 0$, hence is a harmonic function. Any harmonic function (in a simply-connected domain) can be written as

$$f^{(0)}(w) = 2\mathrm{Re}\, g(w) \tag{6}$$

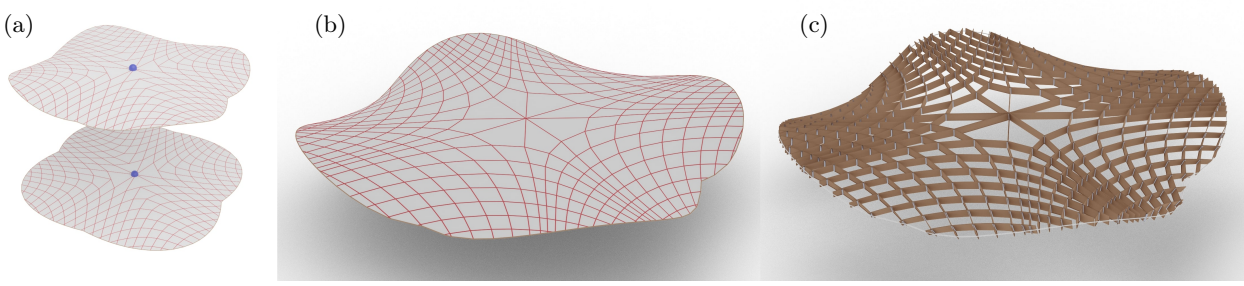


Figure 13: Design of a Euclidean CRPC surface with a prescribed flat point (blue dot) and a 60° angle between asymptotic curves, by optimizing a second-order approximation of the isotropic CRPC surface. The initial shape is constructed using Algorithm 3. It is then remeshed using the libigl [52] implementation of mixed-integer quadrangulation [53]. The remeshed surface, including its top view, is shown in (a). Optimization leads to a Euclidean CRPC surface (b) and a gridshell (c).

Algorithm 3: Design of an isotropic CRPC surface with given flat points (where more than two asymptotic curves meet)

Input: the isotropic angle γ between the asymptotic curves, the top views $w_1, \dots, w_n$ of the flat points

Output: A function $f^{\mathrm{appr}}(w)$ whose graph has approximately constant isotropic angle γ between the asymptotic curves and flat points near $w_1, \dots, w_n$

1 Compute $h(w) := \int (w - w_1) \dots (w - w_n)\, dw$ (expand and integrate term-wise) ;

2 Expand $(w - w_1)^2 \dots (w - w_n)^2$ and integrate term-wise twice to get a polynomial $g(w)$ such that $g''(w) = h'(w)^2$;

3 Set $f^{(0)}(w) := 2\mathrm{Re}\, g(w)$, $\quad f^{(1)}(z) := |h(w)|^2$, $\quad \varepsilon := \cos\gamma$, $\quad f^{(2)}(z) := 2\mathrm{Re}\, \left(h(w)^2\right) \log(|h'(w)| + \varepsilon)$;

4 Return $f^{\mathrm{appr}}(w) := f^{(0)}(w) + f^{(1)}(w)\varepsilon + f^{(2)}(w)\varepsilon^2/2$.

for some complex analytic function $g(w)$ (roughly, a polynomial or power series in w). For an analytic function, $g_{\bar{w}}(w) = 0$ and we denote $g_w(w) =: g'(w)$. The next function $f^{(1)}(w)$ satisfies the equation

$$f^{(1)}_{w\bar{w}} = \sqrt{f^{(0)}_{ww} f^{(0)}_{\bar{w}\bar{w}}}.$$

The right side equals $|g''(w)|$, and it is not hard to guess a particular solution

$$f^{(1)}(w) = |h(w)|^2, \qquad \text{where } h(w) := \int \sqrt{g''(w)}\, dw. \tag{7}$$

Technically, we need some assumptions to get a continuous branch of the square root here; e.g., the requirement that all zeroes of $g''(w)$, if any, have multiplicity 2. We present the details (unessential for implementation) in a subsequent publication [54, Section 5].

A direct computation shows that (5) satisfies (4) up to $O(\varepsilon^3)$, if

$$f^{(2)}(w) = 2\mathrm{Re}\left(h(w)^2\right) \log|h'(w)| \tag{8}$$

and (6)–(7) hold. Eqs. (6)–(8) are *not* the most general form of solutions, but they give much freedom for the design.

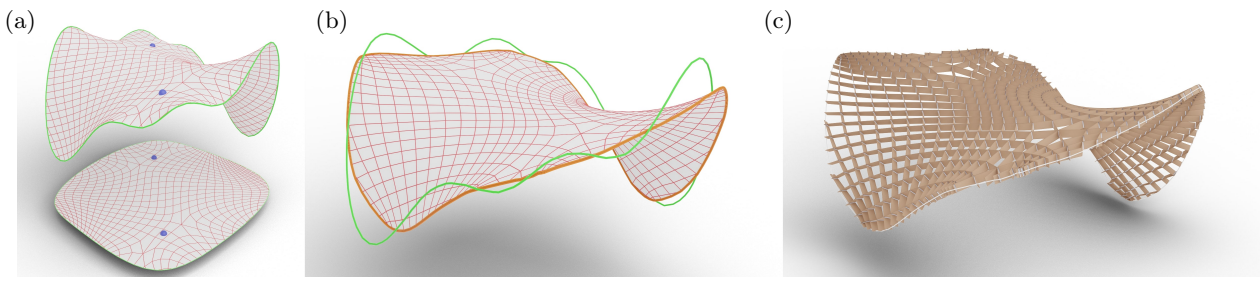


Figure 14: Design of an Euclidean CRPC surface with a given boundary (green), two flat points (blue), and an angle of 70° between asymptotic curves. The initial shape, constructed using Algorithm 4 and remeshed, is shown with its top view in (a). Each of the flat points (blue) has split into a pair of singular vertices due to limitations of the remeshing algorithm. Optimization leads to a Euclidean CRPC surface (b) and a gridshell (c).

Algorithm 4: Design of an approximate isotropic CRPC surface with a given boundary and flat points.

Input: The isotropic angle γ between the asymptotic curves, the top views $w_1, \dots, w_n$ of some of the flat points, a function $b(w)$ on the boundary of a Jordan domain Ω, the approximation order k

Output: A function $f^{\mathrm{appr}}(w)$ in Ω, whose graph has nearly constant isotropic angle γ between the asymptotic curves, flat points near $w_1, \dots, w_n$ (and maybe other), and boundary close to the graph of $b(w)$.

1 Introduce the function $f^{\mathrm{appr}}(h_0, \dots, h_k, g_0, g_1; w) := 2\mathrm{Re}\, g(w) + \varepsilon \cdot |h(w)|^2 + \varepsilon^2 \cdot \mathrm{Re}\, \left(h(w)^2\right) \log\left(|h'(w)| + \varepsilon\right)$, where $h(w) := \int (w - w_1) \cdots (w - w_n)(h_0 + h_1 w + \cdots + h_k w^k)\, dw$, $g(w) := \int \left(\int h'(w)^2\, dw\right) dw + g_0 + g_1 w$, and $\varepsilon := \cos\gamma$;

2 Find $h_0, \dots, h_k, g_0, g_1$ that minimize $\|f^{\mathrm{appr}}(h_0, \dots, h_k, g_0, g_1; w) - b(w)\|$ on $\partial\Omega$ by optimization;

3 Return the resulting function $f^{\mathrm{appr}}(h_0, \dots, h_k, g_0, g_1; w)$

For design, it is convenient to prescribe a polynomial $h'(w)$ first, then compute $h(w)$ and $g(w)$ by integration, and use (6)–(8) to get

$$f^{\mathrm{appr}}(w) = f^{(0)}(w) + f^{(1)}(w)\varepsilon + f^{(2)}(w)\varepsilon^2/2.$$

The roots of $h'(w)$ correspond to the roots of $g''(w)$, K, and H, hence to *flat points*, where more than two asymptotic lines can meet. At these points, the angle between the asymptotic curves can be different from γ, and we need to regularize $f^{(2)}(w)$, e.g., replacing $\log|h'(w)|$ with $\log(|h'(w)| + \varepsilon)$ in (8). Then $z = f^{\mathrm{appr}}(w)$ is the desired approximate isotropic CRPC surface.

For instance, Algorithms 3 and 4 construct an approximate isotropic CRPC surface with given flat points or/and boundary. Optimization leads to Euclidean CRPC surfaces.

We emphasize that we can control (some of) the flat points of the surface (see Figs. 13, 14, and 15(a)) and not just the boundary, as in the known design method for Weingarten surfaces by [55].

This section demonstrates one more advantage of isotropic geometry: it often comes with a powerful toolbox of complex analysis.

6. Optimization

We transform an isotropic shape into a Euclidean one by optimization.

For quad-mesh mechanisms, we apply the optimization algorithm from our previous work [33] as a black box, initializing it with isotropic mechanisms constructed in Sec. 3. The idea of the algorithm is to achieve easily-characterized *infinitesimal flexibility* and *sufficiently many isometric mechanism positions* (about ten); then a general theorem guarantees flexibility. We refer to a detailed description of the algorithm in [33] and focus on webs and CRPC surfaces in this section.

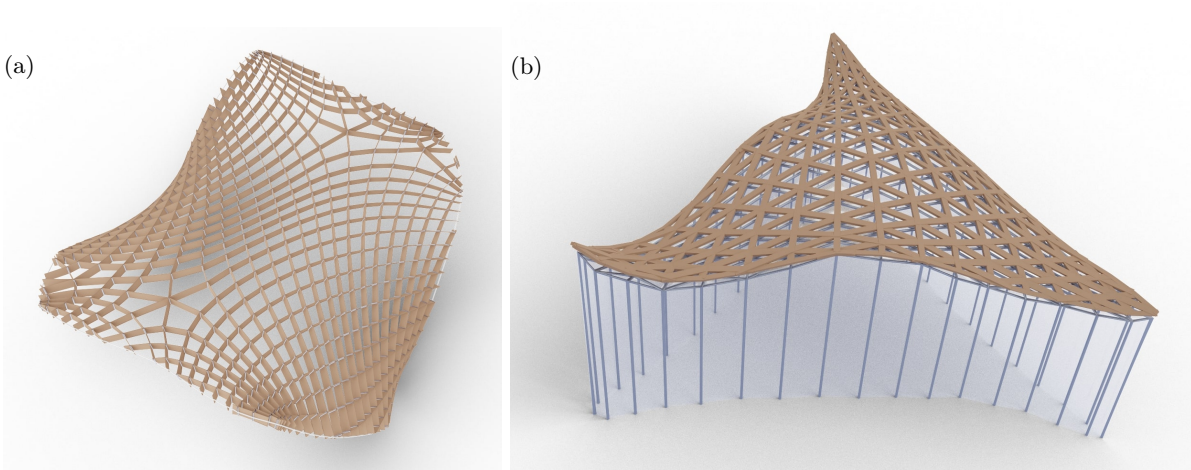


Figure 15: (a) The gridshell from Figure 14(c) from a different point of view. (b) Another gridshell example extracted from a Euclidean GGG web obtained by optimizing an isotropic one.

For webs and CRPC surfaces, we use the *guided projection algorithm* [56]. This algorithm is a regularized Gauss-Newton method, also known as Levenberg-Marquardt method. A key idea is to enhance performance by using at most quadratic constraints. To achieve that, auxiliary variables are introduced.

Also, we introduce a *special inner product* of two vectors $p = (p_1, p_2, p_3)$ and $q = (q_1, q_2, q_3)$ as follows:

$$\langle p, q \rangle_\varepsilon = p_1 q_1 + p_2 q_2 + \varepsilon p_3 q_3, \tag{9}$$

where $\varepsilon = 0$ represents the isotropic metric, and $\varepsilon = 1$ corresponds to the Euclidean metric. During the optimization process, ε is gradually increased from 0 to 1 to ensure a smooth transformation and improve the convergence behavior.

6.1. Constraints

We introduce the following auxiliary variables and constraints.

Discrete A-nets. Consider an A-net f_{ij}, so that $f_{ij}, f_{i-1,j}, f_{i,j-1}$, $f_{i+1,j}$, and $f_{i,j+1}$ are coplanar for each $0 < i, j < n$. This condition is encoded by the quadratic constraints

$$\langle \mathbf{n}_{ij}, f_{ij} - f_{i\pm1,j} \rangle = 0, \quad \langle \mathbf{n}_{ij}, f_{ij} - f_{i,j\pm1} \rangle = 0, \quad \|\mathbf{n}_{ij}\|^2 - 1 = 0, \tag{10}$$

where $\mathbf{n}_{ij}$ is an auxiliary variable representing the Euclidean normal to the net at f_{ij}, and $\langle a, b \rangle$ and $\|a\|$ are the Euclidean inner product and norm. Here the normalization of $\mathbf{n}_{ij}$ is essential to avoid convergence of $\mathbf{n}_{ij}$ to 0 during optimization.

The objective energy E_A is the sum of the squared left sides of (10) over all $0 < i, j < n$. If the $(i - j)$- and $(i + j)$-lines of f_{ij} form two A-nets, the energy is defined analogously.

Discrete geodesics. Consider a discrete curve C with the vertices $f_0, f_1, \ldots, f_k$ on a discrete surface. Recall that the curve C is a discrete geodesic if the surface normal is orthogonal to the curve binormal at each vertex $f_1, \cdots, f_{k-1}$. Orthogonality is expressed using the inner product, but instead of the Euclidean one, we employ the special inner product $\langle \cdot, \cdot \rangle_\varepsilon$. This makes the geodesic constraints quadratic:

$$\langle \mathbf{b}_i, f_i - f_{i\pm1} \rangle_\varepsilon = 0, \ \langle \mathbf{b}_i, \mathbf{n}_i \rangle_\varepsilon = 0, \ \|\mathbf{b}_i\|^2 - 1 = 0, \ \|\mathbf{n}_i\|^2 - 1 = 0, \tag{11}$$

where $\mathbf{b}_i$ is an auxiliary variable representing the binormal of C at f_i and $\mathbf{n}_i$ is an auxiliary variable representing the normal of the discrete surface at f_i (we take a single variable $\mathbf{n}_i$ per surface point, even if several curves C pass through the same point). The normalization of $\mathbf{b}_i$ and $\mathbf{n}_i$ follows the same reasoning as for $\mathbf{n}_{ij}$ above.

The energy $E_G^{C,\varepsilon}$ is the sum of the squared left sides of (11) over all $0 < i < k$.

Fairness. To enforce smoothness of C, we apply a soft fairness constraint. If f_i is not connected to the boundary, we require f_i to be close to the midpoint of $f_{i-1}f_{i+1}$:

$$2f_i - f_{i-1} - f_{i+1} \approx 0. \tag{12}$$

Otherwise, f_{i-1}, f_i, and f_{i+1} are required to be close to collinear:

$$\frac{f_i - f_{i-1}}{\|f_i - f_{i-1}\|} - \frac{f_{i+1} - f_i}{\|f_{i+1} - f_i\|} \approx 0. \tag{13}$$

The fairness energy E_f^C is the sum of the squared left sides of either (12) or (13) over all $0 < i < k$.

Closeness to the previous-iteration vertices. Consider selected vertices $f_0, f_1, \cdots f_s$ of f_{ij}. A self-closeness constraint on f_i ensures that each vertex f_i remains close to its previous position f_i^{pr} for $0 \le i \le s$:

$$\|f_i - f_i^{pr}\| \approx 0. \tag{14}$$

The vertex control energy E_{vcl} is the sum of the squared left side of (14) over all $0 \le i \le s$. The role of E_{vcl} is to keep the selected vertices close to their previous positions, stabilizing the iterative procedure and preventing large updates. We use the squared norm to keep the term at most quadratic, as required by our framework.

Closeness to a reference curve. If C approximates a fixed smooth curve, the vertices of C can move while preserving the overall shape of the curve. We enforce closeness by restricting each vertex f_i to move along directions close to the unit tangent $\mathbf{e}_1$ of the curve at its closest point f_i^{cl} on the curve:

$$\langle f_i - f_i^{cl}, \mathbf{e}_2 \rangle \approx 0, \quad \langle f_i - f_i^{cl}, \mathbf{e}_3 \rangle \approx 0, \tag{15}$$

where $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ form an orthonormal basis. The closeness energy E_{ccl}^C is the sum of the squared left side of (15) over all $0 \le i \le k$.

Closeness to the previous-iteration surface. For better convergence, the vertices should stay close to the previous iteration surface. This is enforced by requiring f_{ij} to move close to the tangent plane at its previous position, f_{ij}^{pr}:

$$\langle f_{ij} - f_{ij}^{pr}, \mathbf{n}_{ij}^{pr} \rangle \approx 0, \tag{16}$$

where $\mathbf{n}_{ij}^{pr}$ is the normal of the previous iteration surface at f_{ij}^{pr}. The closeness energy E_{scl} is the sum of the squared left side of (16) for $0 < i, j < n$.

Constant angle. To measure the intersection angle of parameter lines at f_{ij}, we use the angle between the central lines of the face $f_{ij} f_{i+1,j} f_{i+1,j+1} f_{i,j+1}$ [38]. Let f_a, f_b, f_c, f_d be the midpoints of its edges. The angle constraint, ensuring a constant angle γ between central lines, is

$$\left\langle \frac{f_c - f_a}{\|f_c^{pr} - f_a^{pr}\|}, \frac{f_d - f_b}{\|f_d^{pr} - f_b^{pr}\|} \right\rangle - \cos\gamma = 0. \tag{17}$$

Angle constraints (17) are quadratic as $\|f_c^{pr} - f_a^{pr}\|$ and $\|f_d^{pr} - f_b^{pr}\|$ are taken from the previous iteration. The angle energy E_{angle} is the sum of the squared left sides of (17) for all faces that do not contain flat points or boundary vertices.

6.2. Energies

GGG Webs. The objective E_{GGG}^{ε} for a GGG web for a given ε is the sum of energies $E_G^{C,\varepsilon}$ of all non-boundary i-, j-, and $(i-j)$-lines respectively.

AAG webs. The objective E_{AAG}^{ε} for an AAG web for a given ε is the sum of E_A and the energies $E_G^{C,\varepsilon}$ of all non-boundary $(i-j)$-lines.

AGAG webs. The objective for an AGAG web for a given ε is:

$$E_{AGAG}^{\varepsilon} = E_G^{i,\varepsilon} + E_G^{j,\varepsilon} + E_A^{even} + E_A^{odd}, \tag{18}$$

where $E_G^{i,\varepsilon}$ and $E_G^{j,\varepsilon}$ are the sums of energies $E_G^{C,\varepsilon}$ of all non-boundary i- and j-lines respectively, and E_A^{even} and E_A^{odd} are the energies E_A of the two A-nets formed by $(i-j)$- and $(i+j)$-lines.

CRPC Surfaces. The objective for a CRPC surface is

$$E_{CRPC} = E_A + E_{angle} + E_{ccl}^{\partial} + E_{vcl}^{\partial}. \tag{19}$$

If the surface boundary is prescribed, E_{ccl}^{∂} is the closeness energy for the boundary curve, and E_{vcl}^{∂} is the closeness vertices energy for the boundary vertices. Otherwise, these two terms are omitted.

Main Objective. The non-linear least-squares problem for a given ε aims to minimize

$$E^{\varepsilon} = E_{hard} + \omega_0 E_f + \omega_1 E_{scl} + \omega_2 E_{vcl}, \tag{20}$$

where $E_{hard} \in \{E_{GGG}^{\varepsilon}, E_{AGAG}^{\varepsilon}, E_{AAG}^{\varepsilon}, E_{CRPC}\}$ represents hard constraints. The energy E_f ensures smoothness, while E_{scl} and E_{vcl} prevent drastic shape changes during the optimization (here E_{vcl} is the sum over all vertices f_{ij}). The weights ω_0, ω_1, and ω_2 balance fairness and approximation.

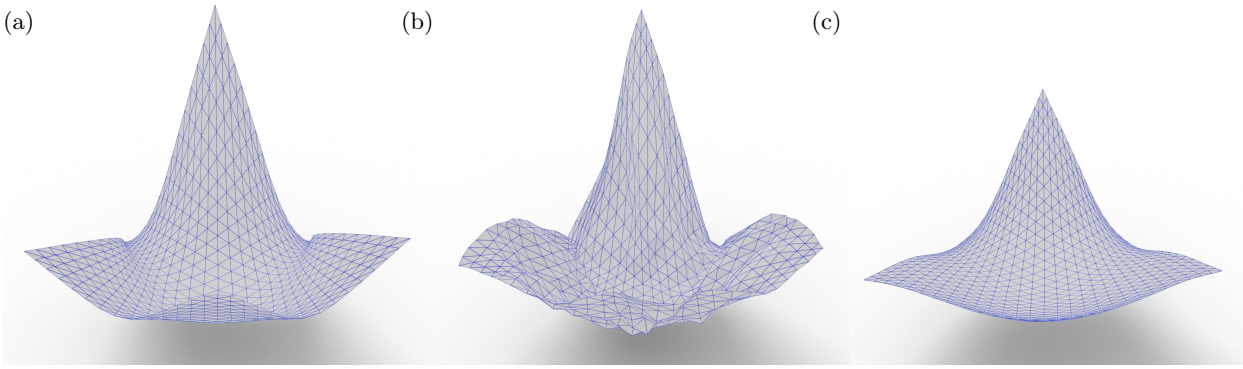


Figure 16: Advantage of using the gradual approach with metric (9). We compare Euclidean GGG webs obtained by optimizing the initial isotropic GGG web (Fig.9(e)) under different conditions. (a) The web optimized with the gradual approach and the set of weights from Table 1. (b) The web optimized without the gradual approach, using the same set of weights; the shape becomes non-smooth. (c) The web optimized without the gradual approach but with a higher fairness weight; here, the shape becomes significantly flatter.

6.3. Initialization

For webs and mechanisms, the shapes are initialized using their isotropic counterparts constructed in Sections 3 and 4. For CRPC surfaces, initialization is based on the second-order approximation of the isotropic CRPC surface constructed using either Algorithm 3 or 4. The resulting surface is then remeshed using the libigl [52] implementation of mixed-integer quadrangulation (MIQ) [53]. Now we provide initialization details for the auxiliary variables.

GGG webs. The binormals $\mathbf{b}_i$ and the normals $\mathbf{n}_i$ are initialized as the discrete binormals and $(0, 0, 1)$, respectively.

AGAG and AAG webs. Two sets of auxiliary normals are introduced: one for the A-net constraints, initialized using discrete Euclidean normals, and another for the geodesic constraints. The latter, along with the binormals, are initialized as in the GGG case. At $\varepsilon = 0$, all constraints are satisfied since the A-nets are the same in both geometries.

CRPC surfaces. The normals $\mathbf{n}_{ij}$ are initialized as the normalized cross product $(f_{i+1,j} - f_{i-1,j}) \times (f_{i,j+1} - f_{i,j-1})$.

6.4. Optimization Process

We minimize the energy E^{ε} given by (20) iteratively for ε increasing from 0 to 1, using the result of each step as the initialization for the next. At each step, we ensure that the hard constraints E_{hard} reach the accuracy of 10^{-5}. Table 1 summarizes the choices of weights, accuracy, and the number of variables for the final iteration with $\varepsilon = 1$. For the other ε, the weights are the same.

The usage of the gradual approach with metric (9) plays a critical role (see Fig. 16). For less constrained webs (i.e., GGG and AAG webs), it prevents drastic changes from the initial shapes. In the case of

Fig.	$\|V\|$	N_v	bbd	ω_1	ω_2	ω_3	**T/iter**	E_{hard}
1	1985	10912	3.99	5e-3	5e-3	5e-3	0.74 s	2.7e-12
2	361	4551	4.54	1e-2	1e-2	1e-2	0.11 s	1e-14
4	873	4381	4.45	5e-3	5e-3	5e-3	0.4 s	4.9e-12
9	631	8457	34.43, 292.98	1e-3	1e-3	1e-3	0.15 s	4.8e-20, 3.7e-20
10	1369	18807	4.50, 1.53	1e-2	1e-2	1e-2	0.36 s	2.7e-12, 3.1e-10
11	441	4572	2012.80	5e-3	5e-3	5e-3	0.13 s	3.9e-20
12	1681	18732	3239.26	5e-3	5e-3	5e-3	0.3 s	8.9e-20
13	1309	7084	6.71	5e-3	5e-3	5e-3	0.6 s	4.9e-12
14	757	3823	6.04	5e-3	5e-3	5e-3	0.38 s	4.9e-12
15(b)	154	1734	187.14	1e-3	1e-3	1e-3	0.08 s	1.1e-20

Table 1: Optimization statistics for selected figures. $|V|$ and bbd denote the number of vertices and the diagonal length of the axis-aligned bounding box of the discrete surface, respectively. N_v is the number of variables in the optimization process. the surface. ω_0, ω_1, and ω_2 are the weights at the first iteration for the fairness term and for approximation terms for each ε. T/iter represents the average time per iteration. E_{hard} is the energy of the hard constraints, namely GGG, AGAG, AAG, and CRPC constraints at the last iteration.

AGAG-webs its role is even more significant, as optimization consistently failed without this approach (see supplementary materials for details).

For quad-mesh mechanisms, the initial guesses via isotropic mechanisms work extremely well with the optimization algorithm of [33]; 10 iterations have been sufficient to satisfy the constraints with high accuracy without using the gradual approach based on metric (9). Supplementary materials include a video showing the isotropic and Euclidean flexion of Figs. 3, 5, and 8 (Euclidean only).

7. Discussion and Conclusion

Implementation. The optimization has been implemented in Python and tested on an Intel Xeon E5-2687W 3.0 GHz processor. Typically, around 10–20 iterations are required to reach the convergence for each value of ε. The weights ω_1, ω_2, and ω_3 are reduced by 10 times every five iterations, with the final reduction setting them to zero to ensure strict convergence. In Table 1, we observe that the time per iteration for CRCP surfaces is higher than for the other webs. This increase is due to angle constraint (17), which requires computing the length of each edge and which is not needed for the other cases.

Advantages. Our approach has advantages compared to known ones. The GGG webs appearing in architecture are mostly spherical patches. This matches the existence theorem for GGG webs on surfaces with constant Gaussian curvature by Mayrhofer [57]. In recent work [42], GGG and AAG webs are constructed by propagation. There, they take the first strip, consisting of two curves, and propagate the web by adding

curves one by one and re-optimizing the whole shape. The propagation algorithm can only produce regular grids without singularities. The shape control is biased: it is defined and controlled by the first strip. The areas far from it need extra manual control to create variation; otherwise, the surface eventually becomes flat except for the areas close to the initial strip. The present paper overcomes these limitations.

Schling et al. [40] discuss AAG and AGAG webs. They initialize the optimization of AAG webs with negatively curved rotational surfaces, thus limiting the variety of shapes. Their examples of AGAG webs are mostly flat and symmetric, due to the lack of initialization. Our initialization provided much more freedom.

Previous constructions of CRPC surfaces use optimization keeping the topology of the surface. We are not aware of a single approach known before that allows control of the positions of flat points.

The only known large-sized quad mesh mechanisms are Voss nets and T-nets. In the work [33], the authors discuss several ways to initialize and optimize general quad mesh mechanisms. The mechanisms were constructed there by subdivision of small-sized meshes or known pairs of isometric surfaces. The former strategy has limited constructional degrees of freedom. The latter requires remeshing, and the resulting initializations are often far from the true solutions, thereby exacerbating optimization difficulties and increasing the risk of convergence failure. The limitation in the variety of shapes again comes from initialization, which is resolved in the present work. Moreover, as observed in Sec. 3, initialization with a single isotropic mechanism yields an entire family of Euclidean ones, depending on several parameters.

Limitation. The main limitation of the suggested approach is the necessity to find and solve a non-degenerate analog of the original Euclidean problem in isotropic geometry. For the CRPC surfaces, our initialization is additionally limited to flat points where four asymptotic lines meet, and our remeshing step uses a cross-field-based algorithm, which limits the placement of flat points. In some cases, flat points may be split into multiple nearby ones to satisfy the quality requirements of the remeshing tool (see Fig. 14). Developing new techniques for more accurate control over flat points placement and remeshing remains an important direction for future work.

Shape change. For the CRPC surfaces with given flat points, GGG- and AAG-webs, and flexible nets, the visual change after optimization from isotropic to Euclidean geometry is minimal (see Figs. 3,9,11,12,14, and 15). In contrast, CRPC surfaces with prescribed flat points and boundary and AGAG-webs, the shape change after optimization can be drastic (see Figs. 10(g), 17, and 4). The latter cases are also the hardest to optimize due to being geometrically overconstrained.

Conclusion. In this work, we demonstrated the efficiency of isotropic geometry as a powerful framework for initializing and solving Euclidean geometric design problems by optimization. Our contributions include methods to construct approximate isotropic CRPC surfaces and discrete isotropic AAG webs, which serve as effective initializations for their Euclidean counterparts. We also studied discrete AGAG webs and quad mesh mechanisms with planar faces, where the recent classification of their isotropic versions are used as initial data for optimization.

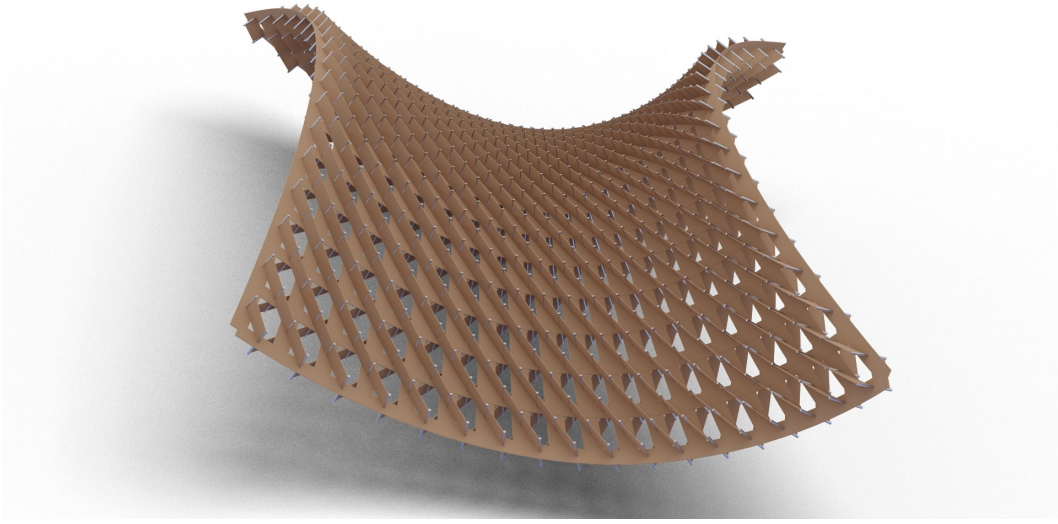


Figure 17: The gridshell from Figure 10(h) from a different point of view.

Future Work. A natural direction for future research is the question of whether one can smoothly transform a minimal surface (isotropic or Euclidean) with a given boundary into a CRPC surface (isotropic or Euclidean) with the same boundary. In a forthcoming publication [54], we prove that such a transformation is possible in the absence of flat points. The case of surfaces with flat points remains open.

Another direction is to explore webs composed of circular flat lamellas instead of straight ones. The case where the lamellas are orthogonal to the reference surface and follow its parameter lines (known as *S-nets*) has been studied in [58]. S-nets are characterized by both families of parameter lines having constant normal curvature. An interesting direction is to consider circular lamellas that are tangent to the reference surface. These lamellas follow curves of constant geodesic curvature. In the isotropic setting, the top view of such a web consists of circular arcs, forming a planar web composed of such arcs. While the planar case of circular webs remains an open problem, several classification results are available (see [59, 60, 61]).

Another direction for further applications of the proposed method is the structural design with non-vertical loads. In particular, one may extend the analysis by [62] of forces that can be supported by a given 2D structure to the 3D case.

Acknowledgement

This work was supported in part by the Beijing Natural Science Foundation (Z240002), the Key Program of the National Natural Science Foundation of China (12494550, 12494553), the National Natural Science Foundation of China (62495092, 62125305), the NSFC–FDCT Joint Project (62461160259), and KAUST Baseline Funding.

References

[1] H. Sachs, Isotrope Geometrie des Raumes, Vieweg, 1990.

[2] C. Müntz, Zum Randwertproblem der partiellen Differentialgleichung der Minimalflächen, J. Reine Angew. Math. 139 (1911) 52–79.

[3] C. Müller, H. Pottmann, Isometric surfaces in isotropic 3-space, SIAM J. Applied Algebra Geom. 10 (2026) 303–349.

[4] K. Strubecker, Differentialgeometrie des isotropen Raumes I: Theorie der Raumkurven, Sitzungsber. Akad. Wiss. Wien 150 (1941) 1–53.

[5] K. Strubecker, Differentialgeometrie des isotropen Raumes II: Die Flächen konstanter Relativkrümmung $K = rt - s^2$, Math. Zeitschrift 47 (1942) 743–777.

[6] K. Strubecker, Differentialgeometrie des isotropen Raumes III: Flächentheorie, Math. Zeitschrift 48 (1942) 369–427.

[7] H. Pottmann, K. Opitz, Curvature analysis and visualization for functions defined on euclidean spaces or surfaces, Computer aided geometric design 11 (6) (1994) 655–674.

[8] H. Pottmann, Y. Liu, Discrete surfaces in isotropic geometry, in: M. Sabin, J. Winkler (Eds.), Mathematics of Surfaces XII, Vol. 4647 of LNCS, Springer, 2007, pp. 431–363.

[9] H. Pottmann, P. Grohs, N. J. Mitra, Laguerre minimal surfaces, isotropic geometry and linear elasticity, Adv. Comp. Math 31 (2009) 391–419.

[10] V. C. Inza, P. Fykouras, F. Rist, D. Haeseker, M. Hojjat, C. Müller, H. Pottmann, Designing triangle meshes with controlled roughness, ACM Trans. Graphics 43 (6) (2024) 166:1–166:20, proc. SIGGRAPH Asia.

[11] K. Strubecker, Airysche Spannungsfunktion und isotrope Differentialgeometrie, Mathematische Zeitschrift 78 (1) (1962) 434–459.

[12] Y.-C. Chiang, Design and fabrication of shell structures: Aided by radial basis functions and reconfigurable mechanisms, Ph.D. thesis, Delft University of Technology (2022).

[13] Y.-C. Chiang, Maxwell-Rankine stress functions of membrane shells and their relation to that of planar funicular gridshells, International Journal of Solids and Structures 252 (2022) 111768.

[14] M. Miki, T. Igarashi, P. Block, Parametric self-supporting surfaces via direct computation of Airy stress functions, ACM Transactions on Graphics 34 (4) (2015).

[15] C. Millar, T. Mitchell, A. Mazurek, A. Chhabra, A. Beghini, J. N. Clelland, A. McRobie, W. F. Baker, On designing plane-faced funicular gridshells, International Journal of Space Structures 38 (1) (2022) 40–63.

[16] X. Tellier, C. Douthe, L. Hauswirth, O. Baverel, Form-finding with isotropic linear Weingarten surfaces, in: Advances in Architectural Geometry 2020, Champs-sur-Marne, 2021, pp. 18–37.

[17] E. Vouga, M. Höbinger, J. Wallner, H. Pottmann, Design of self-supporting surfaces, ACM Trans. Graphics 31 (2012) #87,1–11, proc. SIGGRAPH.

[18] K. Strubecker, Differentialgeometrie des isotropen Raumes IV: Theorie der flächentreuen Abbildungen der Ebene, Math. Zeitschrift 50 (1944) 1–92.

[19] K. Strubecker, Differentialgeometrie des isotropen Raumes V. Zur Theorie der Eilinien, Mathematische Zeitschrift 51 (5) (1949) 1432–1823.

[20] J. J. Koenderink, A. J. van Doorn, Image processing done right, in: A. Heyden, G. Sparr, M. Nielsen, P. Johansen (Eds.), Computer Vision — ECCV 2002, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 158–172.

[21] M. Kilian, D. Pellis, J. Wallner, H. Pottmann, Material-minimizing forms and structures, ACM Trans. Graphics 36 (6) (2017) 173:1–173:12, proc. SIGGRAPH Asia.

[22] A. G. M. Michell, The limit of economy of material in frame-structures, Phil. Mag., Ser. VI 8 (1904) 589–597.

[23] D. Pellis, M. Kilian, F. Dellinger, J. Wallner, H. Pottmann, Visual smoothness of polyhedral surfaces, ACM Trans. Graphics 38 (4) (2019) 260:1–260:11, proc. SIGGRAPH.

[24] D. Pellis, Quad meshes as optimized architectural freeform structures, Ph.D. thesis, Wien (2019).

[25] M. Peternell, H. Pottmann, A laguerre geometric approach to rational offsets, Comput. Aided Geom. Des. 15 (3) (1998) 223–249.

[26] M. Skopenkov, H. Pottmann, P. Grohs, Ruled Laguerre minimal surfaces, Math. Zeitschrift 272 (2012) 645–674.

[27] M. Skopenkov, P. Bo, M. Bartoň, H. Pottmann, Characterizing envelopes of moving rotational cones and applications in CNC machining, Comp. Aided Geometric Design 83 (2020) 101944.

[28] I. Izmestiev, Classification of flexible Kokotsakis polyhedra with quadrangular base, Int. Math. Res. Not. 2017 (3) (2017) 715–808.

[29] A. Nurmatov, M. Skopenkov, F. Rist, J. Klein, D. Michels, Quasi-symmetric nets: A constructive approach to the equimodular elliptic type of Kokotsakis polyhedra, Computer-Aided Design 199 (2026) 104102. doi:https://doi.org/10.1016/j.cad.2026.104102.

[30] R. Sauer, Differenzengeometrie, Springer, 1970.

[31] Z. He, S. D. Guest, On rigid origami II: quadrilateral creased papers, Proceedings of the Royal Society A 476 (2237) (2020) 20200020.

[32] G. Nawratil, From axial C-hedra to general P-nets, in: J. Lenarčič, M. Husty (Eds.), Advances in Robot Kinematics 2024, Springer Nature Switzerland, Cham, 2024, pp. 340–347.

[33] C. Jiang, D. Lyakhov, F. Rist, H. Pottmann, J. Wallner, Quad mesh mechanisms, ACM Trans. Graph. 43 (6) (2024) 243:1–243:17, proc. SIGGRAPH Asia.

[34] E. Schling, Repetitive structures, Ph.D. thesis, Chair of Structural Design, Technical University of Munich (2018).

[35] E. Schling, Z. Wan, A geometry-based design approach and structural behaviour for an asymptotic curtain wall system, Journal of Building Engineering (2022) 104432.

[36] J. Schikore, E. Schling, T. Oberbichler, A. M. Bauer, Kinetics and design of semi-compliant grid mechanisms, Adv. Archit. Geom (2021) 108–129.

[37] Y. Liu, O. Pirahmad, H. Wang, D. Michels, H. Pottmann, Helical surfaces with a constant ratio of principal curvatures, Contributions to Algebra and Geometry 64 (2023) 1087–1105.

[38] H. Wang, H. Pottmann, Characteristic parameterizations of surfaces with a constant ratio of principal curvatures, Computer Aided Geometric Design 93 (2022) 102074.

[39] C. Jiang, C. Wang, E. Schling, H. Pottmann, Computational design and optimization of quad meshes based on diagonal meshes, in: Advances in Architectural Geometry 2020, Presses des Ponts, Champs-sur-Marne, France, 2021, pp. 38–60.

[40] E. Schling, H. Wang, S. Hoyer, H. Pottmann, Designing asymptotic geodesic hybrid gridshells, Computer Aided Design 152 (2022) 103378.

[41] B. Wang, H. Wang, E. Schling, H. Pottmann, Rectifying strip patterns, ACM TOG 42 (6) (Dec. 2023).

[42] B. Wang, M. Almaskin, H. Pottmann, Computational design of asymptotic geodesic hybrid gridshells via propagation algorithms, Computer Aided Design 178 (2025) 103800.

[43] C. Müller, H. Pottmann, The geometry of discrete asymptotic-geodesic 4-webs in isotropic 3-space, Monatsh. Math. 203 (2024) 223–246.

[44] O. Pirahmad, H. Pottmann, M. Skopenkov, Flexible polyhedral nets in isotropic geometry, Beitr. Algebra Geom. (2025). doi:https://doi.org/10.1007/s13366-025-00820-y.

[45] C. Jiang, C. Wang, X. Tellier, J. Wallner, H. Pottmann, Planar panels and planar supporting beams in architectural structures, ACM Trans. Graphics 41 (2) (2022) 1–17.

[46] H. Graf, R. Sauer, Über dreifache Geradensysteme, Sitz. Bayer. Akad. Math.-nat. Abt. (1924) 119–156.

[47] F. Petrov, S. Tikhomirov, About angles and distances, in: St. Petersburg high-school mathematical Olympiads 2000-2002, Nauka (RAN), St. Petersburg, 2006, pp. 330–348.

[48] A. Bobenko, Yu. Suris, Discrete differential geometry: Integrable Structure, no. 98 in Graduate Studies in Math., American Math. Soc., Berlin, 2008.

[49] O. Pirahmad, H. Pottmann, M. Skopenkov, Area preserving Combescure transformations, Results Math 80 (27) (2025).

[50] I. Izmestiev, A. Rasoulzadeh, J. Tervooren, Isometric deformations of discrete and smooth T-surfaces, Computational Geometry 122 (2024) 102104.

[51] K. Yorov, M. Skopenkov, H. Pottmann, Surfaces of constant principal-curvatures ratio in isotropic geometry, Contrib. Algebra Geom. 66 (2025) 775–814. doi:10.1007/s13366-024-00768-5.

[52] A. Jacobson, D. Panozzo, et al., libigl: A simple C++ geometry processing library, https://libigl.github.io/ (2018).

[53] D. Bommes, H. Zimmer, L. Kobbelt, Mixed-integer quadrangulation, ACM Trans. Graph. 28 (3) (Jul. 2009).

[54] M. Skopenkov, K. Yorov, Families of surfaces with constant ratio of principal curvatures and Plateau's problem (2025). arXiv:2510.14527.

[55] D. Pellis, M. Kilian, H. Pottmann, M. Pauly, Computational design of weingarten surfaces, ACM Trans. Graph. 40 (4) (Jul. 2021).

[56] C. Tang, X. Sun, A. Gomes, J. Wallner, H. Pottmann, Form-finding with polyhedral meshes made simple, ACM Trans. Graph. 33 (4) (Jul. 2014).

[57] K. Mayrhofer, Sechseckgewebe aus Geodätischen, Monatshefte f. Mathematik u. Physik 38 (1931) 401–404.

[58] D. Pellis, H. Wang, M. Kilian, F. Rist, H. Pottmann, C. Müller, Principal symmetric meshes, ACM Transactions on Graphics 39 (4) (2020) 127–1.

[59] V. B. Lazareva, Three-webs formed by families of circles on the plane, in: Differential geometry (Russian), Kalinin. Gos. Univ., Kalinin, Kalinin, 1977, pp. 49–64, 140.

[60] H. Pottmann, L. Shi, M. Skopenkov, Darboux cyclides and webs from circles, Computer Aided Geometric Design 29 (1) (2012) 77–97.

[61] F. K. Nilov, On new constructions in the Blaschke-Bol problem, SBORNIK MATHEMATICS 205 (11) (2014) 1650–1667.

[62] A. Amendola, A. Fortunato, F. Fraternali, O. Mattei, G. W. Milton, P. Seppecher, Limit analysis of strut nets, Mathematics and Mechanics of Solids 28 (8) (2023) 1760–1775.