# Computational design of asymptotic geodesic hybrid gridshells via propagation algorithms

Bolun Wang[a,c,*], Maryam Almaskin[c], Helmut Pottmann[b,c]

[a] *RWTH Aachen University, Visual Computing Institute, Germany*
[b] *TU Wien, Austria*
[c] *King Abdullah University of Science and Technology, Visual Computing Center, Saudi Arabia*

**Abstract**

Complex architectural structures may be built in a simple and cost-effective way if their geometry respects the fabrication constraints. Examples of such structures are provided by gridshells that are built from straight and flat slats which are bent on site so that they become tangential or normal to the design surface. Tangential slats follow geodesic curves on the surface, while normal slats are attached along asymptotic curves. Extending work by Frei Otto, Julius Natterer and others, who placed the slats tangentially, Eike Schling proposed structures which also contain slats normal to the reference surface. In the present paper we address those gridshells that consist of three families of bent elements, either tangential or normal to the design surface, and are arranged in a triangular web. We propose algorithms for the computational design of such webs that start from a boundary strip and propagate it, partially under additional guidance, to an entire web.

*Keywords:* mesh optimization, asymptotic curve, geodesic curve, triangular web, construction-aware design, elastic gridshell

## 1. Introduction

The present research has been motivated by certain types of gridshells which Eike Schling and his collaborators have studied and built in recent years [1, 2, 3, 4, 5]; for examples see Fig. 1. Originally straight flat metal lamellas or wooden lats are bent and assembled towards the final structure. The lamellas are either orthogonal to an underlying design surface $S$ or tangential to it. In the former case, they follow asymptotic curves on $S$, in the latter they are aligned with geodesics (shortest paths) of $S$. In the gridshell of Fig. 1, left, the steel lamellas are orthogonal to the reference surface and thus follow asymptotic curves. Due to the right node angle, the underlying surface is a minimal surface. The right image of Fig. 1 shows a gridshell from three families of wooden lats. Two families are orthogonal to the reference surface, the third is tangential to it.

Following up on a recent contribution by Schling et al. [6], we are interested in arrangements of lamellas in form of so-called *triangular webs* (see e.g. Fig. 3). They consist of three discrete families of curves, which may be seen as selected curves from the iso-parameter lines $u = const.$, $v = const.$ and the curves $u + v = const.$ of a parametric surface $\mathbf{x}(u,v)$. For our computational task we can assume an arrangement where three curves pass through each node, while in practice one may shift one family and thus have only crossings of two lamellas (see Fig. 1. right). Depending on whether the curves of a family are asymptotic curves (A) or geodesics (G), we have three types of webs which we call GGG, AGG, and AAG. Clearly, the presence of asymptotic curves requires reference surfaces $S$ of negative Gaussian curvature.

Since there is a two-parametric family of geodesics on a surface, but only two one-parametric families of asymptotic curves (assuming negative Gaussian curvature), the most flexible webs are GGG webs and the most constrained are

*Corresponding author
*Email addresses:* bolun.wang@kaust.edu.sa (Bolun Wang), maryam.almaskin@kaust.edu.sa (Maryam Almaskin), helmut.pottmann@kaust.edu.sa (Helmut Pottmann)

Figure 1: Two gridshells by Eike Schling. Left: Asymptotic pavilion at the campus of TU Munich. Right: Asymptotic geodesic hybrid timber gridshell (images courtesy Eike Schling, Felix Noe and Yilinke Tan).

AAG webs. All three cases imply shape restrictions, which are most severe for AAG webs. This makes surface approximation with such webs a hard task. It has been addressed very recently with a level set method, even for cases where lamellas are not necessarily tangential or orthogonal to a reference surface [7]. However, there is no guarantee that approximation will be possible within a meaningful tolerance or that the computed webs are practically useful. Hence, there is a need for alternative design methods, which is the topic of our paper.

### *1.1. Contributions*

Our contribution is on the geometric and algorithmic side and addresses the problem of computational design of geodesic webs (GGG) or hybrid asymptotic-geodesic webs (AGG, AAG). For a careful discussion of the underlying geometry and all steps of the computational pipeline, we refer to [6]. We focus here on discrete webs. These are nothing else than triangle meshes of regular combinatorics where the three families of major mesh polylines are discrete geodesics or asymptotic curves. Unlike [6], we express these properties only with discrete osculating planes and surface normals and avoid discretizations based on angles between mesh edges. While angle-based discretizations are correct and in the case of GGG webs already appeared in the work of R. Sauer [8], we will see that they are less effective than our discretizations.

Sauer's construction of discrete GGG webs [8] is a *propagation algorithm* that starts with a boundary strip and step-by-step adds further rows of vertices to generate the entire web; see Fig. 3. We present an alternative version of propagation based on global optimization that enables slight changes of the input strip and already computed parts. Moreover, it allows us to steer the shape generation process in various ways. Our main contributions are similar algorithms for hybrid webs of type AAG and AGG.

### *1.2. Prior work*

In view of recent contributions on the present topic, we refer to [6] and [7] for an extensive overview of prior and related work. We concentrate here only on closely related work from those areas to which we contribute, namely discretization and algorithms for computational design.

The core geometric topic of the present paper are webs of geodesics and asymptotic curves. The geometry of webs in general is treated in detail in the monograph by Blaschke and Bol [9]. GGG webs in the plane are webs of straight lines, which according to a seminal paper by Graf and Sauer [10] are formed by the tangents of an algebraic curve of class 3. GGG webs on surfaces have been studied by Sauer [8], with the already addressed discrete model and further explicit nontrivial examples on rotational and spiral surfaces. The partial differential equation which characterizes GGG webs is found in [11], but there is little hope to obtain a solution which provides useful insight for design. GGG webs exist on all surfaces of constant Gaussian curvature, since those possess mappings to the plane in which geodesics get mapped to straight lines [10, 12]. While 4-webs of type AGAG have been numerically computed in [6], their exact existence in Euclidean space is still unclear. However, AGAG webs exist in so-called isotropic

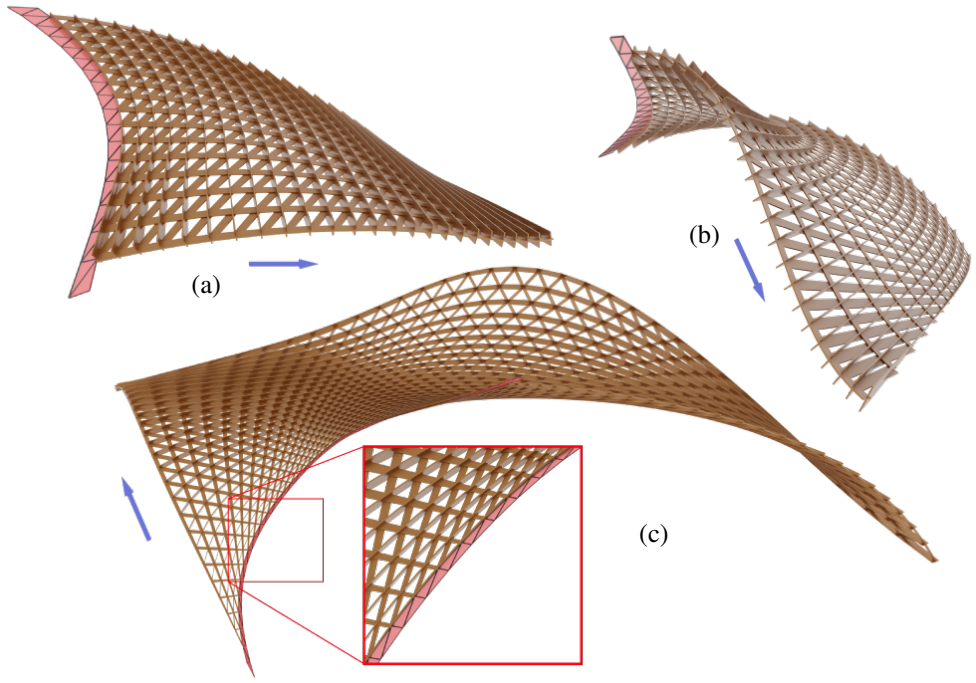



Figure 2: AGG gridshells generated by our propagation algorithm. The gridshells are extracted using the inner vertices of the corresponding webs (refer to Fig. 3).The blue arrows indicate the propagation directions. The red strips are the first strips of the final AGG webs.

geometry, a simplified version of Euclidean geometry, and possess remarkable properties and relations to various topics in differential geometry [13].

Discretizations of asymptotic and geodesic curves on surfaces appear in numerous contributions to discrete differential geometry, including the books by Sauer [14] and Bobenko and Suris [15], and in papers related to so-called Voss nets (conjugate nets of geodesics) [16, 17], discrete developable surfaces in form of orthogonal geodesic nets [18, 19] and asymptotic nets with a constant node angle [20].

Computational design of webs has been addressed in connection with architectural applications [21, 22] and in design tools for weaving patterns [23, 24, 25], including those where the flat states are curved [26, 27].

## 2. Discrete geometric models and basic optimization algorithm

In a 3-web, the three families of curves $U$, $V$, and $W$ are sampled from the iso-$u$, iso-$v$, and iso-$w = u + v$ curves of a parametric surface $\mathbf{x}(u,v)$. To construct a 3-web $S$, e.g., a GGG web, we take a strip that consists of two neighboring polylines $V_0$ and $V_1$ of the same family $V$ as input (Fig. 3 (a)) and iteratively expand the patches into larger ones using our propagation method. Assuming that each $V$ curve has $n$ vertices $\mathbf{p}_i$, and there are $k \geq 2$ curves $V_0, \ldots, V_{k-1}$ in $S$, the propagation is in the direction from $V_{k-2}$ to $V_{k-1}$, i.e. $n$ new vertices are added as a new boundary curve $V_k$ of $S$, and each of the $n$ $U$ curves and $n-1$ $W$ curves are extended by 1 vertex, see Fig. 4. The propagation generates the next row of vertices so that the webs approximately satisfy the constraints for GGG, AAG, or AGG. After propagating one or a few rows, we optimize the whole structure for a smooth, accurate result. We iterate the two steps until the web grows to the size that fulfills the design purposes (Fig. 3 (b), (c)). The propagation algorithms for the three types of web structures are introduced in Sec. 3. We now explain our discrete models and the corresponding global optimization algorithms for the three types.

*Discrete curves.* A polyline $C$ that contains $n$ vertices $\mathbf{p}_0, \ldots, \mathbf{p}_{n-1}$ is a discrete version of a smooth curve. Its edges are discrete tangents and three consecutive vertices $\mathbf{p}_{i-1}\mathbf{p}_i\mathbf{p}_{i+1}$ span the discrete osculating plane at vertex $\mathbf{p}_i$. A discrete

Frenet frame $(\mathbf{t}_i, \mathbf{n}_i^p, \mathbf{b}_i)$ at point $\mathbf{p}_i$ can be defined as $\mathbf{t}_i = \frac{\mathbf{p}_{i+1}-\mathbf{p}_{i-1}}{\|\mathbf{p}_{i+1}-\mathbf{p}_{i-1}\|}$, $\mathbf{n}_i^p = \frac{2\mathbf{p}_i-\mathbf{p}_{i+1}-\mathbf{p}_{i-1}}{\|2\mathbf{p}_i-\mathbf{p}_{i+1}-\mathbf{p}_{i-1}\|}$, $\mathbf{b}_i = \frac{(\mathbf{p}_{i+1}-\mathbf{p}_i)\times(\mathbf{p}_i-\mathbf{p}_{i-1})}{\|(\mathbf{p}_{i+1}-\mathbf{p}_i)\times(\mathbf{p}_i-\mathbf{p}_{i-1})\|}$, where $\mathbf{t}_i, \mathbf{n}_i^p, \mathbf{b}_i$ are the tangent vector, the principal normal vector, and the binormal vector, respectively. If edges have the same length, tangent and principal normal are outer and inner bisector of the edges through $\mathbf{p}_i$. Otherwise, we may compute them as bisectors via unit edge vectors $\mathbf{e}_i = \frac{\mathbf{p}_{i+1}-\mathbf{p}_i}{\|\mathbf{p}_{i+1}-\mathbf{p}_i\|}$ as $\mathbf{t}_i = \frac{\mathbf{e}_i+\mathbf{e}_{i-1}}{\|\mathbf{e}_i+\mathbf{e}_{i-1}\|}$ and $\mathbf{n}_i^p = \frac{\mathbf{e}_i-\mathbf{e}_{i-1}}{\|\mathbf{e}_i-\mathbf{e}_{i-1}\|}$.

*Discrete asymptotic curves.* Along an asymptotic curve on a surface, osculating planes of the curve are tangent planes of the surface. In other words, the surface normal vectors are orthogonal to the osculating planes. Thus, the constraints on each inner vertex $(i = 1, \ldots, n-2)$ can be expressed as

$$\begin{aligned} (\mathbf{n}_i, \mathbf{p}_i - \mathbf{p}_{i+1}) &= 0, \\ (\mathbf{n}_i, \mathbf{p}_{i-1} - \mathbf{p}_i) &= 0, \end{aligned} \tag{1}$$

where $\mathbf{n}_i$ is the surface normal vector at $\mathbf{p}_i$. Here and in the following, $(\mathbf{a}, \mathbf{b})$ denotes the scalar product of vectors $\mathbf{a}, \mathbf{b}$. Thus, the objective function for the asymptotic curves is

$$E_a = \sum_i ((\mathbf{n}_i, \mathbf{p}_i - \mathbf{p}_{i+1})^2 + (\mathbf{n}_i, \mathbf{p}_{i-1} - \mathbf{p}_i)^2). \tag{2}$$

The above objective function takes the surface normal $\mathbf{n}_i$ at each inner vertex $\mathbf{p}_i$ as an auxiliary variable. $\mathbf{n}_i$ is defined as a unit vector orthogonal to the tangent vectors of two of the three curves passing through $\mathbf{p}_i$, as shown in Fig. 4. We use the following objective function to obtain the surface normal vectors during optimization,

$$E_n = \sum_i ((\|\mathbf{n}_i\|^2 - 1)^2 + (\mathbf{n}_i, \mathbf{p}_{i+1} - \mathbf{p}_{i-1})^2 + (\mathbf{n}_i, \mathbf{p}_{i+n} - \mathbf{p}_{i-n})^2). \tag{3}$$

*Discrete geodesic curves.* For geodesic curves, the surface normals should be contained in the discrete osculating planes, i.e., orthogonal to the binormal vectors of the curves:

$$E_g = \sum_i (\mathbf{n}_i, \mathbf{b}_i)^2. \tag{4}$$

$\mathbf{n}_i$ and $\mathbf{b}_i$ are used as auxiliary variables besides $\mathbf{p}_i$. The binormal vectors are obtained through

$$E_b = \sum_i ((\|\mathbf{b}_i\|^2 - 1)^2 + (\mathbf{b}_i, \mathbf{p}_{i+1} - \mathbf{p}_i)^2 + (\mathbf{b}_i, \mathbf{p}_i - \mathbf{p}_{i-1})^2). \tag{5}$$

*Discrete webs via optimization.* The objective function for a GGG web can then be expressed as

$$E_{ggg} = E_g^u + E_g^v + E_g^w + E_b^u + E_b^v + E_b^w + E_n, \tag{6}$$

where the superscripts indicate that the constraints are applied to the $u = const.$, $v = const.$, or $w = u + v = const.$ family of curves on $\mathbf{x}(u, v)$. For AAG webs (see Fig. 10), we select the iso-$v$ curves as geodesic curves and iso-$u$ and iso-$w$ curves as asymptotic curves. For AGG (see Fig. 7), the iso-$v$ curves are the asymptotic curves, while the other two families of curves are geodesics. Thus the AAG and AGG objective functions are

$$\begin{aligned} E_{aag} &= E_a^u + E_g^v + E_a^w + E_b^v + E_n, \\ E_{agg} &= E_g^u + E_a^v + E_g^w + E_b^u + E_b^w + E_n. \end{aligned} \tag{7}$$

To obtain high-quality web structures, we optimize $E_{ggg}$, $E_{aag}$, or $E_{agg}$ along with constraints for curve fairness and approximation to the first strip. For each curve, the fairness term places every inner vertex $\mathbf{p}_i$ of the curve to the midpoint of its two neighboring vertices,

$$E_f = \sum_i (2\mathbf{p}_i - \mathbf{p}_{i-1} - \mathbf{p}_{i+1})^2. \tag{8}$$

The fairness for the 3-web structures can be expressed as:

$$E_{fair} = E_f^u + E_f^v + E_f^w. \tag{9}$$

To approximate the first strip consisting of $V_0$ and $V_1$, we use the term

$$E_{appro} = \sum_i (\mathbf{p}_i - \mathbf{p}_i^{ori})^2, \tag{10}$$

where $i \in [0, 2n-1]$. The $\mathbf{p}_i^{ori}$ is the original position of the $i$-th vertex of the first strip.

The overall objective function $E$ for our web structure optimization can be expressed as

$$E = E_c + \lambda_{fair} E_{fair} + \lambda_{appro} E_{appro}, \tag{11}$$

where $E_c$ is either $E_{ggg}$, $E_{aag}$ or $E_{agg}$, while $\lambda_{fair}$ and $\lambda_{appro}$ are the weights for fairness and approximation. We employ a Levenberg-Marquardt method as in [28] to minimize $E$, such that the term $E_c$ reaches a desired accuracy, e.g. $10^{-5}$. The choices for the weights of our results are listed in Table 1.

Sauer [8] proposed to use the angle constraints in Eq. 12 for GGG web propagation. Similar constraints have been successfully applied to the global optimization of GG nets and AGG webs in [6]. However, we discovered that the angle constraints for GGG webs are too restrictive for global optimization, possibly leading to optimization failures, as shown in Fig.12 (a). Thus, we adopt the constraints based on discrete osculating planes Eq. 6, which have also been used in [7].

## 3. Propagation algorithms

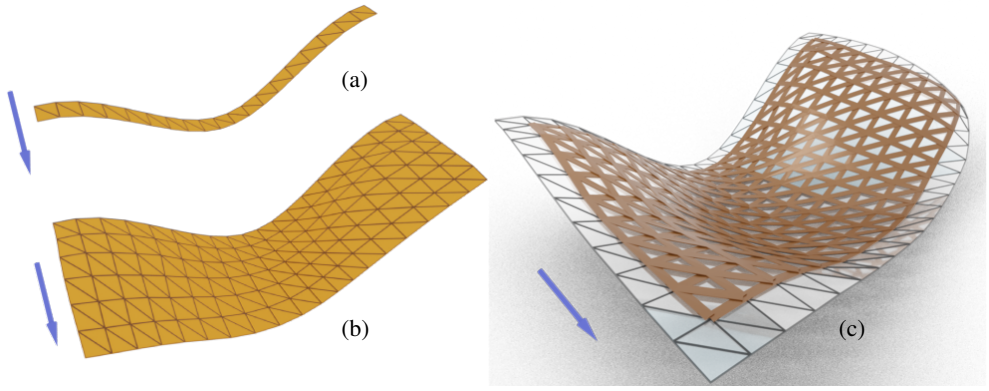



Figure 3: An example of GGG propagation. Starting with an initial strip (a), we iteratively propagate the boundaries and optimize for accurate GGG webs (b). The blue arrows indicate the propagation directions. In (c), we extract the gridshell structure (brown) using the inner vertices of the GGG web (light gray), since the AAG, AGG, or GGG constraints (Eq. 11) are not applied on boundary vertices during global optimization.

In this section, we introduce our propagation algorithms for GGG, AAG, and AGG webs. As mentioned above, our webs consist of 3 families of curves $U$, $V$, and $W$, and the propagation always adds further $V$ curves. The purpose of propagation is to generate approximate GGG, AAG, or AGG webs before global optimization. We will demonstrate the effectiveness of proper propagation in Section 4. It is not enough to propagate by adding a kind of offset to the current boundary and hoping that subsequent global optimization will repair an initial choice that did not obey the constraints at all. We will show that this increases the risk of divergence or requires more global optimization steps along with careful and in practice undesirable tuning of parameters.

### *3.1. GGG webs*

Before going into detail we mention that GGG propagation adds a polyline $V_k$ and makes sure that the existing $V$ polyline $V_{k-1}$ becomes a geodesic and that the $U$ and $W$ polylines have the geodesic property at vertices of $V_{k-1}$. One does not yet care about the geodesic property of $V_k$.

*GGG propagation by R. Sauer.* Sauer [8] introduced a GGG propagation method illustrated in Fig. 4. The vectors $\mathbf{u}_{i,1}$, $\mathbf{v}_{i,1}$ and $\mathbf{w}_{i,1}$ are the edge vectors of the three curves cast from a point $\mathbf{p}_i$, and $\mathbf{u}_{i,0}$, $\mathbf{v}_{i,0}$ and $\mathbf{w}_{i,0}$ are cast to $\mathbf{p}_i$. Obviously, $\mathbf{u}_{i-n,1} = \mathbf{u}_{i,0}$ if $\mathbf{p}_{i-n}$ and $\mathbf{p}_i$ are on the same $U$ polyline. Analogously, $\mathbf{v}_{i-1,1} = \mathbf{v}_{i,0}$ if $\mathbf{p}_{i-1}$ and $\mathbf{p}_i$ are on the same $V$ polyline, $\mathbf{w}_{i-n+1,1} = \mathbf{w}_{i,0}$ if $\mathbf{p}_{i-n+1}$ and $\mathbf{p}_i$ are on the same $W$ polyline. To compute the vertices $\mathbf{p}_{i+n-1}, \mathbf{p}_{i+n}, \ldots$ of the propagated curve $V_k$, Sauer's method uses angle constraints of GGG webs: the one-ring neighborhood of an inner vertex $\mathbf{p}_i$ has 6 edges connecting to it, forming 6 angles each of which is between every 2 neighboring edges. A GGG web requires equality of opposite angles, i.e.,

$$\alpha_0 = \alpha_1, \beta_0 = \beta_1, \gamma_0 = \gamma_1. \tag{12}$$

As we propagate a boundary layer of $S$, only the newly added vertices $\mathbf{p}_{i+n-1}, \mathbf{p}_{i+n}, \ldots$ are unknown, but the angles $\alpha_0, \beta_0, \gamma_0$ around the vertex $\mathbf{p}_i$ are already fixed (Fig. 5). Similarly, $\gamma_2$ is also fixed, thus the shape of the triangle $\mathscr{T}_1$ is determined since the angles $\alpha_1, \gamma_3$ and the edge vector $\mathbf{v}_{i,1}$ are given. The same holds for the other triangles attached to edges of $V_{k-1}$, e.g. $\mathscr{T}_0$ in Fig. 5. Each of them has one degree of freedom, expressed by rotation about the common edge with $V_{k-1}$, which has to be used to achieve the correct $\beta$-angles. For example, the position of $\mathbf{p}_{i+n+1}$ can be solved by rotating $\mathscr{T}_1$ along edge $\mathbf{v}_{i,1}$ such that the angle $\beta_1$ between $\mathbf{w}_{i,1}$ and $\mathbf{u}_{i,1}$ equals $\beta_0$.

However, finding the right triangle positions with correct angles $\beta$ can lead to ambiguous results: Assume $\mathscr{T}_0$ is already computed and we want to find $\mathscr{T}_1$ with $\beta_1 = \beta_0$; see Fig. 5. When $\mathscr{T}_1$ rotates around $\mathbf{v}_{i,1}$, $\mathbf{w}_{i,1}$ is on a rotational half-cone $C_0$ with apex $\mathbf{p}_i$, $\mathbf{v}_{i,1}$ defining the axis, and angle $\alpha_1$ between axis and rulings. Regarding $\mathbf{u}_{i,1}$ as given, a constant angle $\beta_1 = \beta_0$ between $\mathbf{w}_{i,1}$ and $\mathbf{u}_{i,1}$ means that $\mathbf{w}_{i,1}$ is on a half-cone $C_1$ with $\mathbf{u}_{i,1}$ defining the axis. Since $C_0$ and $C_1$ have a shared apex $\mathbf{p}_i$, there could be 0, 1, or 2 intersection lines between the two cones, meaning that Eq. 12 may not have solutions, or have 1 or 2 solutions. If there is no solution, the point $\mathbf{p}_{i+n+1}$ cannot be determined. If there are 2 solutions, it is also hard to pick the one that best fulfills the global fairness of $S$: this ambiguity of the orange triangles in Fig. 5 causes many local minima for the global optimization based on the constraints in Eq. 12, leading to optimization failures as shown in Fig. 12 (a).

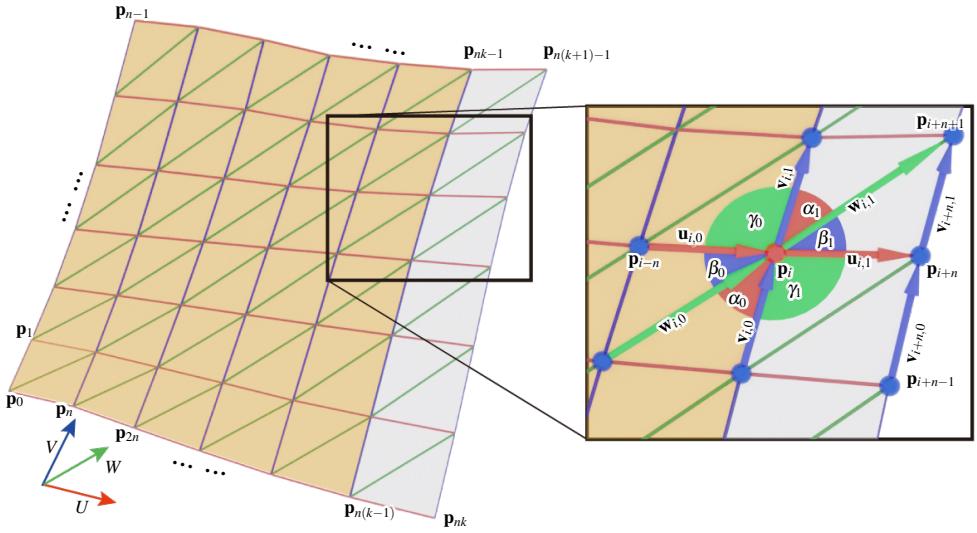



Figure 4: The red, blue, and green polylines are the $U$, $V$ and $W$ curves, as indicated by the arrows in the lower left corner. The gray strip is generated by the propagation from the boundary curve $V_{k-1} = \{\mathbf{p}_{n(k-1)}, \ldots, \mathbf{p}_{nk-1}\}$ to $V_k = \{\mathbf{p}_{nk}, \ldots, \mathbf{p}_{n(k+1)-1}\}$. The closeup shows GGG propagation of Sauer [8]. To compute the new boundary $V_k$, Sauer uses a discretization of GGG webs based on the angles between the edges connecting to the vertices $\mathbf{p}_i, i \in (n(k-1), nk-1)$ on the curve $V_{k-1}$: $\alpha_0 = \alpha_1$, $\beta_0 = \beta_1$, $\gamma_0 = \gamma_1$. However, these constraints lead to ambiguous computational results.

*Our GGG propagation method.* Our approach is based on slightly less restrictive constraints and uses discrete osculating planes, to guarantee that the propagation can always proceed. As shown in Fig. 6, we compute the surface normal vectors $\mathbf{n}_i$ and $\mathbf{n}_{i-1}$ at $\mathbf{p}_i$ and $\mathbf{p}_{i-1}$. Since the $V$ curve passing through $\mathbf{p}_i$ is a geodesic curve, the discrete surface

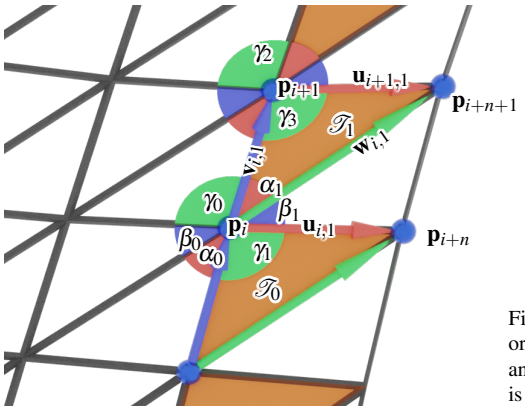



Figure 5: Sauer's propagation method rotates the orange triangle $\mathscr{T}_0$ around the edge $\mathbf{v}_{i,1}$ to meet the angle constraints Eq. 12. However, the computation is not trivial, and it does not always have solutions.

normal vectors should be equivalent to the principal normal vectors of the $V$ curve, thus,

$$\mathbf{q}_0 = \frac{\mathbf{v}_{i,0}}{\|\mathbf{v}_{i,0}\|}, \mathbf{q}_1 = \frac{\mathbf{v}_{i,1}}{\|\mathbf{v}_{i,1}\|}, \mathbf{n}_i = \frac{\mathbf{q}_0 - \mathbf{q}_1}{\|\mathbf{q}_0 - \mathbf{q}_1\|}. \tag{13}$$

It means that the normal vector $\mathbf{n}_i$ is the angle bisector of the two edge vectors $\mathbf{v}_{i,0}$ and $-\mathbf{v}_{i,1}$. Similarly, we compute vectors $\tilde{\mathbf{u}}_{i,1}$ and $\tilde{\mathbf{w}}_{i-1,1}$ such that $\mathbf{n}_i$ bisects $\mathbf{u}_{i,0}$ and $-\tilde{\mathbf{u}}_{i,1}$, $\mathbf{n}_{i-1}$ bisects $\mathbf{w}_{i-1,0}$ and $-\tilde{\mathbf{w}}_{i-1,1}$, separately. Note here the vectors $\tilde{\mathbf{u}}_{i,1}$ and $\tilde{\mathbf{w}}_{i-1,1}$ are only presenting directions but not edge vectors, thus taken as unit vectors. In general, the two rays cast from $\mathbf{p}_i$ and $\mathbf{p}_{i-1}$ in their directions $\tilde{\mathbf{u}}_{i,1}$ and $\tilde{\mathbf{w}}_{i-1,1}$ will not meet at the same point, thus we compute points

$$\tilde{\mathbf{p}}_0(t_0) = \mathbf{p}_i + t_0\tilde{\mathbf{u}}_{i,1}, \tilde{\mathbf{p}}_1(t_1) = \mathbf{p}_{i-1} + t_1\tilde{\mathbf{w}}_{i-1,1}, \tag{14}$$

where $t_0, t_1$ are non-negative numbers that are yet unknown, and take the average $\mathbf{p}_{i+n} = (\tilde{\mathbf{p}}_0 + \tilde{\mathbf{p}}_1)/2$ as the propagated vertex position. As we illustrated in Fig. 5, the shape of the triangle $\mathscr{T}_0$ is determined by the angle constraints Eq. 12, thus the lengths $l_0$ and $l_1$ of the $U$ and $W$ edges (marked as $\tilde{\mathbf{u}}_{i,1}$ and $\tilde{\mathbf{w}}_{i-1,1}$ in Fig. 6) can be obtained: $l_0 = \|\mathbf{u}'_{i,1}\|, l_1 = \|\mathbf{w}'_{i-1,1}\|$. We compute $\tilde{\mathbf{p}}_0$ and $\tilde{\mathbf{p}}_1$ by

$$\|\tilde{\mathbf{p}}_0 - \mathbf{p}_i\| = l_0, \|\tilde{\mathbf{p}}_1 - \mathbf{p}_{i-1}\| = l_1. \tag{15}$$

Assuming that there are $k$ $V$ curves $V_0, \ldots, V_{k-1}$ before propagation, then the propagated vertices are $\mathbf{p}_{kn}, \ldots, \mathbf{p}_{n(k+1)-1}$. Using our method, the propagated boundary vertices can be constructed, except for the first 2 vertices $\mathbf{p}_{kn}$, $\mathbf{p}_{kn+1}$, and the last vertex $\mathbf{p}_{n(k+1)-1}$. The last vertex doesn't have a properly defined normal vector $\mathbf{n}_{nk-1}$ at $\mathbf{p}_{nk-1}$, since $\mathbf{p}_{nk-1}$ is also a boundary vertex, thus we take $\mathbf{n}_{kn-1} = \mathbf{n}_{kn}$ so that $\mathbf{p}_{n(k+1)-1}$ can also be simultaneously constructed. We compute the first 2 vertices using a simple fairness strategy: $\mathbf{p}_{kn} = 2\mathbf{p}_{(k-1)n} - \mathbf{p}_{(k-2)n}$ and $\mathbf{p}_{kn+1} = 2\mathbf{p}_{(k-1)n+1} - \mathbf{p}_{(k-2)n+1}$.

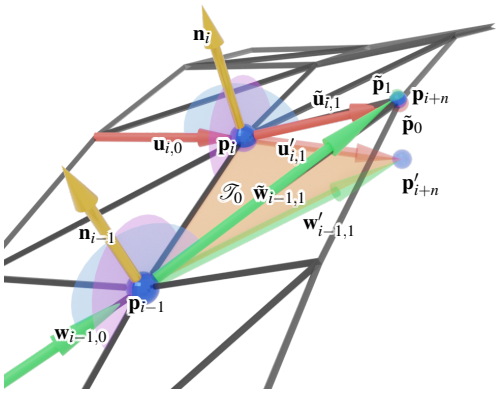
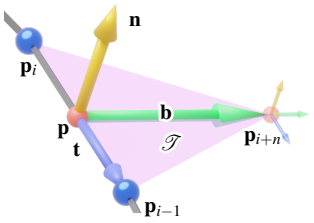



Figure 6: Our GGG propagation method. According to Sauer [8], $\mathscr{T}_0 = \triangle\mathbf{p}_i\mathbf{p}_{i-1}\mathbf{p}'_{i+n}$ is a triangle whose shape is fixed, with a rotational degree of freedom around the edge $\mathbf{p}_i\mathbf{p}_{i-1}$. Our approach approximates edge lengths of $\mathscr{T}_0$, but the main objective is to make the surface normals at vertices $\mathbf{p}_{i-1}, \mathbf{p}_i, \ldots$ discrete principle normal vectors of the polylines passing through it. Angles in the same color around the same normal vector (e.g., $\mathbf{n}_i$) indicate equality.




*Choosing the initial strip.* The initial strip shall be formed by polylines $V_0, V_1$ which in the final gridshell should be close to geodesics of an underlying smooth surface. Since propagation takes care of the geodesic property of $V_1$, we want to ensure that $V_0$ is also close to a geodesic. Although fairness in global optimization almost takes care of that, we suggest to design the initial strip as shown in the inset. We choose as $V_0$ an arbitrary fair polyline, e.g. by discrete samples of a B-spline curve. Now we consider the triangles $\mathscr{T}$ of the strip, which connect an edge $\mathbf{p}_{i-1}\mathbf{p}_i$ of $V_0$ with a vertex $\mathbf{p}_{i+n}$ of $V_1$ (see inset). Since these triangles represent approximate tangent planes of the generated surface, they should lie in approximate rectifying planes of $V_0$. Such a discrete rectifying plane attached to an edge of $V_0$ can be spanned by the unit edge vector $\mathbf{t}$ and $\mathbf{b}$ as an average of the binormal vectors at its end points $\mathbf{p}_{i-1}$ and $\mathbf{p}_i$.

The shapes of triangles matter, since their two transversal edges are initial discrete tangents of the $U$ and $W$ polylines. In practice, we first compute the midpoint $\mathbf{p} = \frac{\mathbf{p}_{i-1}+\mathbf{p}_i}{2}$ of the edge, and put $\mathbf{p}_{i+n} = \mathbf{p} + \frac{\sqrt{3}\|\mathbf{p}_i-\mathbf{p}_{i-1}\|}{2}\mathbf{b}$ such that the triangle $\mathscr{T} = \triangle\mathbf{p}_i\mathbf{p}_{i-1}\mathbf{p}_{i+n}$ is equilateral. This initial choice of points $\mathbf{p}_{i+n}$ may not be the final one. We can change these points within the planes of their triangles and even slightly outside them, to come closer to the targeted shape.

### 3.2. AGG webs

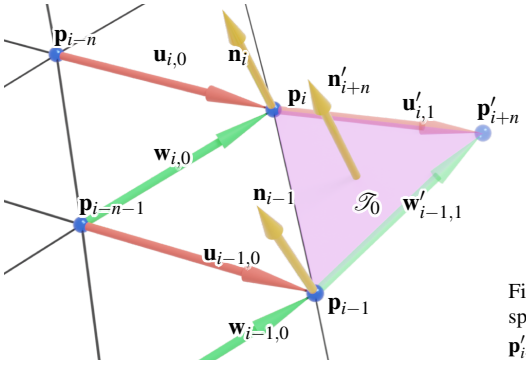



Figure 7: AGG propagation. We adjust the point $\mathbf{p}'_{i+n}$ in the plane spanned by $\mathbf{u}'_{i,1}$ and $\mathbf{w}'_{i-1,1}$ so that the $U$ and $W$ curves passing through $\mathbf{p}'_{i+n}$ are geodesics.

AGG propagation is similar to GGG propagation, but now $V$ polylines are discrete asymptotic curves. By adding a polyline $V_k$, we ensure that the existing $V$ polyline $V_{k-1}$ becomes an asymptotic curve, while the $U$ and $W$ polylines have the geodesic property at vertices of $V_{k-1}$.

Our propagation of AGG webs starts from a strip consisting of two asymptotic curves $V_0$ and $V_1$. As shown in Fig. 7, to generate the propagated curve $V_k$, we first compute the normal vectors on each vertex $\mathbf{p}_i$ of the curve $V_{k-1}$. Since $V_{k-1}$ is asymptotic, the surface normal $\mathbf{n}_i$ at $\mathbf{p}_i$ should be the binormal vector of $V_{k-1}$, which can be computed as $\mathbf{n}_i = \frac{(\mathbf{p}_i-\mathbf{p}_{i+1})\times(\mathbf{p}_{i-1}-\mathbf{p}_i)}{\|(\mathbf{p}_i-\mathbf{p}_{i+1})\times(\mathbf{p}_{i-1}-\mathbf{p}_i)\|}$. In practice, we take $\mathbf{n}_i = \frac{(\mathbf{p}_i-\mathbf{p}_{i-n})\times(\mathbf{p}_{i-1}-\mathbf{p}_i)}{\|(\mathbf{p}_i-\mathbf{p}_{i-n})\times(\mathbf{p}_{i-1}-\mathbf{p}_i)\|}$ to avoid instabilities caused by nearly collinear points $\mathbf{p}_{i-1}, \mathbf{p}_i, \mathbf{p}_{i+1}$. We assume that the normal vector $\mathbf{n}'_{i+n}$ of the generated triangle $\mathscr{T}_0 = \triangle\mathbf{p}_i\mathbf{p}_{i-1}\mathbf{p}'_{i+n}$ is the average of $\mathbf{n}_i$ and $\mathbf{n}_{i-1}$: $\mathbf{n}'_{i+n} = \frac{\mathbf{n}_i+\mathbf{n}_{i-1}}{\|\mathbf{n}_i+\mathbf{n}_{i-1}\|}$.

Computation of the position $\mathbf{p}_{i+n}$ is based on optimization which is initialized by a first guess $\mathbf{p}'_{i+n}$. Assuming that the triangle $\mathscr{T}_0$ is equilateral, this initial guess $\mathbf{p}'_{i+n}$ can be computed as $\mathbf{p}'_{i+n} = (\mathbf{p}_i + \mathbf{p}_{i-1})/2 + l\mathbf{d}$, where $l = \sqrt{3}\|\mathbf{p}_i - \mathbf{p}_{i-1}\|/2$, $\mathbf{d} = \frac{(\mathbf{p}_{i-1}-\mathbf{p}_i)\times\mathbf{n}'_{i+n}}{\|(\mathbf{p}_{i-1}-\mathbf{p}_i)\times\mathbf{n}'_{i+n}\|}$. Since $U$ and $W$ curves are geodesic curves, we update $\mathbf{p}'_{i+n}$ such that the discrete osculating plane of the $U$ curve at $\mathbf{p}_i$ defined by $\mathrm{span}(\mathbf{u}_{i,0}, \mathbf{u}'_{i,1})$ is coplanar with $\mathbf{n}_i$, and $\mathrm{span}(\mathbf{w}_{i-1,0}, \mathbf{w}'_{i-1,1})$ is coplanar with $\mathbf{n}_{i-1}$, while keeping $\mathbf{p}'_{i+n}$ always coplanar with $\mathscr{T}_0$. Thus, the constraints are:

$$\begin{aligned} \mathbf{p}''_{i+n} &= \mathbf{p}'_{i+n} + t_1\mathbf{u}'_{i,1} + t_2\mathbf{w}'_{i-1,1}, \\ \det(\mathbf{p}''_{i+n} - \mathbf{p}_i, \mathbf{p}_i - \mathbf{p}_{i-n}, \mathbf{n}_i) &= 0, \\ \det(\mathbf{p}''_{i+n} - \mathbf{p}_{i-1}, \mathbf{p}_{i-1} - \mathbf{p}_{i-n-2}, \mathbf{n}_{i-1}) &= 0, \end{aligned} \tag{16}$$

where $\mathbf{p}''_{i+n}$ is the updated position of $\mathbf{p}'_{i+n}$. We use the block-coordinate descent method [29] to solve for the variables $t_1, t_2$ for a vertex $\mathbf{p}'_{i+n}$, while assuming all other vertices are constant. We loop over $i$ such that all the vertices of $V_k$ are

updated. We take $10 \sim 20$ iterations to find the final location $\mathbf{p}_{i+n}$ of $\mathbf{p}'_{i+n}$. In Fig. 8, we show that after the adjustments, the propagation generates an approximate AGG web.

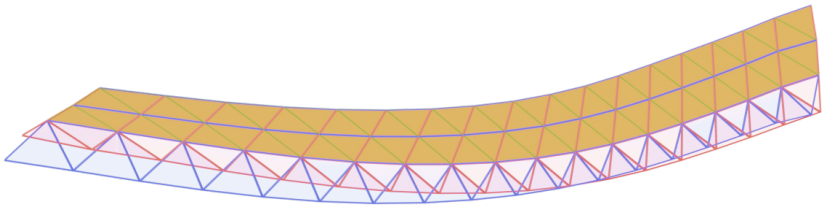


Figure 8: AGG propagation generates $V_3$ as the new boundary. The strips colored in blue and red are the propagated boundary edges before and after the adjustment. With 10 iterations, the error $E_{agg}$ (Eq. 7) evaluated on $V_2$ decreases from 2.51 to $2.68 \times 10^{-3}$.

*Curve guided AGG generation.* Our AGG propagation adds further asymptotic curves, meaning that the initial two curves $V_0, V_1$ control the shape of the whole patch $S$ generated by a pure propagation-optimization procedure. However, the two curves $V_0$ and $V_1$ provide very weak bending of the surface $S$: The principal normal vectors of the asymptotic curves are tangential to the surface, implying that the $V$ curves bend tangentially to the surface, instead of orthogonal to the surface like what the initial geodesic curves in GGG webs do. Thus, even if the initial $V_0$ and $V_1$ are curved, the surfaces tend to become flat with nearly straight geodesics unless one provides additional steering (Fig.9 (a)).

AGG webs provide sufficient flexibility for further shape guidance. We use this by prescribing a guide curve, which provides better shape control and helps to bend the surface and thus to avoid flatness. As shown in Fig. 9 (b) and (c), the blue curve is the guide curve $G = \{\mathbf{g}_0, \mathbf{g}_1, \dots\}$, where $\mathbf{g}_i$ are vertices. In each surface, the guide curve is used to guide the directions of the boundary geodesic curve $U_0 = \{\mathbf{p}_0, \mathbf{p}_n, \mathbf{p}_{2n}, \dots\}$. Thus we place the curve $G$ such that $\mathbf{g}_0 = \mathbf{p}_0$. Since the curve $V_0$ is asymptotic, the normal vector $\mathbf{n}_0$ on $\mathbf{p}_0$ should be the binormal vector $\mathbf{b}_0$ of $V_0$ on $\mathbf{p}_0$. Although both of the two vectors are not properly defined on $\mathbf{p}_0$ as it is a corner point of $S$, we can approximately take $\mathbf{b}_0 = \mathbf{b}_1$, $\mathbf{n}_0 = \frac{\mathbf{v}_{0,1} \times (\mathbf{g}_1 - \mathbf{p}_0)}{\|\mathbf{v}_{0,1} \times (\mathbf{g}_1 - \mathbf{p}_0)\|}$. It means that the edge $\mathbf{g}_0\mathbf{g}_1$ should lie in the plane spanned by the tangent vector $\mathbf{v}_{0,1}$ and the principal normal vector $\mathbf{n}_0^p = \frac{\mathbf{v}_{0,1} \times \mathbf{b}_0}{\|\mathbf{v}_{0,1} \times \mathbf{b}_0\|}$ of $V_0$. The arbitrary curves we select as guide curves shall be free of inflection points as discussed also in Sec. 3.3. Then we apply a rigid transformation to the guide curve $G$ such that $\mathbf{g}_0\mathbf{g}_1$, $\mathbf{v}_{0,1}$ and $\mathbf{n}_0^p$ are coplanar. In Sec. 4, we show how the different guide curves $G$ affect the shapes of the AGG webs. We approximate $U_0$ to $G$ using the following constraints:

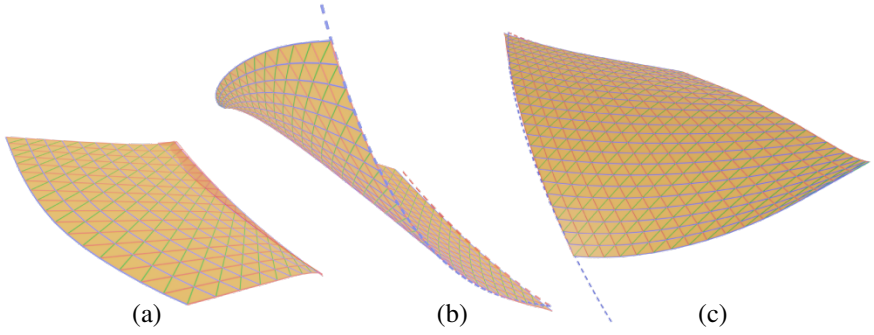



Figure 9: Approximating the initial asymptotic curves (red strip), the AGG web becomes flat (a). We use the dashed blue curve in (b) to guide the behavior of the boundary geodesic curve. The initial asymptotic curves $V_0$ of (a) and (b) are the same, marked as the red dashed line in (b). (c) is another view of (b).

$$\begin{aligned} \mathbf{p}_i &= \mathbf{g}^i, \\ \mathbf{p}_i - \mathbf{g}^i &= \mathbf{t}^i \|\mathbf{p}_i - \mathbf{g}^i\|, \end{aligned} \tag{17}$$

where $i = kn$, $k = 0, 1, \ldots$, $\mathbf{g}^i$ is the projection point of $\mathbf{p}_i$ on $G$, and $\mathbf{t}^i$ is the tangent vector of $G$ at $\mathbf{g}^i$. The first equation of Eq. 17 restricts the vertex not too far away from $G$, and the second equation only allows the vertex to move tangentially along the curve. Since we use $G$ to bend the surface, instead of approximating the initial two $V$ curves, we only approximate the first asymptotic curve $V_0$ marked in red in Fig. 9 (b). We define the objective function for approximating the guide curve as:

$$E_{guide} = \sum_i ((\mathbf{p}_i - \mathbf{g}^i)^2 + (\mathbf{p}_i - \mathbf{g}^i - \mathbf{t}^i \|\mathbf{p}_i - \mathbf{g}^i\|)^2), \tag{18}$$

with $i \in [0, 2, \ldots, 2k]$. The overall objective for AGG optimization can now be expressed as

$$E = E_c + \lambda_{fair} E_{fair} + \lambda_{appro} E'_{appro} + \lambda_{guide} E_{guide}, \tag{19}$$

where the $\lambda_{appro}$ and $\lambda_{guide}$ are the weights for the approximation to $V_0$ and $G$. The only difference of $E'_{appro}$ to $E_{appro}$ in Eq. 10 is that $i \in [0, n-1]$ for $E'_{appro}$.

*Choosing the initial strip.* Since we use guide curves to control the bending of the surfaces, the initial strip is obtained by simply taking $\mathbf{p}_{i+n} = \mathbf{p}'_{i+n}$, where $\mathbf{p}_i \in V_0$, and $\mathscr{T}_0 = \triangle \mathbf{p}_i \mathbf{p}_{i-1} \mathbf{p}'_{i+n}$ is equilateral, taking Fig. 7 as a reference. The vectors $\mathbf{n}_i$ are taken as the principal normal vectors of $V_0$ at $\mathbf{p}_i$.

Clearly, one can also use guide curves for better control of GGG webs, since those are even more flexible than AGG webs. However, the method would not work for the most restricted type of webs, namely the AAG webs discussed next.

### 3.3. AAG webs

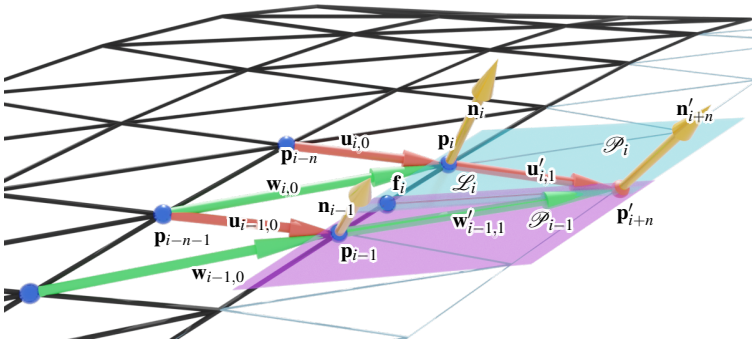



Figure 10: Our AAG propagation direction is from $V_0$ to $V_1$ which are geodesic curves. The edges colored in light gray belong to the propagation layer that is to be computed. The planes $\mathscr{P}_i$ and $\mathscr{P}_{i-1}$ colored in blue and purple are the tangent planes at $\mathbf{p}_i$ and $\mathbf{p}_{i-1}$, respectively. The thin red line $\mathscr{L}_i$ is their intersection line. We update the propogated vertex $\mathbf{p}'_{i+n} \in \mathscr{L}_i$ to an optimal position $\mathbf{p}_{i+n} \in \mathscr{L}_i$ to meet the AAG constraints.

AAG propagation takes the discrete geodesics as V curves, but differs from the previous propagation methods in the following way. One adds a polyline $V_k$ and makes sure that this polyline becomes a discrete geodesic, while maintaining the asymptotic property of the $U$ and $W$ polylines at vertices of $V_{k-1}$. This is rooted in the fact that asymptotic curves have osculating planes tangent to the surface. Thus, two families of discrete asymptotic curves form a quad mesh with planar vertex stars, known as A-net in discrete differential geometry. These vertex planes are already determined by one edge of a $U$ polyline and one edge of a $W$ polyline. Thus we know surface normals at the vertices of the added polyline $V_k$ and can make it a discrete geodesic.

We start with a strip consisting of 2 geodesic polylines $V_0, V_1$ in $V$ direction. The propagation adds $V_k$ as a geodesic polyline and extends $n$ asymptotic $U$ curves and $n-1$ asymptotic $W$ curves associated with it. As shown in Fig. 10, $\mathbf{p}_{i-1}, \mathbf{p}_i, \ldots$ are the vertices of the boundary polyline $V_{k-1} = \{\mathbf{p}_{n(k-1)}, \ldots, \mathbf{p}_{nk-1}\}$, and $\mathbf{p}'_{i+n-1}, \mathbf{p}'_{i+n}, \ldots$ are the propagated vertices to be computed. The $U$ and $W$ curves through $\mathbf{p}_i$ are asymptotic, i.e. $(\mathbf{n}_i, \mathbf{u}_{i,0}) = (\mathbf{n}_i, \mathbf{u}'_{i,1}) = 0$, $(\mathbf{n}_i, \mathbf{w}_{i,0}) = (\mathbf{n}_i, \mathbf{w}'_{i,1}) = 0$, where $\mathbf{n}_i$ is the normal vector at $\mathbf{p}_i$, $\mathbf{u}'_{i,1}$ and $\mathbf{w}'_{i,1}$ are the yet unsolved $U$ and $W$ edge vectors cast from $\mathbf{p}_i$. This means that the point $\mathbf{p}'_{i+n}$ lies on the plane $\mathscr{P}_i$ spanned by $\mathbf{u}_{i,0}$ and $\mathbf{w}_{i,0}$. Similarly, $\mathbf{p}'_{i+n}$ also lies on $\mathscr{P}_{i-1}$ spanned by $\mathbf{u}_{i-1,0}$ and $\mathbf{w}_{i-1,0}$. Hence, $\mathbf{p}'_{i+n}$ lies in the intersection line $\mathscr{L}_i = \mathscr{P}_i \cap \mathscr{P}_{i-1}$. We denote by $\mathbf{r}_i = \frac{\mathbf{n}_i \times \mathbf{n}_{i-1}}{\|\mathbf{n}_i \times \mathbf{n}_{i-1}\|}$ the unit direction vector of the line $\mathscr{L}_i$, and orient it by $\mathbf{r}_i = -\mathbf{r}_i$ if $(\mathbf{u}_{i,0}, \mathbf{r}_i) < 0$. Note that $\mathbf{n}_i$ is also the principal normal vector of the $V$ curve passing through $\mathbf{p}_i$ since $V$ curves are geodesics. The intersection line $\mathscr{L}_i$ is the ruling of the discrete rectifying strip of the $V$ curve, and the plane $\mathscr{P}_i$ is a discrete rectifying plane [7].

As for AGG propagation, we provide an initial guess for the vertex $\mathbf{p}'_{i+n}$ and update it to find the final position $\mathbf{p}_{i+n}$ through a simple optimization. We first project the polyline $V_{k-1}$ onto the discrete rectifying planes$\{\mathscr{P}_i\}$ as a curve $F$ that consists of a list of foot points $\{\mathbf{f}_i | i = n(k-1), \ldots, nk-1\}$, such that $\mathbf{f}_i \in \mathscr{L}_i$, $\frac{\mathbf{f}_i - \mathbf{f}_{i-1}}{\|\mathbf{f}_i - \mathbf{f}_{i-1}\|}$ is the unit tangent vector of $V_k$ at $\mathbf{p}_{i-1}$. Each foot point $\mathbf{f}_i$ can be found as follows:

$$\mathbf{f}_i = \mathbf{f}_{i-1} + t\mathbf{t}_{i-1}, (\mathbf{f}_i - \mathbf{p}_i, \mathbf{n}_i) = 0, \tag{20}$$

where $t \geq 0$, $\mathbf{t}_{i-1}$ is the unit tangent vector at $\mathbf{p}_{i-1}$ of the $V$ curve passing through it: $\mathbf{t}_{i-1} = \frac{\mathbf{p}_i - \mathbf{p}_{i-2}}{\|\mathbf{p}_i - \mathbf{p}_{i-2}\|}$. We take the first foot point $\mathbf{f}_{n(k-1)} = (\mathbf{p}_{n(k-1)} + \mathbf{p}_{n(k-1)+1})/2$, and the first tangent vector $\mathbf{t}_{n(k-1)} = \frac{\mathbf{p}_{n(k-1)+1} - \mathbf{p}_{n(k-1)}}{\|\mathbf{p}_{n(k-1)+1} - \mathbf{p}_{n(k-1)}\|}$. The first equation in Eq. 20 keeps $\mathbf{f}_i \in \mathscr{P}_{i-1}$, and the second one keeps $\mathbf{f}_i \in \mathscr{P}_i$. In this way, we trace the foot points $F = \{\mathbf{f}_i\}$ as the vertices of the medial line of the rectifying strip of $V_{k-1}$ [7]. The point $\mathbf{p}'_{i+n} \in \mathscr{L}_i$ then can be expressed as $\mathbf{p}'_{i+n} = \mathbf{f}_i + r\mathbf{r}_i$ with $r \geq 0$. We first find $\mathbf{p}'_{i+n}$ such that $|(\mathbf{p}'_{i+n} - \mathbf{f}_i, \mathbf{b}_i)| = 1$, where $\mathbf{b}_i$ is the binormal vector of $\mathbf{p}_i$ on curve $V_{k-1}$, then our initial estimate for $\mathbf{p}'_{i+n}$ is

$$\mathbf{p}'_{i+n} = \frac{\sqrt{3}\|\mathbf{p}_i - \mathbf{p}_{i-1}\|(\mathbf{p}'_{i+n} - \mathbf{f}_i)}{2} + \mathbf{f}_i, \tag{21}$$

by assuming that $\triangle \mathbf{p}_i \mathbf{p}_{i-1} \mathbf{p}'_{i+n}$ is an equilateral triangle.

We then find the optimal position of $\mathbf{p}'_{i+n}$. Since the propagated $V_k$ is a geodesic, and the $U$ and $W$ curves passing through $\mathbf{p}'_{i+n}$ are asymptotic, we optimize the vertices of $V_k$ such that for each $i \in (n(k-1), nk-1)$,

$$\begin{aligned} (\mathbf{n}'_{i+n}, \mathbf{u}'_{i,1}) &= 0, \\ (\mathbf{n}'_{i+n}, \mathbf{w}'_{i-1,1}) &= 0, \\ \det(\mathbf{n}'_{i+n}, \mathbf{p}'_{i+n} - \mathbf{p}'_{i+n-1}, \mathbf{p}'_{i+n+1} - \mathbf{p}'_{i+n}) &= 0. \end{aligned} \tag{22}$$

This is essentially the objective function $E_{aag}$ (Eq. 7) of the vertices of $V_k$. The constraints above are not linear, since the vertices $\mathbf{p}'_{i+n-1}, \mathbf{p}'_{i+n}, \ldots$ and the normal vectors $\mathbf{n}'_{i+n-1}, \mathbf{n}'_{i+n}$ are all unknown. We again leverage a block-coordinate descent method [29] to simplify the problem and to avoid global optimization: For each $i$, we take $\mathbf{p}'_{i+n-1}$, $\mathbf{p}'_{i+n+1}$ and $\mathbf{n}'_{i+n}$ as constant, and only adjust $\mathbf{p}'_{i+n}$. Since $\mathbf{p}'_{i+n} \in \mathscr{L}_i$, only $r$ is the variable. Loop over $i$ such that all the vertices get adjusted. Then update $\mathbf{n}'_i$ by $\mathbf{n}'_i = \frac{\mathbf{u}'_{i,1} \times \mathbf{w}'_{i-1,1}}{\|\mathbf{u}'_{i,1} \times \mathbf{w}'_{i-1,1}\|}$ for each $i$. We note that generally, the constraints Eq. 22 cannot be completely satisfied. We thus optimize for $10 \sim 20$ iterations to find the optimal positions $\mathbf{p}_{i+n}$. We show an example of AAG propagation before and after adjustments in Fig. 11. We again apply global optimization of the whole surface patch $S$ for an accurate and smooth result after propagation.

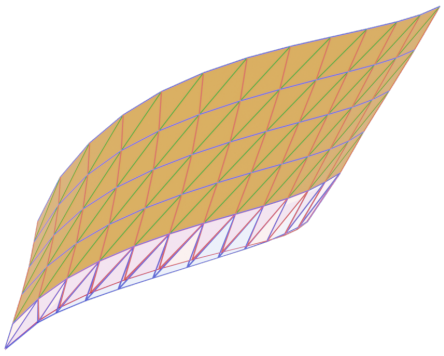
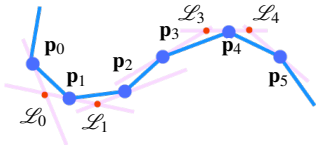


Figure 11: The strips colored in blue and red are the propagated boundary layers before and after the adjustments. With 16 iterations, the error $E_{aag}$ for the vertices on the propagated boundary curve decreases from 0.80 to 0.053.

We point out that the first curve $V_0$ should be free of inflection points since the vanishing curvature of $V_0$ at an inflection point $\mathbf{p}$ implies vanishing normal curvature of $S$ in the direction of the tangent of $V_0$ at $\mathbf{p}$, and thus a third asymptotic direction. This can only appear at flat points. Such a surface is very special and



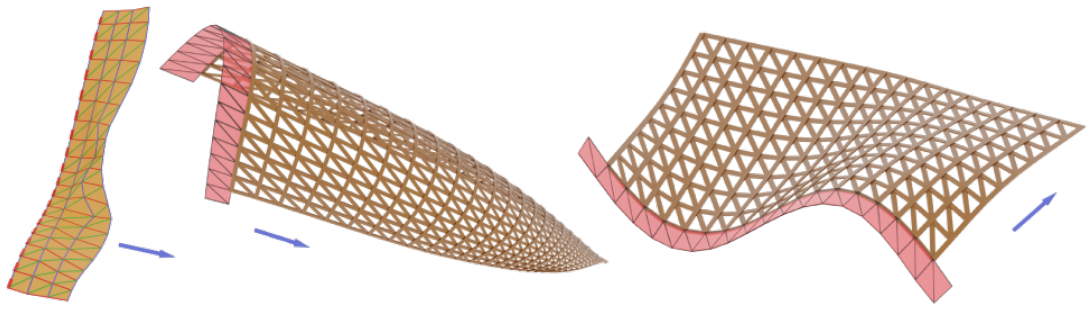
restrictive, and thus the optimization always fails. The in-feasibility can also be revealed in our discrete AAG construction. The inset is a 2D illustration: the pink lines represent the discrete rectifying planes of the vertices $\mathbf{p}_i$. The intersection lines are marked as red dots. $\mathscr{L}_2$ is not properly defined around the inflection area between $\mathbf{p}_2$ and $\mathbf{p}_3$.

*Choosing the first strip.* We generate the first strip for AAG propagation such that the initial curve $V_0$ is approximately geodesic and thus use the method for computing initial strips of GGG webs.

## 4. Results and discussion

Table 1: Evaluation of our results. $|V_c|$ is the number of vertices of the first curve $V_0$. $V$ is the number of vertices of the final GGG, AGG, or AAG web. *bbd* is the diagonal length of the axis-aligned bounding box of the web. $\lambda_{fair}$ and $\lambda_{appro}$ are the weights for the fairness and approximation terms in Eq. 11. $T_p$ is the average time (in seconds) for each propagation. For AAG and AGG, we take 10 loops of coordinate descent optimization for each propagation. $T_o/iter$ is the average global optimization time in seconds for each iteration. $E_c$ is the optimization error $E_{ggg}$, $E_{aag}$, or $E_{agg}$, depending on the type of the web.

| *Fig.* | $\lvert V_c\rvert$ | $\lvert V\rvert$ | *bbd* | $\lambda_{fair}$ | $\lambda_{appro}$ | $\lambda_{guide}$ | $T_p$ | $T_o$/*iter* | $E_c$ |
|---|---|---|---|---|---|---|---|---|---|
| 2(a) | 18 | 486 | 1.18 | 1e-4 | 1e-4 | 0.01 | 0.0153 | 0.0397 | 1.32e-5 |
| 2(c) | 36 | 1008 | 4.92 | 1e-4 | 1e-4 | 1e-4 | 0.0154 | 0.091 | 1.31e-5 |
| 3 | 20 | 240 | 9.26 | 1e-4 | 0.01 | - | 2.60e-3 | 0.0233 | 4.50e-7 |
| 9(a) | 18 | 198 | 0.80 | 1e-4 | 1e-3 | - | 0.0105 | 0.0173 | 4.38e-8 |
| 9(b) | 18 | 450 | 1.07 | 1e-4 | 1e-3 | 0.01 | 0.0116 | 0.0400 | 4.31e-6 |
| 12(b) | 20 | 480 | 5.25 | 1e-4 | 1e-4 | - | 2.13e-3 | 0.0484 | 4.74e-12 |
| 12(c) | 20 | 240 | 9.89 | 0.01 | 0.01 | - | 2.34e-3 | 0.0245 | 1.04e-7 |
| 13(d) | 32 | 288 | 3.74 | 5e-3 | 1.00 | - | 2.43e-3 | 0.0286 | 4.05e-8 |
| 14(b) | 32 | 672 | 6.06 | 1e-3 | 1e-2 | - | 2.67e-3 | 0.0767 | 6.00e-5 |
| 15 | 21 | 315 | 10.87 | 1e-4 | 0.10 | - | 3.64e-3 | 0.0356 | 1.32e-6 |
| 17(a) | 18 | 450 | 1.07 | 1e-4 | 1e-4 | 1e-3 | 0.0148 | 0.0428 | 2.32e-8 |
| 17(b) | 18 | 450 | 1.42 | 1e-4 | 1e-4 | 1e-3 | 0.0153 | 0.0396 | 1.14e-8 |
| 17(c) | 18 | 468 | 1.66 | 1e-4 | 1e-4 | 1e-4 | 0.0160 | 0.0392 | 1.18e-8 |
| 21(a) | 13 | 156 | 3.03 | 1e-4 | 1e-3 | - | 0.0161 | 0.0139 | 1.40e-5 |
| 21(b) | 13 | 156 | 3.48 | 1e-4 | 1e-4 | - | 0.0178 | 0.0147 | 1.87e-5 |
| 23(c) | 36 | 1512 | 4.18 | 1e-4 | 1.00 | - | 4.65e-3 | 0.187 | 1.66e-5 |
| 25(a) | 13 | 169 | 6.94 | 1e-4 | 1e-4 | - | 0.0166 | 0.0145 | 5.35e-5 |
| 25(b) | 13 | 221 | 3.63 | 1e-4 | 1.00 | - | 0.0160 | 0.0162 | 5.73e-5 |
| 25(c) | 36 | 1512 | 3.96 | 1e-4 | 1e-4 | - | 0.0160 | 0.188 | 6.61e-5 |



(a) (b) (c)

Figure 12: Using $E_{angle}$ (Eq. 23) instead of $E_{ggg}$ Eq. 6, optimization failed (a). While using our propagation method and optimization based on osculating planes, high-quality results are generated (b), (c). The blue arrows indicate the propagation directions.

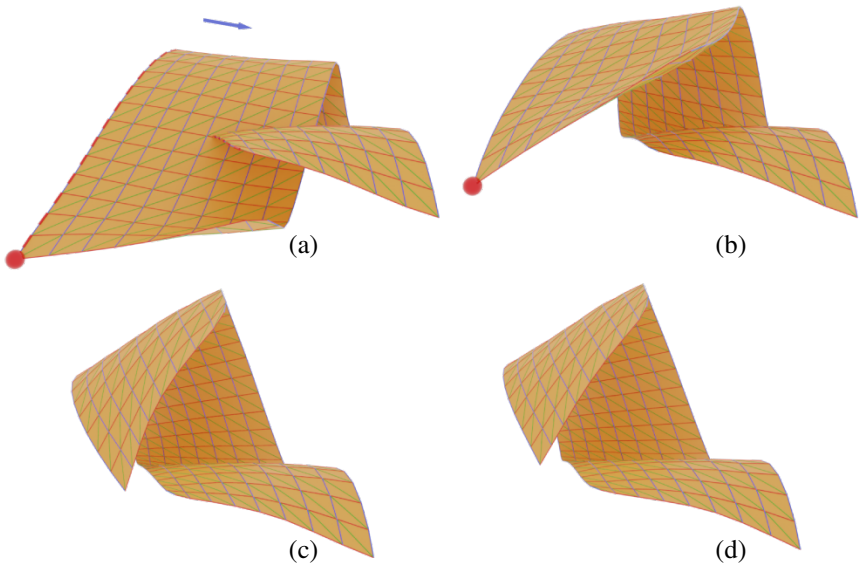



Figure 13: The GGG web in (a) has self-intersections. We pick a point and drag it to the target position marked as the red dot in (b) to modify the surface under isometric deformation. We repeat these steps until reaching a desired shape (c), which is already a GGG web with error $E_{ggg} = 8.12 \times 10^{-5}$. Optionally, we can further optimize for accuracy and smoothness (d) (maximal deviation of the corresponding vertices of (c) and (d) is 4.2% of the diagonal length of the axis-aligned bounding box of (c)). The average runtime per iteration for optimizing $E_{ggg}$ is 0.0286s (Table. 1), while optimizing $E_{iso}$ takes $9.11 \times 10^{-3}$s per iteration.

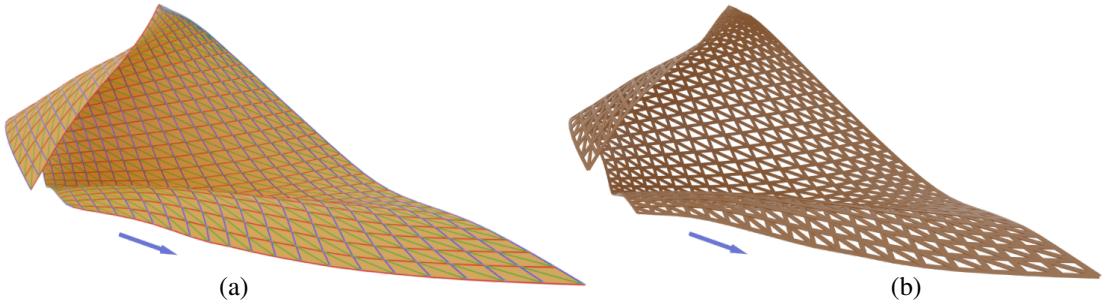



Figure 14: We continue the GGG propagation-optimization procedures after the isometric deformations in Fig. 13 (c) until the surface grows to a desired size (a). (b) is the rendering of the gridshell extracted from (a).

*Implementation.* Our algorithms include GGG, AAG, and AGG web propagation and global optimization. For GGG, we also developed an interactive design method based on isometric deformation. Our algorithms are implemented in C++. The choices of weights and the evaluations of our results are listed in Table 1. The algorithms are tested on an Intel Xeon E5-2687W 3.0 GHz processor. The implementation of our algorithms has been released as an open-source project (`https://github.com/wangbolun300/WebsViaPropagation`).

After each propagation, we apply $10 \sim 20$ iterations of global optimization to improve the quality of the webs, with default parameters $\lambda_{fair} = 10^{-4}$, $\lambda_{appro} = 1.0$, and $\lambda_{guide} = 0.1$ if applicable. After the surfaces reach the desired scales, we reduce $\lambda_{appro}$ and $\lambda_{guide}$ until the web structures are accurate enough. It can be seen from our results that the runtime on each AGG and AAG propagation is in the same magnitude as one iteration of global optimization, and for GGG, each propagation is $\sim 10\times$ faster than each iteration of global optimization. Thus, without heavy computations, our propagation methods provide high-quality initialization for optimization of GGG, AAG, and AGG web structures. In this section, the $U$, $V$, and $W$ edges of the webs are colored in red, blue, and green, respectively. The initial $V$ curves are rendered as red dashed lines. The gridshell structures are extracted using the inner vertices as explained in Fig. 3 and Wang et al. [7]. In the rendering figures of the gridshells, we highlight the first strip of the web from where each of the gridshells is extracted as a red strip and use blue arrows to indicate propagation directions.

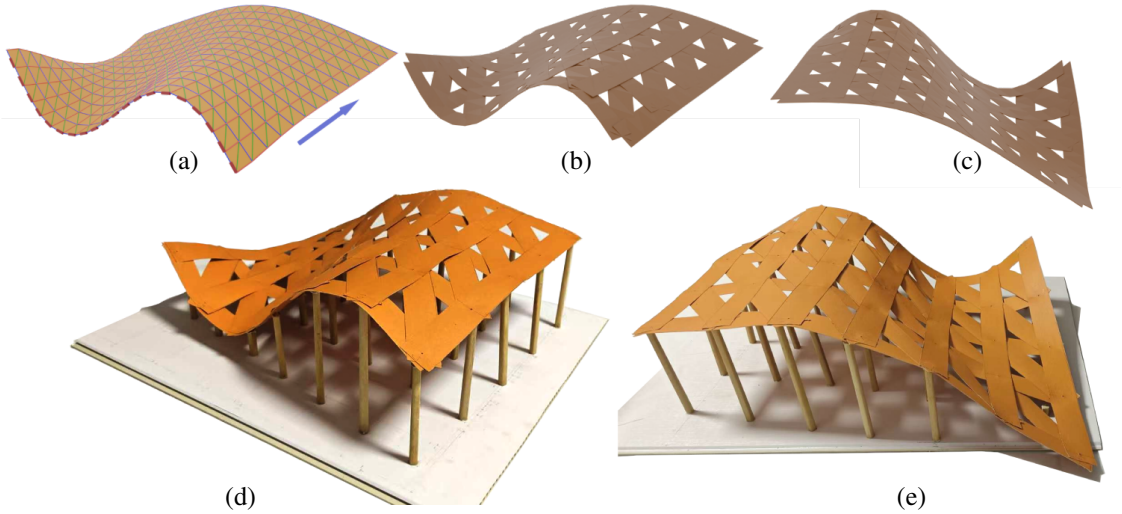



Figure 15: To validate our algorithm, we built a paper model (shown in (d) and (e)) of a GGG gridshell extracted from the GGG web (a). The shape of the web is similar to Fig. 12 (c), but with different $|V_c|$ and $|V|$ to ease the fabrication. (b) and (c) show the rendered gridshell in the views corresponding to (d) and (e), respectively.

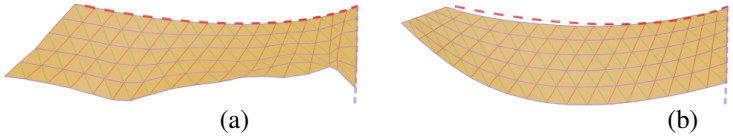



Figure 16: The AGG propagation of (a) is simply offsetting the vertices, while (b) uses our method. Both examples are generated from the same strip, approximate the same boundary curve (red dashed curve), use the same guide curve (blue dashed curve) and optimization weights. After each propagation, we take 10 iterations of global optimization for both (a) and (b). The results show that (a) is highly non-smooth, with error $E_{agg} = 36.16$, while for (b) the error is $E_{agg} = 4.81 \times 10^{-4}$. We use the default weights for the optimization as aforementioned.

*GGG webs.* We first show that the angle constraints (Eq. 12) employed by Sauer [8] are too restrictive and thus not feasible for GGG web optimization. In Fig. 12 (a), we take the same first strip as in Fig. 3, but optimize $E_{angle}$ that is defined as

$$E_{angle} = \sum_i \Big( \Big( \frac{(\mathbf{v}_{i,1}, \mathbf{w}_{i,1})}{\|\mathbf{v}_{i,1}\| \|\mathbf{w}_{i,1}\|} - \frac{(\mathbf{v}_{i,0}, \mathbf{w}_{i,0})}{\|\mathbf{v}_{i,0}\| \|\mathbf{w}_{i,0}\|} \Big)^2 + \Big( \frac{(\mathbf{u}_{i,1}, \mathbf{w}_{i,1})}{\|\mathbf{u}_{i,1}\| \|\mathbf{w}_{i,1}\|} - \frac{(\mathbf{u}_{i,0}, \mathbf{w}_{i,0})}{\|\mathbf{u}_{i,0}\| \|\mathbf{w}_{i,0}\|} \Big)^2 + \Big( \frac{(\mathbf{u}_{i,1}, \mathbf{v}_{i,0})}{\|\mathbf{u}_{i,1}\| \|\mathbf{v}_{i,0}\|} - \frac{(\mathbf{u}_{i,0}, \mathbf{v}_{i,1})}{\|\mathbf{u}_{i,0}\| \|\mathbf{v}_{i,1}\|} \Big)^2 \Big), \tag{23}$$

for the angle constraints Eq. 12, instead of $E_{ggg}$ in Eq. 6. The optimization eventually failed, generating very non-smooth and low-accuracy results. On the other hand, using the proposed objective $E_{ggg}$ yields high-quality GGG webs (Fig. 12(b) (c)).

Since the size of the surface grows as the propagation goes on, the surface may self-intersect even if the input strips don't. However, this is not acceptable for gridshell design. One may directly modify the input strip and rerun the optimization with the hope that there are no self-intersections. Note that GGG webs are less restrictive and contain more degrees of freedom than AAG and AGG webs. Since geodesic curves map to geodesics under isometric transformations, we employ an *isometry-based interactive design method* to help the designers with shape modifications. Our method allows users to pick a vertex on the web, and drag the vertex to a user-specific position. During the deformations, we replace the energy term $E_{ggg}$ with $E_{iso}$ as defined in [30]. We add the term $E_{ver} = (\mathbf{p}_{select} - \mathbf{p}_{target})^2$ to place the selected point $\mathbf{p}_{select}$ to the target position $\mathbf{p}_{target}$. The overall objective function is

$$E = E_{iso} + \lambda_{appro} E_{ver} + \lambda_{fair} E_{fair}. \tag{24}$$

We cancel the approximation to the first strip so that we can also modify it by selecting $\mathbf{p}_{select}$ on it (Fig. 13 (a) (b)). The

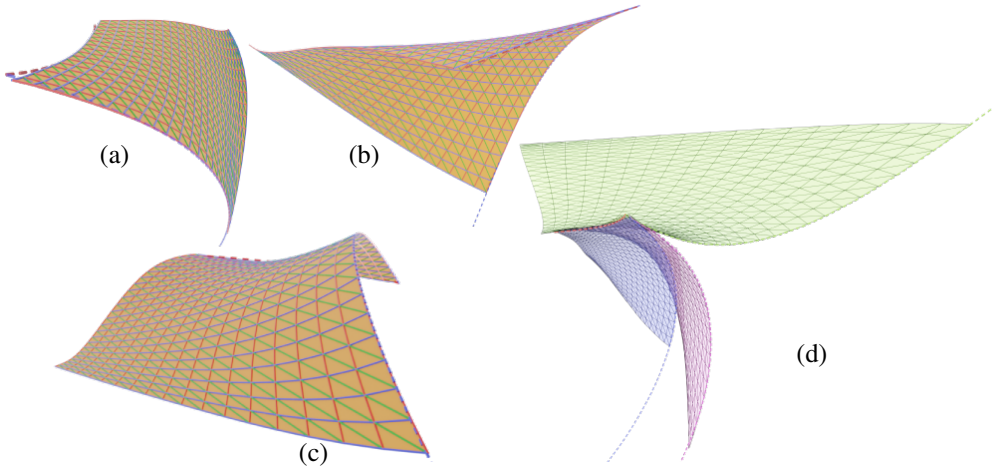



Figure 17: (a), (b), and (c) are the AGG webs with a shared initial boundary $V_0$ (red). The shapes differ a lot due to the different choices of guide curves (blue). (d) shows a side-by-side comparison of (a), (b), and (c) which are colored in blue, purple, and green, respectively. A rendering of the gridshell extracted from (c) is presented in Fig. 2 (b).

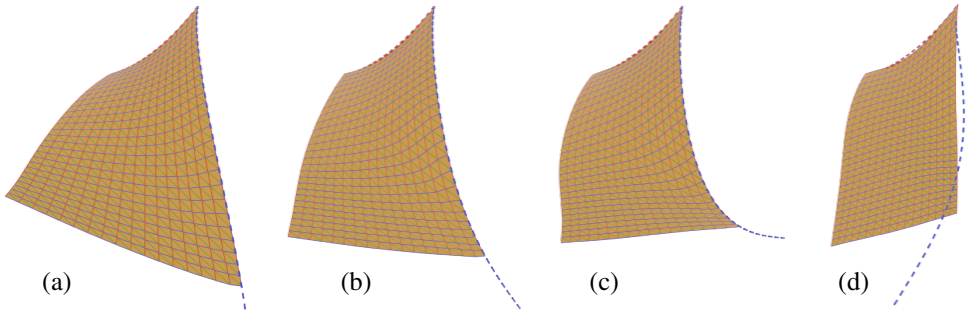



Figure 18: Interactively steering the guide curves for AGG optimization. (a) is the same surface as in Fig. 17 (b). After generating (a), we modify its shape by interactively steering the guide curves to obtain different AGG webs (b) and (c). (d) is a failure case where the boundary of the web cannot follow the guide curve.

web obtained from isogeometric deformation (c) is already a GGG web with error $E_{ggg} = 8.12 \times 10^{-5}$. We optimize (c) for an accurate and smooth GGG web (d). The isometric deformation provides a $3\times$ acceleration compared with $E_{ggg}$ since $E_{iso}$ doesn't need any auxiliary variables like normal vectors or binormal vectors. After the interactive deformations, we can continue our propagation-optimization procedures to expand the patch into the desired size (Fig. 14).

Fig. 15 shows a paper model fabricated from a GGG web generated using our propagation-optimization method. The size of the web in Fig. 15 (a) is $21 \times 15$, with 18 inner $U$ polylines, 13 inner $V$ polylines, and 29 inner $W$ polylines where we extract each of the strips from. We take one from every three polylines to downsample the model. Then the paper strips are placed such that the medial lines of the strips match the polylines, forming a final structure with 7 $U$ strips, 5 $V$ strips, and 9 $W$ strips. The side-by-side comparisons (b)-(d) and (c)-(e) show that the fabricated result matches the digital model well in appearance.

*AGG webs.* In Fig. 16, we show the effectiveness of our AGG propagation method. The only difference between the simple strategy in Fig. 16 (a) and our method in (b) is that (a) doesn't update the vertices to meet the constraints of Eq. 16. Both (a) and (b) are generated using the same procedures and optimization weights. It shows that our method (b) converges well, while simply offsetting (a) yields a highly non-smooth web after global optimization.

We also show how the guide curves affect the shapes of AGG webs. In Fig. 17, the AGG webs (a), (b), and (c) have a common boundary curve $V_0$. The surfaces behave differently (d) with different guide curves, meaning that the

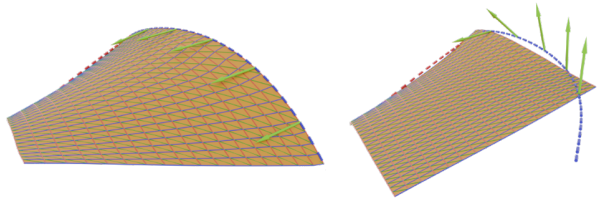


Figure 19: The left and right figures show another view of Fig. 18 (b) and (d). The green arrows are the binormal vectors of the different guide curves. The design of Fig. 18 (d) fails because the binormal vectors of the guide curve are almost orthogonal to the surface, thus violating the principle of the AGG constraints.

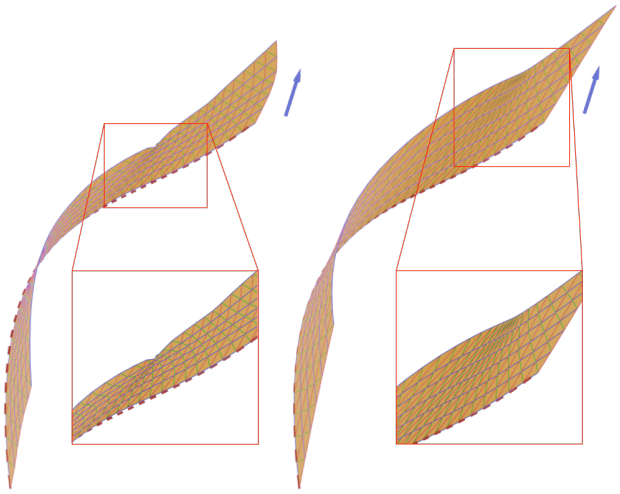


Figure 20: The propagation for the AAG web on the left are simply offsetting the boundaries, while for the one on the right we use our method. Both of the two webs start from the same strips, and after each propagation, we take 3 iterations of global optimization. The results show that the simple offset strategy failed, containing self-intersections and flipped elements, with optimization error $E_{aag} = 5.21$. Our method generates a high-quality result with $E_{aag} = 5.81 \times 10^{-6}$. We use the default weights for the optimization as aforementioned.

varieties in the guide curves lead to the varieties in the AGG webs. Fig. 18 shows examples where the guide curves are modified after generating the AGG webs (a) to add extra varieties (b)-(c). However, improperly placing the guide curves may cause undesired design results (d). As explained in Fig. 19, the placing of the guide curve should not violate the constraints of AGG webs: the guide curve acts as a geodesic curve of the web, thus the binormal vectors of it should lie (almost) tangentially to the surface. More AGG gridshells are presented in Fig. 2.

*AAG webs.* We show that our AAG propagation method is effective in Fig. 20. We take a simple strategy as a comparison to our method: we offset the boundary in the direction of the binormal vectors of $V$ curves such that the triangles $\triangle \mathbf{p}_i \mathbf{p}_{i-1} \mathbf{p}_{i+n}$ are equilateral (take Fig. 7 as a reference). After a few propagation-optimization steps, the generated surface patch is highly irregular, leading to construction failures. Under the same number of steps, our method generates AAG webs of higher quality. In Fig. 21, we show that the shapes of the AAG webs strongly depend on the first strip, formed by $V_0$ and $V_1$. Even if $V_0$ is the same, different $V_1$ curves imply a clearly visible shape difference. More AAG webs are shown in Fig. 25.

In Fig. 22, we show the geometric error $E_{aag}$ and the average fairness energy per vertex $E_f/|V|$ for the global optimizations of the first 5 propagations of Fig. 21 (a). For each propagation, we take 5 iterations of global optimization to improve the quality and accuracy. We compare our propagation method (green lines) with the simple offsetting strategy (red lines). It shows that for both strategies, the errors created by the new-added layers can be gradually decreased by the global optimization. Our strategy outperforms the simple offsetting strategy since our method creates boundary layers with smaller errors thus improving the convergence. Since the simple offsetting strategy only ensures smoothness, its fairness is better than our method for the first 10 iterations. However, as the meshes expand and the total numbers of iterations grow, both strategies achieve smooth webs.

In Fig.23, we show examples of constructing GGG, AGG, and AAG webs from the same initial boundary curve. The GGG and AAG propagation starts from the same boundary strip, which is distinctive from the one that generates AGG, since for GGG and AAG, the initial boundary curves are geodesics, while for AGG, the first boundary curve is asymptotic.

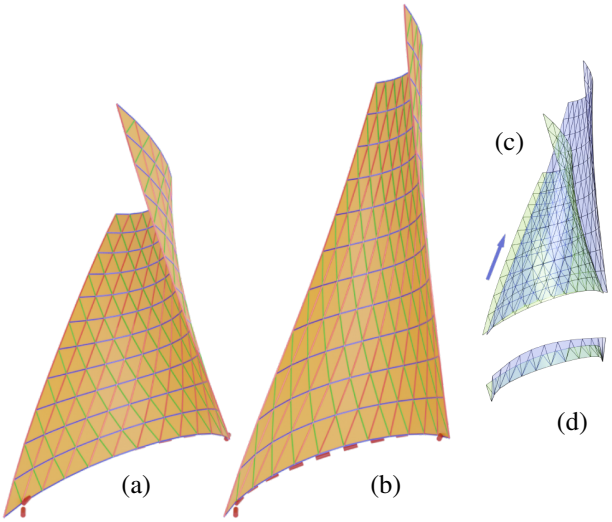



Figure 21: The first curves $V_0$ of the initial strips for AAG webs (a) and (b) are the same, but the shapes of (a) and (b) are a lot different due to the different second $V$ curves $V_1$. In (c), the green and blue surfaces are for (a) and (b), respectively. The initial strips for the two webs are shown in (d).

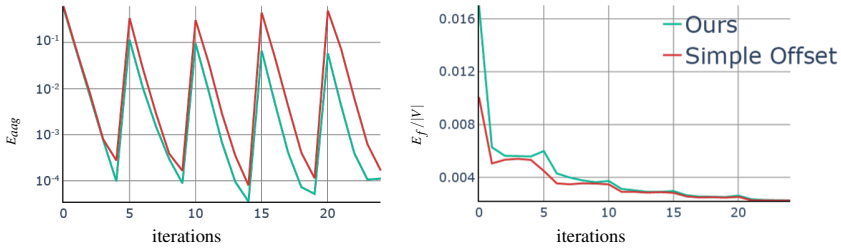



Figure 22: The plots show the geometric error $E_{aag}$ (left) and the average fairness energy per vertex $E_f/|V|$ (right) for the global optimization of the first 5 propagations of Fig. 21 (a). The $E_{aag}$ are shown in log.

*Usage as a design tool.* One may get the impression that the design approach based on propagation is hard to control and provides little room for the user to control the shape, apart from the use of guide curves in AGG or GGG webs. However, this is not the case. One obtains much closer control by interleaving between propagation and shape editing. After one or several propagation steps, one can change the so far generated web via standard editing tools such as relocation of selected vertices, as shown in Fig. 26 and Fig. 27. GGG webs may also be edited via isometric deformation, as shown in Fig. 14

There is also no need to work with the boundaries as discussed above. One may cut out desirable parts for the final design, as shown in Fig. 28.

*Summary.* In this section, we have shown the effectiveness of our methods and presented different ways to add extra varieties during or after the web generation. The first boundary curves, the shapes of the first strips, the guide curves, and the editing points are all key factors that affect the final shapes. During the propagation, we estimate the shapes of the boundary strips to be generated. It generates shapes that naturally meet the hard constraints that approximate the first strip, while sacrificing the accuracy ($E_{ggg}$, $E_{aag}$, and $E_{agg}$) and smoothness ($E_{fair}$). Global optimization is later adopted to improve accuracy and smoothness as shown in Fig. 22. The approximation to the guide curves and the edited points are also expressed as energy terms such that they can be included in the global optimization schemes. Moreover, we observe that the global optimization re-parametrizes surfaces to obtain overall high-quality webs. Along the propagation direction of the AGG web in Fig. 2 (c), the width between every two neighboring $V$ curves varies a lot, as shown in Fig. 24, although our propagation algorithm generates boundary strips with almost equal widths. However, as the propagation goes on, the areas far from the initial strip tend to become flat since the influence of the boundary strip becomes weaker and the fairness term $E_{fair}$ in the global optimization dominates, as shown in Fig. 12. The guide curves and the edited points can be adopted as powerful tools to help the designers deform these areas.

However, these guide curves or edited points cannot be arbitrarily placed. They should still follow the principles of the corresponding web constraints (as explained in Fig. 19). In practice, the users may start from a straight line as a guide curve to generate an almost flat patch (which is likely to fulfill the corresponding web constraints and fairness conditions), then gradually deform the guide curve as desired, or take several steps to deform the web by dragging the

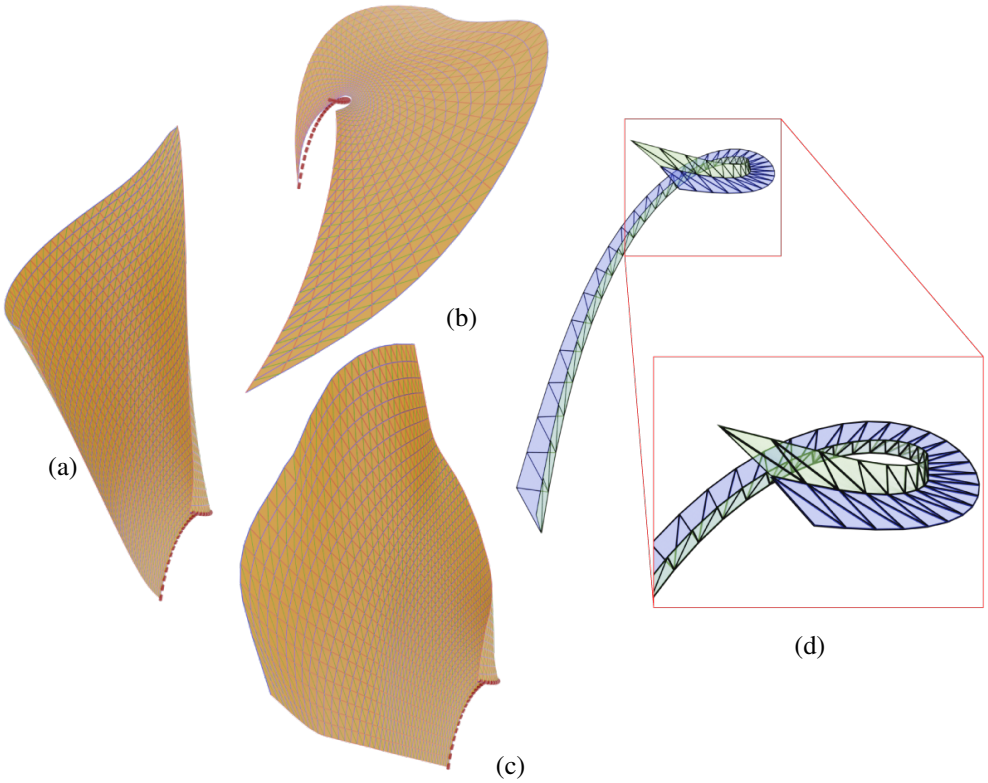



Figure 23: The 3 webs are AAG (a), AGG (b) and GGG (c) propagated from the same initial boundary curve (red dashed lines). The green strip in (d) is the initial boundary strip for both AAG and GGG, and the blue strip is the initial boundary strip for AGG. The AAG web (a) is the same surface as in Fig.25 (c), and the AGG (b) web is the same as in Fig. 2 (c). The quality of the GGG web is shown in Table 1.

selected point to a position not too far away in each step, to avoid failures caused by the infeasibilities of guide curves and the target positions of selected points.

### *4.1. Conclusion and future research*

Extending recent work by Schling et al. [6] on hybrid asymptotic geodesic gridshells, we presented a computational design approach to the underlying webs. We included the geodesic webs that had been discussed at first by Sauer [8]. The focus of our discussion has been on propagation from initial strips, and we mentioned ways to embed this process into a design environment.

However, we did not develop a complete user interface for design, as this is not really a research topic. For example, one can think of various ways to support the user in designing the initial strip. Keep in mind that not only the shapes of the two $V$ curves $V_0, V_1$ matter, but also the transversal edges, since those determine the directions in which the $U$ and $W$ curves start to grow during propagation.

While we provided an effective way to access the design space of the considered webs, one still knows little about the essential limitations in their shapes. This is also related to the lack of knowledge on explicit examples, which – for AAG and AGG webs – does not go beyond rather trivial rotational or cylindrical shapes mentioned in [6]. Thus, future work may try to better understand the shape limitations present in the considered webs. One could also develop propagation algorithms for webs containing pseudo-geodesics. Those belong to lamellas which are inclined against the reference surface under a constant angle different from 0 and 90 degrees. Gridshells based on pseudo-geodesics have already received attention [31, 7].

*Acknowledgements*

This research has been supported by KAUST baseline funding (grant BAS/1/1679-01-01).

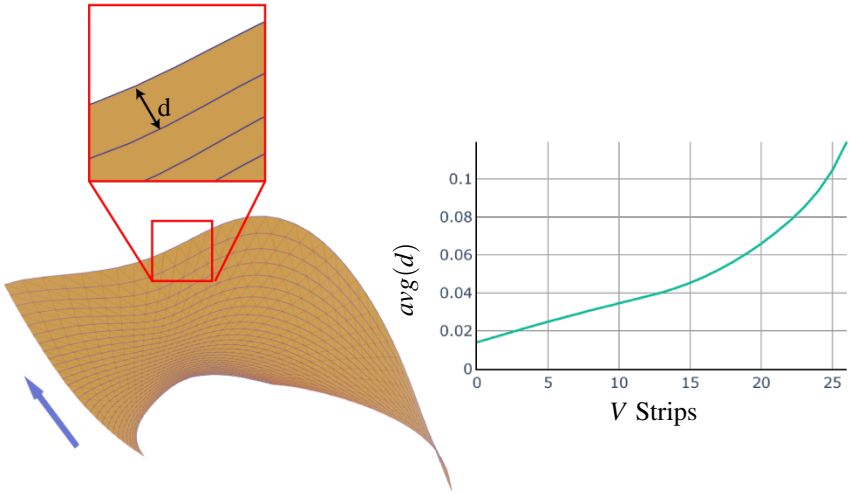



Figure 24: The AGG web is the same as in Fig. 2 (c) and Fig. 23 (b). We compute the average width of each of the $V$ strips, e.g. the average of the distance $d$ between every two neighboring asymptotic curves, along the propagation direction, as shown in the closeup figure. The average strip width $avg(d)$ of each strip is plotted on the right, showing a significant variety and a smooth transition in strip widths along the propagation direction.

## References


[1] E. Schling, D. Hitrec, J. Schikore, R. Barthel, Design and construction of the asymptotic pavilion, in: K.-U. Bletzinger, E. Oñate, B. Kröplin (Eds.), VIII International Conference on Textile Composites and Inflatable Structures, International Center for Numerical Methods in Engineering (CIMNE), 2017, pp. 178–189.

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

[3] E. Schling, M. Kilian, H. Wang, D. Schikore, H. Pottmann, Design and construction of curved support structures with repetitive parameters, in: L. H. et al. (Ed.), Adv. in Architectural Geometry, Klein Publ. Ltd, 2018, pp. 140–165.

[4] J. Schikore, A. M. Bauer, R. Barthel, K.-U. Bletzinger, Large torsion on elastic lamella grid structures, in: C. Lázaro, K.-U. Bletzinger, E. Oñate (Eds.), FORM and FORCE 2019, International Centre for Numerical Methods in Engineering (CIMNE), Barcelona, 2019, pp. 788–795.

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

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

[7] B. Wang, H. Wang, E. Schling, H. Pottmann, Rectifying strip structures, ACM Trans. Graphics 42 (6) (2023) 256:1–256:19.

[8] R. Sauer, Flächen mit drei ausgezeichneten Systemen geodätischer Linien, die sich zu einem Dreiecksnetz verknüpfen lassen, Sitz. Bayer. Akad. Math.-nat. Abt. (1926) 353–397.

[9] W. Blaschke, G. Bol, Geometrie der Gewebe, Springer, 1938.

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

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

[12] O. Volk, Über Flächen aus geodätischen Dreiecksnetzen, Sitzungsber. Heidelb. Akad. Wiss. Math.-Natur. Kl. 1929 (1) (1929) 3–32.

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

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

[15] A. I. Bobenko, Y. B. Suris, Discrete differential geometry. Integrable structure, Vol. 98 of Graduate Studies in Mathematics, American Mathematical Society, Providence, RI, 2008.

[16] A. Voss, Über diejenigen Flächen, auf denen zwei Scharen geodätischer Linien ein conjugirtes System bilden, Sitzungsber. Bayer. Akad. Wiss., math.-naturw. Klasse (1888) 95–102.

[17] W. Wunderlich, Zur Differenzengeometrie der Flächen konstanter negativer Krümmung, Österreich. Akad. Wiss. Math.-Nat. Kl. S.-B. IIa. 160 (1951) 39–77.

[18] M. Rabinovich, T. Hoffmann, O. Sorkine-Hornung, The shape space of discrete orthogonal geodesic nets, ACM Trans. Graph. 37 (6) (2018) 228:1–228:17.

[19] M. Rabinovich, T. Hoffmann, O. Sorkine-Hornung, Discrete geodesic nets for modeling developable surfaces, ACM Trans. Graph. 37 (2) (2018) 16:1–16:17.

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

[21] H. Pottmann, Q. Huang, B. Deng, A. Schiftner, M. Kilian, L. Guibas, J. Wallner, Geodesic patterns, ACM Trans. Graphics 29 (4) (2010) 43,1–10.

[22] B. Deng, H. Pottmann, J. Wallner, Functional webs for freeform architecture, Computer Graphics Forum 30 (2011) 1369–1378, proc. Symp. Geometry Processing.

[23] J. Vekhter, J. Zhuo, L. F. G. Fandino, Q. Huang, E. Vouga, Weaving geodesic foliations, ACM Transactions on Graphics (TOG) 38 (4) (2019) 1–22.

[24] P. Ayres, A. G. Martin, M. Zwierzycki, Beyond the basket case: A principled approach to the modelling of kagome weave patterns for the fabrication of interlaced lattice structures using straight strips, in: Advances in Architectural Geometry 2018, Chalmers University of Technology, 2018, pp. 72–93.

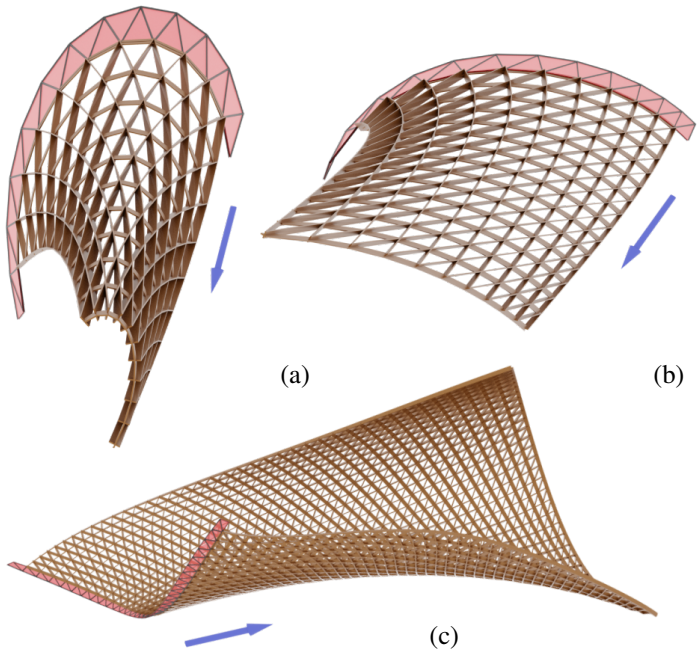



Figure 25: AAG webs generated with our method. Detailed choices of the weights and the evaluations on the quality are shown in Table 1.

[25] P. Ayres, S. Bornaz, A. Orlinski, M. Heimrath, A. G. Martin, Architectural scale kagome weaving: Design methods and fabrication concepts, in: FABRICATE: Design & Making, UCL Press, 2020, pp. 178–185.
[26] Y. Ren, J. Panetta, T. Chen, F. Isvoranu, S. Poincloux, C. Brandt, A. Martin, M. Pauly, 3d weaving with curved ribbons, ACM Transactions on Graphics (TOG) 40 (4) (2021) 1–15.
[27] C. Baek, A. G. Martin, S. Poincloux, T. Chen, P. M. Reis, Smooth triaxial weaving with naturally curved ribbons, Physical Review Letters 127 (10) (2021) 104301.
[28] K. Madsen, H. B. Nielsen, O. Tingleff, Methods for non-linear least squares problems (2nd ed.), 2004.
[29] S. J. Wright, Coordinate descent algorithms, Mathematical programming 151 (1) (2015) 3–34.
[30] C. Jiang, C. Wang, F. Rist, J. Wallner, H. Pottmann, Quad-mesh based isometric mappings and developable surfaces, ACM Trans. Graphics 39 (4) (2020) 128:1–128:13.
[31] R. Mesnil, O. Baverel, Pseudo-geodesic gridshells, Engineering Structures 279 (2023) 115558.

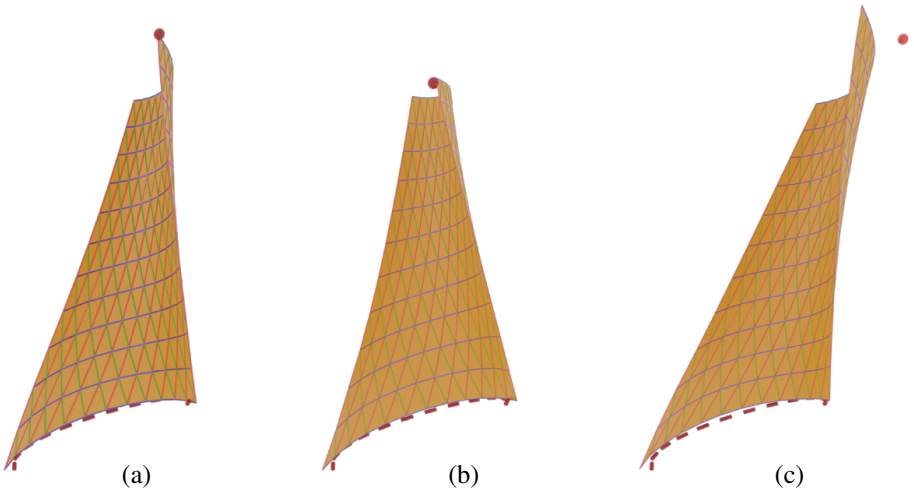



Figure 26: (a) is the same AAG web as in Fig. 21 (b). Similarly to the interactive design of GGG (Eq. 24), we select a point (red dot) on the web and drag it to the target position while optimizing $E = E_{aag} + \lambda_{fair}E_{fair} + \lambda_{appro}E_{appro} + \lambda_{appro}E_{ver}$. However, the target position cannot be arbitrarily selected. In (c), the target position is too far from the original position, thus the AAG web cannot reach the desired position.

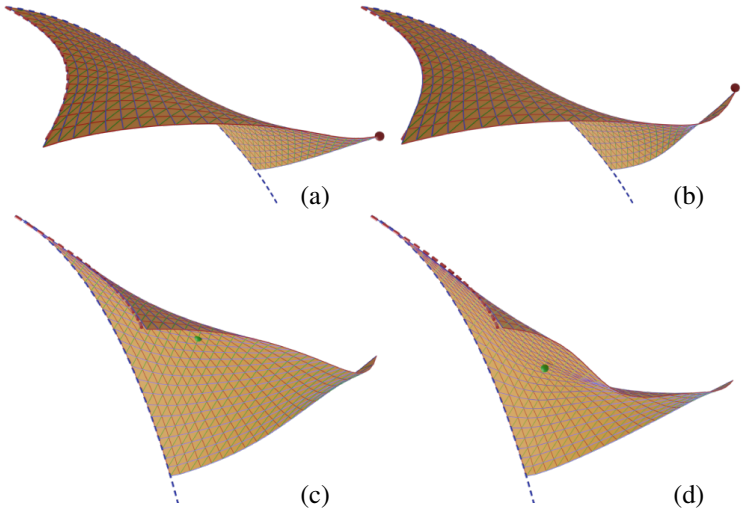



Figure 27: We also extend the interactive design of GGG to AGG webs, by optimizing $E = E_c + \lambda_{fair}E_{fair} + \lambda_{appro}E'_{appro} + \lambda_{appro}E_{ver} + \lambda_{guide}E_{guide}$. (a) is the same AGG web as in Fig. 9 (b). The selected point in (a) and the target position in (b) are marked as red dots. After obtaining the deformed AGG web (b), we continue to select and edit another point (green dots) of it (c)-(d).

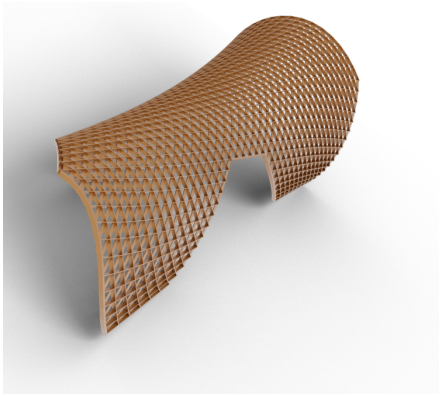


Figure 28: This gridshell is generated by trimming the boundary of the AAG web in Fig. 25 (c) along curves of that web.