

# Curved Crease Edge Rounding of Polyhedral Surfaces

Rupert Maleczek<sup>1\*</sup>, Klara Mundilova<sup>2\*</sup>, Tomohiro Tachi<sup>3\*</sup>

<sup>1</sup> i.sd | Structure and Design | Department of Design , University of Innsbruck, Austria

<sup>2</sup> Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, USA

<sup>3</sup> Graduate School of Arts and Sciences, The University of Tokyo, Japan

\* Corresponding authors e-mail: rupert.maleczek@uibk.ac.at, kmundil@mit.edu, tachi@idea.c.u-tokyo.ac.jp

## Abstract

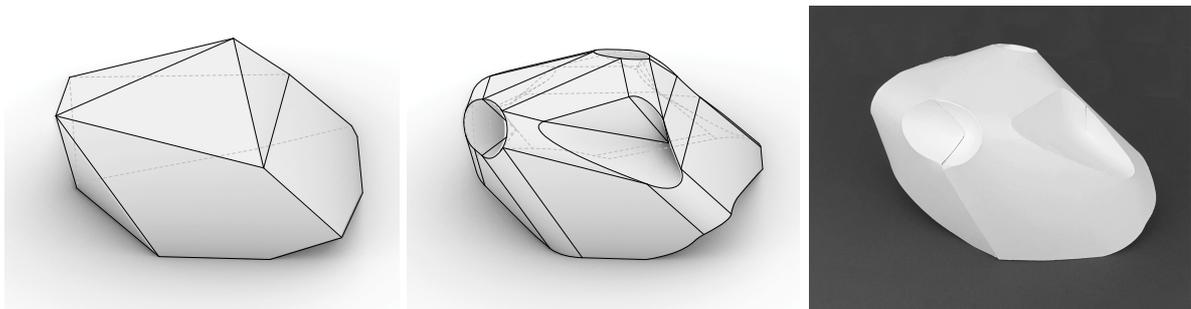
*We show a novel method to design a curved crease folding that constructs the edge-rounded, i.e., filleted, version of a given polyhedral surface. We replace each edge with a smoothly rounded cylinder and each vertex with a generalised cone, such that the surfaces joined through curved creases form a single developable surface with possible cuts at the singular cone apices. Because the curved crease can be explicitly computed from the isometry of corresponding line segments for given locations of the cone apex in 2D and 3D, our problem reduces to identifying the locations of the apices. We characterise the conditions for the apex positions and provide a numerical scheme to find the apices for the given mesh by solving a nonlinear optimisation problem. In general, the rounding of edges reduces the surface area, so the resulting curved folded surface is not isometric to the original polyhedron; in particular, the surface is not guaranteed to be foldable from a single piece of uncut paper if applied to a developable polyhedral surface. We solve this problem by computing consistent material loss caused by rounding radii.*

**Keywords:** Origami, Curved Folding, Developable Surface

## 1 Introduction

Folding along a prescribed pattern of curved creases on a thin sheet of material creates a 3D shape composed of elastically bent surfaces connected through curved hinges. The folded forms act as hybrids of bending active shells and folded plate structures and have potential applications in architectural design due to their benefits in fabrication (Maleczek et al. 2016, 2020; Koschitz 2019).

However, finding patterns that fold or approximate a given target shape is challenging due to the highly constrained geometry of the folds. As such, it is an active area of research. See Demaine et al. (2015) and Koschitz (2016) for reviews on curved crease designs. Note that there are different ways to prescribe target 3D shapes, which result in different shapes and structures: Chandra et al. (2015) use polyhedra as the target shape and obtain their approximation through smoothing, and Jiang et al. (2019) use multiple curved pleats to approximate given surfaces.



**Figure 1:** Illustration of the proposed design pipeline and the folded result.

In this paper, we show a method to design a curved crease folding from a given polyhedral surface, such that the folded shape represents the edge-rounded, i.e., filleted, version of the polyhedral surface (**fig. 1**). The input polyhedron can specify the overall shape of open and closed shells. Each vertex of input polyhedron will become a dimple surrounded by a closed curved crease, which works as a structural rib. Our method can be interpreted as a generalisation of some existing artworks of curved folding; e.g., Mosely (2002) computes modular origami from a rounded cube with circular dimples at the corners, and Mundilova (2019a) computes similar systems for regular polyhedra. Our method can be applied to non-regular (but not arbitrary) polyhedral surfaces. In addition, we characterise conditions for our construction to be applicable and conclude that not all polyhedra can be converted into such a curved folded design with our method.

The overview of our design approach consists in the following steps:

1. we first replace each edge of the polyhedron with a smoothly rounded cylinder and
2. we replace each vertex by a general cone meeting the incident cylinders and faces with curved and straight creases respectively, such that the developability is maintained at the creases. This further decomposes into two steps:
  - (a) For each vertex, we first locate the positions of the apex in 2D and 3D. A desirable location results in a collection of valid, non-self-intersecting patches whose tangent continuous crease curves are in a valid range.
  - (b) We then compute the curved crease explicitly from the isometry of corresponding line segments using the length constraint given by Mundilova (2019b).

In [sec. 2](#), we explain the construction process of [step 2b](#) to identify conditions for desirable positions of apices in 2D and 3D. Thus our problem reduces to identifying the locations of apices ([step 2a](#)) according to these constraints. The intersection of valid regions corresponding to the identified conditions may not exist. In particular, we show that we cannot apply our construction to saddle shaped vertices as characterised in [sec. 2.5](#). We numerically solve a non-linear constrained optimisation problem from the given conditions to locate positions of apices; the details of this process is described in [sec. 3](#).

In general, the rounding of edges reduces the surface area, so the resulting curved folded surface is not isometric to the original polyhedron. In particular, even if the original polyhedral surface is developable, the resulting curved crease surface is in general not closed around the cone apices. In [sec. 4](#), we show the constraints for the valid rounding radii used in [step 1](#) so that the resulting surface is also developable when applied to an originally developable surface. We solve this problem as a linear constrained optimisation problem for a given general polygonal mesh. In particular, we prove the existence of a consistent edge-rounding and cone apex construction for a limited class of polyhedral surfaces, namely, a conical convex mesh ([sec. 4.4](#)).

We implemented our construction method including the optimisation as a component for *Grasshopper*, so the designer can interactively design curved crease folding by modelling an initial mesh and adjusting other design parameters. [Section 5](#) shows how the parameters in each process affect the results. Furthermore, we show design variations using our methods that have potential architectural applications.

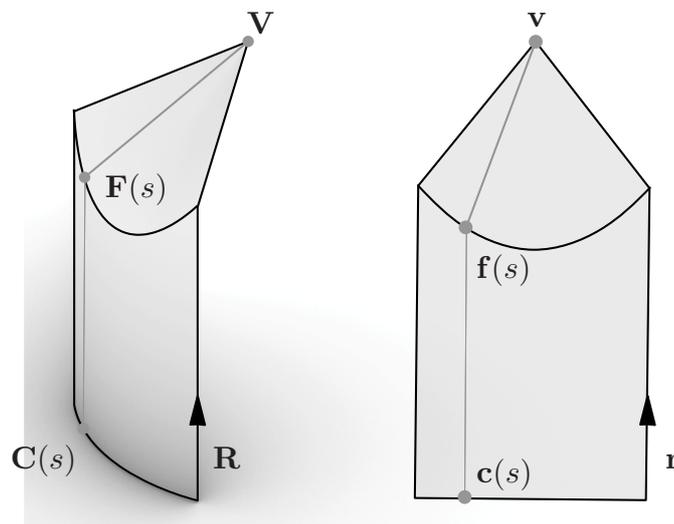
## 2 Finding Curved Creases between Cylinders and Cones

### 2.1 Computation

In our design, we fold right circular cylinders into cones with specified apices. The following computations are a special case of the formulas developed in Mundilova (2019b) to determine the crease curve between a given developable surface and cylinders or cones with given ruling direction or apex, respectively.

The main idea is that the crease curves can be computed only by “fitting” the lengths of the rulings of cylinders and cones. Let  $\mathbf{C}(s)$  with  $s \in [0, s_{\max}]$  denote the base curve of a cylinder, i.e., a planar curve that lies in a plane perpendicular to the ruling direction  $\mathbf{R}$ , see [fig. 2](#). Without loss of generality, we assume  $\mathbf{C}(s)$  to be an arc length parametrised curve in the  $xy$ -plane and  $\mathbf{R} = (0, 0, 1)$ . If we unroll this cylinder into the plane, let  $\mathbf{c}(s)$  denote the straight line corresponding to  $\mathbf{C}(s)$ , i.e.  $\mathbf{c}(s) = (s, 0)$ , and  $\mathbf{r}$  the ruling direction in the development,  $\mathbf{r} = (0, 1)$ . To keep the isometry between the cylinder and its development,  $\mathbf{c}(s)$  is perpendicular to  $\mathbf{r}$  and the parametrisation speeds of  $\mathbf{C}(s)$  and  $\mathbf{c}(s)$  need to match. The distance between a crease curve  $\mathbf{F}(s)$  and the curve  $\mathbf{C}(s)$  must be the same as the distance between its developed counterpart  $\mathbf{f}(s)$  and  $\mathbf{c}(s)$ , respectively. Therefore, we make the ansatz for the parametrisation.

$$\mathbf{F}(s) = \mathbf{C}(s) + l(s)\mathbf{R} \text{ and } \mathbf{f}(s) = \mathbf{c}(s) + l(s)\mathbf{r}.$$



**Figure 2:** A curved crease between a cylinder and a cone and its development.

Now, for given predefined positions of apices  $\mathbf{V}$  and  $\mathbf{v}$  in 3D and in the development, respectively, the length of every cone ruling, i.e. the distance from a point on the crease to the apex, must be equal in 3D and in the development, so

$$|\mathbf{F}(s) - \mathbf{V}|^2 = |\mathbf{f}(s) - \mathbf{v}|^2.$$

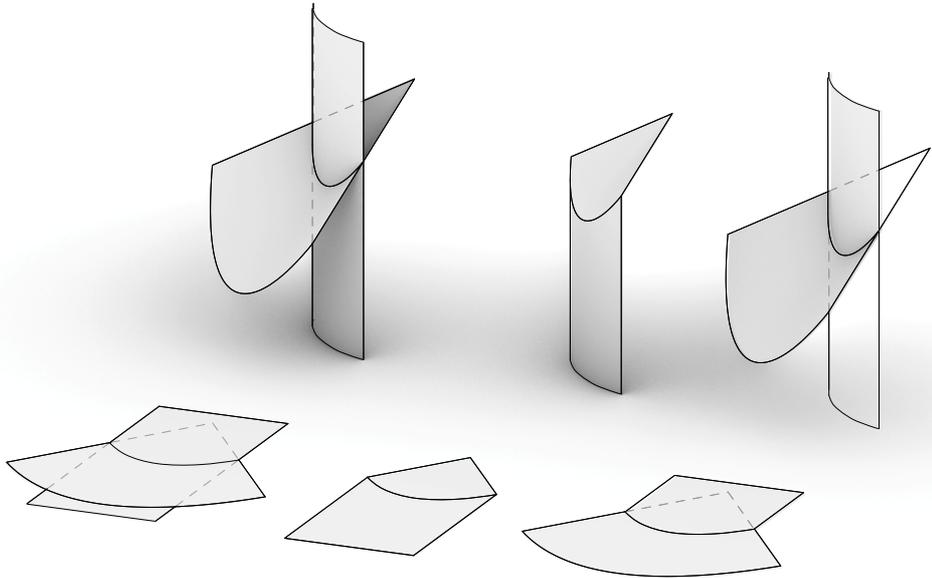
Thus we obtain

$$l(s) = \frac{1}{2} \frac{|\mathbf{v} - \mathbf{c}(s)|^2 - |\mathbf{V} - \mathbf{C}(s)|^2}{(\mathbf{v} - \mathbf{c}(s)) \cdot \mathbf{r} - (\mathbf{V} - \mathbf{C}(s)) \cdot \mathbf{R}}. \quad (1)$$

For  $\mathbf{V} = (V_x, V_y, V_z)$  and  $\mathbf{v} = (v_x, v_y)$ , this simplifies to

$$l(s) = \frac{1}{2} \frac{|\mathbf{v} - \mathbf{c}(s)|^2 - |\mathbf{V} - \mathbf{C}(s)|^2}{v_y - V_z}.$$

The length is computed if  $v_y - V_z \neq 0$  and locates the crease curve on the surfaces and in the development. However, there are four possible ways to combine the cylinder and the cone patches, and only two of them create a developable surface (see [fig. 3](#)).



**Figure 3:** The two developable combinations of a curved crease between a cylinder and a cone.

Therefore, we need to carefully choose the intended developable combination. In our construction we want to choose

- the cone patch between the crease curve and the apex
- the cylinder patch from the crease curve in  $-\mathbf{R}$  direction,

where  $\mathbf{R}$  is oriented toward the incident vertex of interest.

In the following subsection, we will argue when this combination is “valid”. Valid combinations of surface patches are characterised by the contraction property: the Euclidean distance in its folded state in 3D is shorter than or equal to the geodesic distance along the folded surface, or the Euclidean distance in its development, where the equality is satisfied when there is no crease between the points. In addition, we identify conditions that restrict the crease curve to a range along each cylinder, i.e.,  $0 \leq l(s) \leq l_{\max}$ .

## 2.2 Valid surface patch combinations and valid range

The numerator and the denominator of [eq. \(1\)](#) have both geometric interpretations that are linked to the valid surface patch combination and valid length. Note that the valid patch condition is a necessary condition for the folding to exist (if not satisfied, there is no way to put creases in the correct direction), while the valid range condition is relative to the position of the base curve defined (if not satisfied, the computed length will be negative and thus the crease go over the base curve). The latter will be particularly important for the implementation.

### Valid surface patch combination

The denominator compares the heights of the apices measured from the base curve in its development and in the folded state. This is equivalent to the difference in distances between a point along the ruling and the apex, when the point approaches infinity in  $-\mathbf{R}$  direction. For the points on the valid patch, every intermediate difference is positive due to the contraction property. Therefore, this expression is also positive in the limit.

**Theorem.** *A valid patch combination is characterised by the sign of  $D := v_y - V_z$  and undefined for  $D = 0$ . In particular, a valid combination with the conical part containing the apex is the cylindrical part in  $-\text{sign}(D)\mathbf{R}$  direction.*

*Proof.* Consider the overlay of a cross section through the plane spanned by  $\mathbf{V}$ ,  $\mathbf{C}(s)$  and  $\mathbf{R}$  with the development such that  $\mathbf{c}(s)$  and  $\mathbf{C}(s)$  coincide with the origin, and the ruling directions  $\mathbf{r}$  and  $\mathbf{R}$  are aligned, see [fig. 4](#). We parametrise the points on the ruling through  $\mathbf{c}(s)$  and  $\mathbf{C}(s)$  by  $\mathbf{b}(t) = t\mathbf{r}$  and  $\mathbf{B}(t) = t\mathbf{R}$ . Then

$$D_t = |\mathbf{v} - \mathbf{b}(t)|^2 - |\mathbf{V} - \mathbf{B}(t)|^2 = |\mathbf{v}|^2 - |\mathbf{V}|^2 - 2t(\mathbf{v} \cdot \mathbf{r} - \mathbf{V} \cdot \mathbf{R}) = |\mathbf{v}|^2 - |\mathbf{V}|^2 - 2tD.$$

Note that  $D_t$  indicates whether the distances between points on the ruling and the apex are shortening. As  $D_t$  is linear in  $t$  and  $D \neq 0$ , it changes sign at  $t_0 = \frac{|\mathbf{v}|^2 - |\mathbf{V}|^2}{2D}$ . We distinguish between two cases:

- If  $D > 0$ , then  $D_t > 0$  for all  $t < t_0$ . The contraction happens for points below the crease curve, i.e. the valid cylindrical patch has ruling direction  $-\mathbf{R}$ .
- If  $D < 0$ , then  $D_t > 0$  for all  $t > t_0$ . The contraction happens for points above the crease curve, i.e., the valid cylindrical patch has ruling direction  $+\mathbf{R}$ .

□

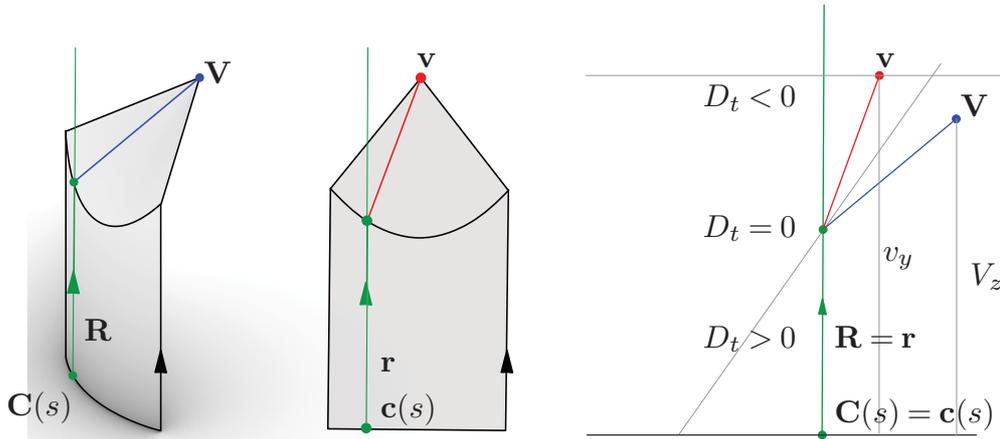


Figure 4: Figure illustrating proof of theorem .

### Valid range

As described in [sec. 3](#), we set the base curve at the midpoint of the rounded edge, so  $l(s) \geq 0$  for all  $s \in [0, s_{\max}]$  ensures that the creases from two incident vertices of a rounded crease will not affect each other. Furthermore, we upper bound the length of the boundary rulings to not exceed the intersection of cylinders around a vertex.

For valid surface patch combinations, the denominator is positive and thus the sign of the length function  $l(s)$  is determined by the numerator. The numerator is positive, if and only if the distances between the base curve and apex contract after folding, i.e.,  $l(s) \leq 0$  if

$$|\mathbf{v} - \mathbf{c}(s)|^2 - |\mathbf{V} - \mathbf{C}(s)|^2 \geq 0.$$

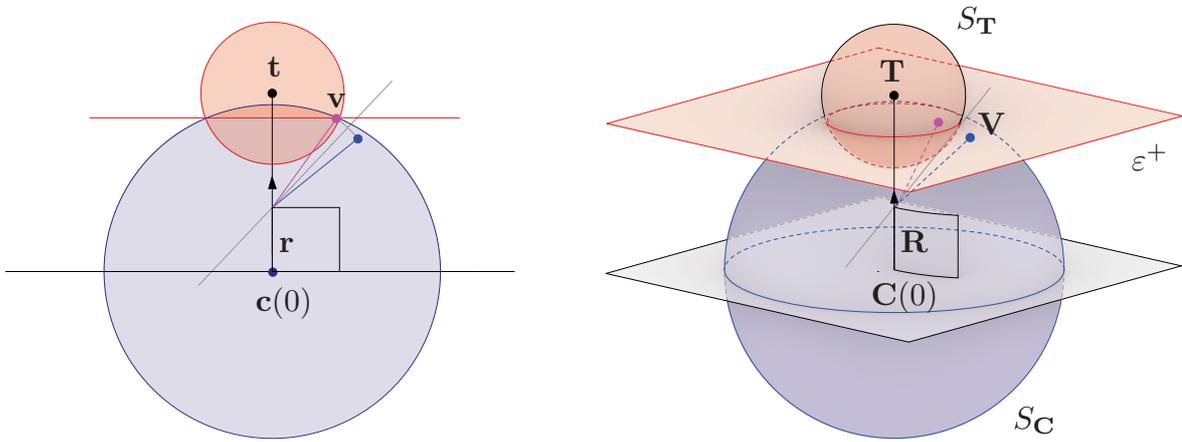
We can rewrite the upper bound  $l(s) \leq l_{\max}$  as

$$|\mathbf{v} - \mathbf{t}|^2 - |\mathbf{V} - \mathbf{T}|^2 \leq 0$$

where  $\mathbf{t} = \mathbf{c}(s) + (0, l_{\max})$  and  $\mathbf{T} = \mathbf{C}(s) + (0, 0, l_{\max})$ . Therefore, the distances between the apices and the upper bounds need to expand during folding to prevent the crease from escaping the intended range.

### Geometric interpretation

For a given developed apex  $\mathbf{v}$ , let  $S_{\mathbf{C}}$  be the sphere with center  $\mathbf{C}(s)$  radius  $|\mathbf{v} - \mathbf{c}(s)|$  and  $S_{\mathbf{T}}$  the sphere with center  $\mathbf{T}$  radius  $|\mathbf{v} - \mathbf{t}(s)|$ , see [fig. 5](#). We indicate the interior of a sphere  $S$  by  $S^+$ .



**Figure 5:** Given a developed apex, the desirable region for the spatial apex for  $0 \leq l(s)$  is  $S_{\mathbf{C}}^+ \cap \varepsilon^+$  and for  $0 \leq l(s) \leq |\mathbf{c}(0) - \mathbf{t}|$  it is  $S_{\mathbf{C}}^+ \setminus S_{\mathbf{T}}^+$ .

An apex  $\mathbf{V}$  satisfies the valid patch condition, if it lies in the half space defined by  $\varepsilon^+ : z < v_y$ . Moreover, an apex  $\mathbf{V}$  satisfies the valid range condition, if it lies in the region  $S_{\mathbf{C}}^+$  or  $S_{\mathbf{C}}^+ \setminus S_{\mathbf{T}}^+$ , respectively.

Therefore, for a proper position of the base curve, the intersection of conditions can be represented by  $\mathbf{V} \in S_{\mathbf{C}}^+ \cap \varepsilon^+$  or  $\mathbf{V} \in S_{\mathbf{C}}^+ \setminus S_{\mathbf{T}}^+ \cap \varepsilon^+ = S_{\mathbf{C}}^+ \setminus S_{\mathbf{T}}^+$ .

The intersection of all valid spaces for good candidates along the points of a base curve lies in the interior of  $\varepsilon^+$ , which we include as a constraint for the later described optimisation. The intersection of the spherical location constraints along the base curve is approximated by taking the intersection of valid regions corresponding to the boundary and at center points of the curve.

### 2.3 Preventing self-intersection

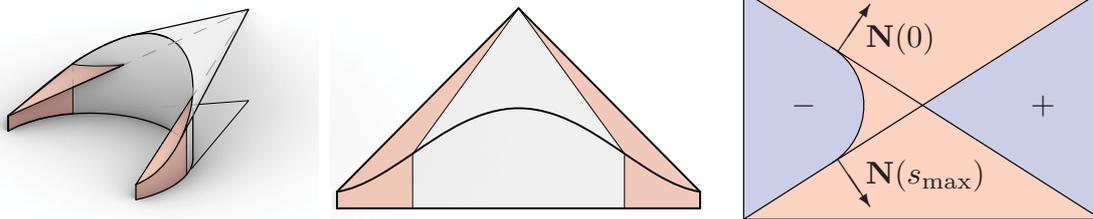
In addition to the valid patch and valid range constraints, an improper choice of the apex  $\mathbf{V}$  can cause intersections between the cylinder and cone. Consider, for example, the construction of a crease curve on a cylinder, whose base curve is a half of a circle, for an apex lying outside the cylinder as in [fig. 6](#). Directly applying the above-mentioned geometric construction yields a self-intersection of the paper when the ruling emanating from the apex goes past the point of tangency as the crease curve wraps around the cylinder. At the points of tangency, the

rulings emanating from the apex transition from the convex to the concave side of the cylinder, and thus the crease assignment changes between mountain and valley while passing through complete  $180^\circ$  folding. Intrinsically, this can be also observed as the crease curve having an inflection point at the transition point. This is because the convexity or concavity of the crease curve and its position w.r.t. the current cylinder patch side corresponds to the sign of fold angles of the crease (see Demaine et al. 2015).

Locating such a transition point is equivalent to locating the shade line of the cylinder when we put a point light source at the cone apex. Therefore the target cylinder region needs to be constantly lit or shaded with respect to the light source. Denoting the surface normals of the cylinders along  $\mathbf{C}(s)$  by  $\mathbf{N}(s)$ , the *non-intersection condition* reads

$$\begin{aligned} (\mathbf{V} - \mathbf{C}(s)) \cdot \mathbf{N}(s) &> 0 \text{ for all } s \in [0, s_{\max}] \text{ or,} \\ (\mathbf{V} - \mathbf{C}(s)) \cdot \mathbf{N}(s) &< 0 \text{ for all } s \in [0, s_{\max}] \end{aligned} \tag{2}$$

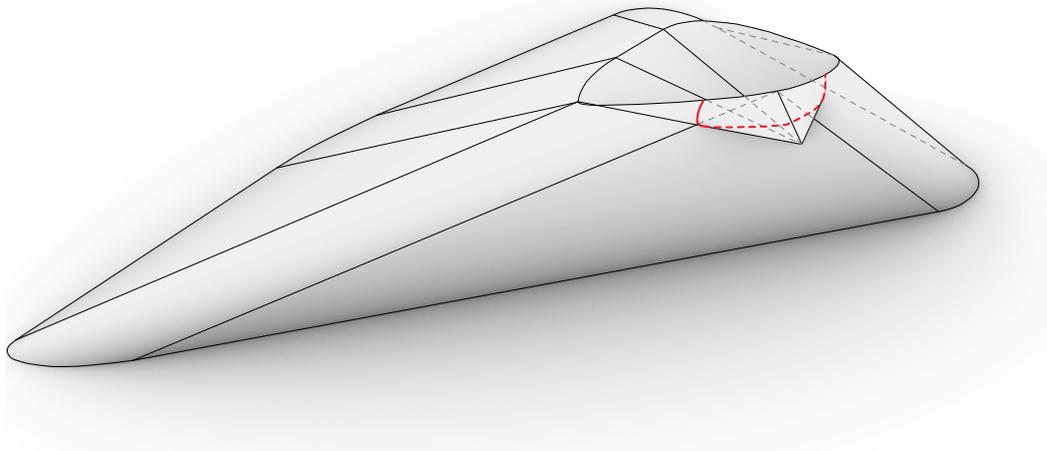
For each rounded crease, this determines the feasible regions for the candidate apex as in **fig. 6** right (each component corresponds to plus or minus sign).



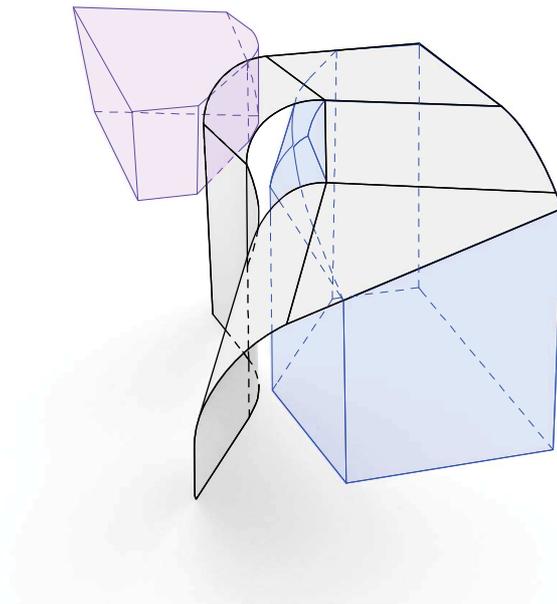
**Figure 6:** (a) Curved crease with self-intersections. (b) The development of the self-intersecting crease. (c) Top view of admissible and forbidden regions for the location of the 3D apex (blue and red, respectively) for given cylindrical patch.

In the regions of admissible points, the non-intersection condition are obtained by taking the intersection of admissible regions for all incident rounded edges of an vertex. This is equivalent to taking the apex consistently on the front normal side or back normal side measured from any point on the surface. **Figure 7** shows an invalid apex position, which causes intersection at either of incident rounded edges. The existence of such a region depends on the types of vertices as we illustrate in **sec. 2.5**. We potentially get two portions of the solution space on the outside and the inside of the vertex, corresponding to whether a light source constantly illuminates and shades the surface, respectively (see **fig. 8**). However, when combined with the valid patch condition, at most one of these components is feasible as explained in **sec. 2.5**.

Note, that for the above observation, we did not need  $C(s)$  to be circular, only convex or concave.



**Figure 7:** An apex with self intersection. Note that the intersection happens at the rounded creases.

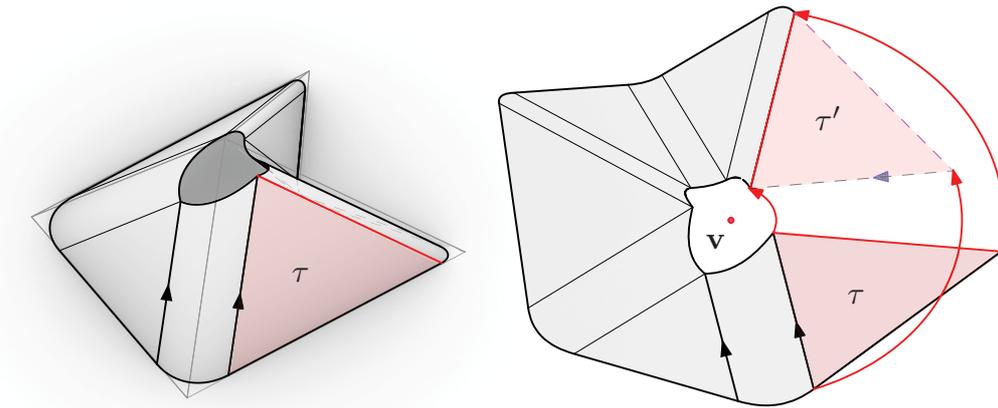


**Figure 8:** Illustration of the two admissible regions of a vertex.

## 2.4 Tangent Continuity

In our design, we are rounding edges of polyhedral surfaces with tangent continuous cylinders. Therefore, we are ultimately interested in crease curves between tangent continuous combinations of cylinders, planes and cones or triangles, respectively. To prevent possibly undesirable kinks in the crease curve, we require the crease curve to be tangent continuous in the transitions between every cylinder and plane.

Suppose we have rounded the edges adjacent to a vertex, developed them successively into the plane and fixed an apex in space and development, see [fig. 9](#). When we transition from a cylinder to an adjacent plane  $\tau$ , this plane is the tangent plane of the cylinder at the transitioning ruling by construction. Furthermore, the tangent of the crease curve is the intersection of  $\tau$  and the tangent plane of the adjacent cone. As the development of the cylinder and cone preserves the angles between the tangent and the rulings, we see that the tangent is the unique extension of the crease curve from the cylinder to the plane.



**Figure 9:** Vertex with open development whose 2D apex is determined as the center of rotation transforming  $\tau \mapsto \tau'$ .

As we progress on the cylinders and planes around a vertex, we obtain tangent continuous crease curves. If the development closes up, i.e., the rounded surface is developable, the crease curve closes up for free due to the imposed length constraints. However, if the development has a gap due to an angular defect, we need to impose further constraints to guarantee a closed and tangent continuous curve.

Suppose we started the unrolling with a planar patch  $\tau$  and there is a gap between the last cylinder and  $\tau$ . Let  $\tau'$  be the copy of the first plane oriented to be attached to the last cylinder. For consistent tangent continuity, we ensure that any point along the crease embedded on  $\tau$  and the corresponding crease on  $\tau'$  have the same distance to the developed apex. Therefore, this developed apex needs to be the center of the rotation transforming  $\tau$  into  $\tau'$  in plane, uniquely determining the developed apex.

Note that the above construction can be generalised to other tangent continuous and in the development open surfaces when folded tangent continuously into a cone.

## 2.5 Feasible apex directions

We characterise the feasible regions for  $\mathbf{V}$  given by the intersection of constraints for each vertex, namely the valid patch and the non-intersection conditions. We omit the valid range condition because we can move the base curve sufficiently far away from the apex, as we consider each vertex separately. In addition, we focus on a simplified necessary condition for those conditions under an assumption described below, which is also sufficient in the limit case with rounding radius approaching to 0.

Let  $\mathbf{U}$  denote the original vertex position and  $\mathbf{R}_i$  the direction of each incident edge  $i$ , oriented towards the current vertex, with its development  $\mathbf{r}_i$ . Similarly, we use a subscript to identify the base curve  $\mathbf{C}_i(s_i)$  and  $\mathbf{c}_i(s_i)$  for each edge. Here, we assume that the height of the developed apex position is smaller than or equal the height of the original vertex position, i.e.,  $(\mathbf{v}_i - \mathbf{c}_i(0)) \cdot \mathbf{r}_i \leq (\mathbf{U} - \mathbf{C}_i(0)) \cdot \mathbf{R}_i$ . We conjecture that the assumption is always true for our construction for non-developable vertices with consistent material loss due to the uniqueness of the developed apex positions derived from the tangent continuity constraint (sec. 2.4) together with the material loss (sec. 4).

The intersection of the valid patch conditions  $(\mathbf{v} - \mathbf{c}_i(0)) \cdot \mathbf{r}_i - (\mathbf{V} - \mathbf{C}_i(0)) \cdot \mathbf{R}_i > 0$  is contained in  $(\mathbf{U} - \mathbf{C}_i(0)) \cdot \mathbf{R}_i - (\mathbf{V} - \mathbf{C}_i(0)) \cdot \mathbf{R}_i > 0$ . Using  $\mathbf{D} := \mathbf{V} - \mathbf{U}$ , the condition is represented as

$$\mathbf{D} \cdot (-\mathbf{R}_i) > 0. \quad (3)$$

Solving eq. (3) is equivalent to finding a plane of normal  $\mathbf{D}$  passing through  $\mathbf{U}$  such that all adjacent faces lie on one side of this plane. An important consequence is that the construction does not work for a *saddle* vertex, i.e., a vertex without such a plane. Note that while the sum of incident face angles of a saddle vertex needs to be more than  $2\pi$  and thus has negative integral curvature, not all negative vertices are saddles. For example, consider a negative vertex corrugated to approximate a convex vertex. In addition, since a developable vertex, i.e., a vertex whose sum of sector angles is  $2\pi$ , folds to a half plane either by popping up or down as shown in Abel et al. (2016), it is always possible to find a direction  $\mathbf{D}$  satisfying eq. (3).

In addition, we sample the non-intersection conditions given by eq. (2) at the extrema  $s = 0$  for each edge, where the surface normal equals the face normal  $\mathbf{N}_{i-1,i}$  between consecutive edges  $i-1$  and  $i$ . Because the extremal point  $\mathbf{C}_i(0)$

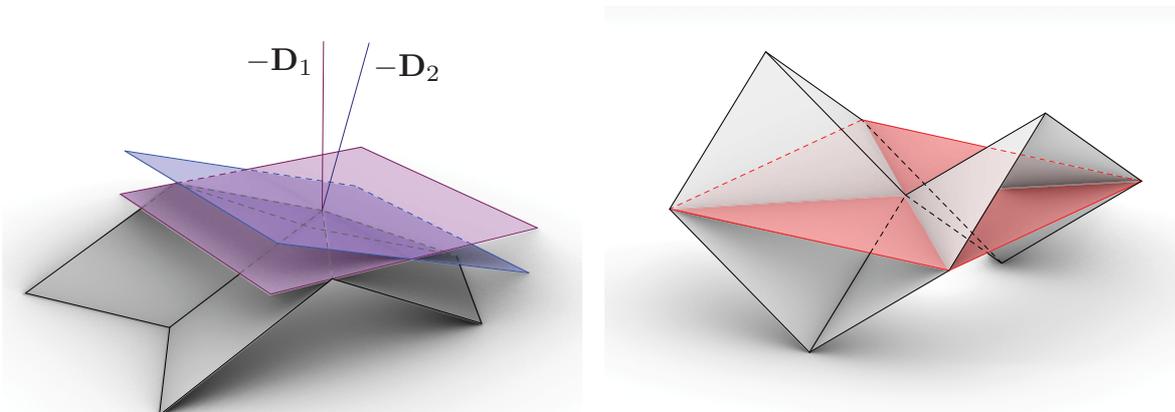
and  $\mathbf{U}$  are both on face  $i - 1, i$ , [eq. \(2\)](#) yields

$$\mathbf{D} \cdot \sigma \mathbf{N}_{i-1,i} > 0 \text{ for all } i - 1, i, \tag{4}$$

for either  $\sigma = \pm 1$ .

Note that at most one of the signs  $\sigma$  can be satisfied. The half-plane discussion given from [eq. \(3\)](#) predicts the correct candidate of  $\sigma$  as follows. Consider the solid angle  $\Omega$  of the vertex on the front side defined by  $\mathbf{N}_i$ , then the back side has solid angle of  $4\pi - \Omega$ . Because of the half-plane property, the solid angle of the vertex on  $\mathbf{D}$  side needs to be smaller than  $2\pi$ . So, we choose  $\sigma = 1$  if  $\Omega < 2\pi$ , and  $\sigma = -1$  if  $\Omega > 2\pi$ .

We call the direction  $\mathbf{D}$  computed from both [eq. \(3\)](#) and [eq. \(4\)](#) the *vertex normal* ([fig. 10](#)). In [sec. 3.1](#), we start our computation by constructing the normal from given edge directions and face normals. Once a valid normal is found, the vector can be arbitrarily scaled to find other candidates for apex positions. Because the effect of rounding becomes relatively small by scaling up the vector, there is a sufficiently distant position of apex along the normal that creates a valid folding.



**Figure 10:** (a) Conical vertex with two vertex normal candidates  $\mathbf{D}_1$  and  $\mathbf{D}_2$ . Note that  $\mathbf{D}_1$  computed as the average of face normals do not satisfy [eq. \(3\)](#), while  $\mathbf{D}_2$  is a valid vertex normal, the normal plane to which bounds a half space containing the adjacent faces of the vertex. (b) Saddle shaped vertex.

### 3 Finding Apices

For every interior vertex of the mesh, we solve an optimisation problem to find the location of

- the apex  $\mathbf{V}$ , if the development is open or
- the apices  $\mathbf{V}$  and  $\mathbf{v}$ , if the development is closed,

that fold the incident cylinders and planes along tangent continuous curves into

cones and triangles. We implement the constraints collected in [sec. 2](#) to avoid invalid surface patch combinations and intersections.

### 3.1 Vertex normals

Before we start the optimisation, we determine a normal direction  $\mathbf{D}$  of the vertex as explained in [sec. 2.5](#) through optimisation; we later locate the initial apex on the computed normal. We solve [eq. \(3\)](#) and [eq. \(4\)](#) by maximizing the minimum dot product of a unit vector with the adjacent edge directions  $-\mathbf{R}_i$  and face normal directions  $\sigma\mathbf{N}_i$ .

**Variables:** The variables are the three coordinates of the vertex normal and a scalar  $b$  which is going to be the lower bound that we maximise.

**Equality constraint:** To prevent unboundedness of variables, we keep  $\mathbf{D} \cdot \mathbf{D} = 1$ .

**Inequality constraint:** We bound  $\mathbf{D} \cdot \mathbf{R}_i \geq b$  and  $\mathbf{D} \cdot (\sigma\mathbf{N}_i) \geq b$  for every adjacent edge and face.

**Objective:** We maximise the lower bound  $b$ .

**Initialisation:** As an initial guess, we take the normalised mean over all edge directions and oriented face normals.

### 3.2 Optimisation set-up

To locate the apices, we solve a non-linearly constrained optimisation problem. We use hard inequality constraints given from [sec. 3](#) and minimise the objective function to achieve regularity. Specifically, we sample the quadratic inequality constraints resulting from the valid range condition, by evaluating the base curve at sampled points, e.g., at the beginning, middle and end of a cylindrical arc. For every incoming edge, we let  $\mathbf{C}(s_0)$ ,  $\mathbf{C}(s_1)$  and  $\mathbf{C}(s_2)$  denote the points on the central circle of the cylinder corresponding to the start, middle and end parameters, and  $\mathbf{c}(s_0)$ ,  $\mathbf{c}(s_1)$  and  $\mathbf{c}(s_2)$  denote their developed locations. Let furthermore  $\mathbf{T}_i$  with  $i \in \{0, 2\}$  be the intersection at the start and end parameter of the cylinder with the neighbouring cylinder and  $\mathbf{R}_i$  the ruling direction towards the vertex. The corresponding developments are again indicated by lower cases.

**Variables:** The variables are the three coordinates of  $\mathbf{V}$  and the two coordinates of  $\mathbf{v}$  if the development is closed.

**Inequality constraints:** For every adjacent edge, we add five quadratic and two linear inequality constraints, namely:

- We require that the developed height is larger than the spatial height, i.e.,

$$(\mathbf{V} - \mathbf{C}(s_0)) \cdot \mathbf{R} \leq (\mathbf{v} - \mathbf{c}(s_0)) \cdot \mathbf{r}.$$

- We require that the current combination for  $(\mathbf{V}, \mathbf{v})$  lies in the respective spheres  $S^+(s_i)$  at the three parameter values, i.e.,

$$|\mathbf{V} - \mathbf{C}(s_i)|^2 \leq |\mathbf{v} - \mathbf{c}(s_i)|^2 \quad \text{for } i \in \{0, 1, 2\}.$$

- We require that the lengths do not exceed the distance between the base point to the intersection of cylinders, i.e.,  $l(s_i) \leq |\mathbf{T}_i - \mathbf{C}(s_i)|$  for  $i \in \{0, 2\}$ , or equivalently,

$$|\mathbf{v} - \mathbf{t}_i|^2 \leq |\mathbf{V} - \mathbf{T}_i|^2.$$

- We linearise the rounding depicted in [fig. 6](#) by a plane through  $\mathbf{C}(s_0)$ ,  $\mathbf{C}(s_2)$  and  $\mathbf{T}_0$  or  $\mathbf{T}_2$ . We believe that this is not too constraining as candidates close to the cylinders are not very desirable. This yields

$$(\mathbf{V} - \mathbf{T}_0) \cdot \sigma \mathbf{N}_e > 0 \quad \text{where} \quad \mathbf{N}_e = \frac{(\mathbf{C}(s_2) - \mathbf{C}(s_0)) \times (\mathbf{T}_0 - \mathbf{C}(s_0))}{|(\mathbf{C}(s_2) - \mathbf{C}(s_0)) \times (\mathbf{T}_0 - \mathbf{C}(s_0))|}.$$

Furthermore, for every adjacent face, we want the vertex to lie on the correct side of the faces to ensure non-intersections as described in [sec. 2.3](#). This amounts in

$$(\mathbf{V} - \mathbf{T}_i) \cdot \mathbf{N} > 0,$$

where  $\mathbf{T}_i$  is the intersection of cylinders in the current face.

**Initialisation:** We initialise the spatial apex  $\mathbf{V}$  with a user specified point  $\mathbf{P} = \mathbf{V} + d\mathbf{D}$  on the vertex normal, where  $d > 0$  is the *depth* along the vertex normal. If the development is closed, we initialise  $\mathbf{v}$  as the intersection  $\mathbf{p}$  of the central rulings of two consecutive developed cylinders.

**Objective:** We use the objective for regularisation. Depending whether the development is open or closed, we minimise

$$\min_{\mathbf{V}} (\mathbf{V} - \mathbf{P})^2 \quad \text{or} \quad \min_{(\mathbf{V}, \mathbf{v})} (\mathbf{V} - \mathbf{P})^2 + (\mathbf{v} - \mathbf{p})^2.$$

This objective function tries to keep the original depth  $d$ ; [fig. 11](#) shows the results using different depths.

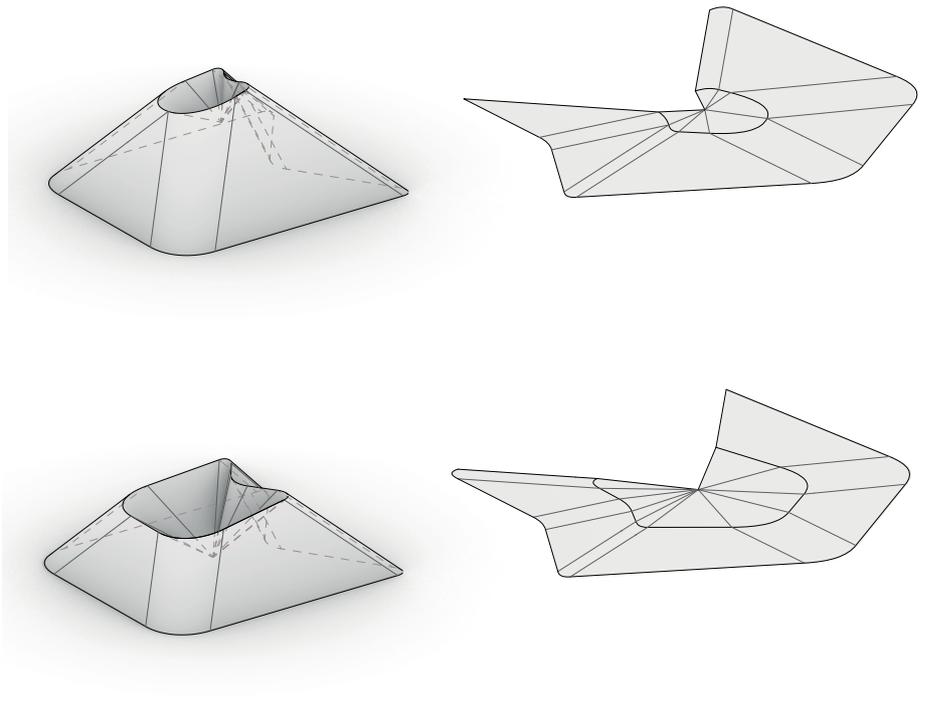


Figure 11: Results for two different depths.

## 4 Consistent Material Loss

By edge rounding, the intrinsic distance between adjacent faces gets closer. So the target polyhedral surface may be understood as the folding of the “shrunk” version of the polyhedral surface. However, it is not always possible to define such a “shrunk” paper if the resulting material loss for each rounded edge around a vertex is not consistent. An arbitrary material loss around the vertex ruins the consistent disk topology of the original sheet of paper, leaving a non-fillable hole around the vertex.

In this section, we seek out the consistent material loss that allows the “shrunk” polyhedral paper to fold into the rounded and curved-creased target shape without cutting. In particular, when starting from a polyhedral surface with developable vertices, e.g., an origami tessellation, this consistency ensures that the curved crease version can still be made from a single sheet of paper without any cuts.

### 4.1 Material loss

First, we compute the material loss when rounding an edge with a right circular cylinder (fig. 12). Let us denote the half angle between the adjacent surface normals by  $\gamma$  and the radius of rounding by  $r$ . Then the length  $a$  of the rounded edge and the original length  $A$  are

$$a = 2r\gamma \quad \text{and} \quad A = 2r \tan \gamma.$$

We call  $w = A - a$  the *material loss* induced by rounding an edge. In the following process, we first compute a consistent amount of material loss, which conversely, determines the radius of rounding. For given  $w$  and  $\gamma$ , the remaining quantities read

$$a = \frac{w\gamma}{\tan\gamma - \gamma}, \quad A = \frac{w \tan\gamma}{\tan\gamma - \gamma}, \quad r = \frac{w}{2(\tan\gamma - \gamma)}.$$

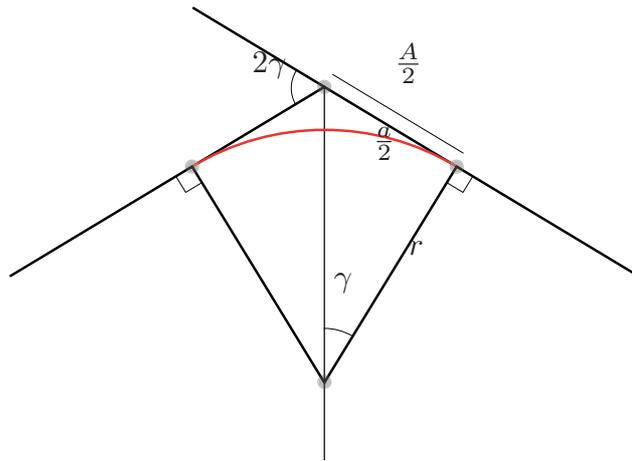


Figure 12: Illustration of the material loss induced by rounding an edge.

## 4.2 Consistency Condition

The main idea is as follows. Consider that if we walk on the surface around a vertex in the direction perpendicular to the adjacent edges by the amount of material loss, we need to end up at the same point we started from (fig. 13). This forms a closed polygonal cycle around each vertex. For each polyhedral edge, we can draw a rectangular region (coloured green in fig. 13) between the corresponding congruent edge of the cycles on both ends. The material loss is equivalent to contracting these cycles to points and rectangles to segments, where points and segments correspond to vertices and edges of polyhedral sheet of paper that is then folded to a curved crease folding.

If the original polyhedron is developable, the construction of the polygonal cycle perpendicular to the polyhedral edge is equivalent to drawing a reciprocal diagram on polyhedral the development (see fig. 14). In particular, the edge graph needs to be a *spiderweb*, i.e., there exists a positive edge-length reciprocal diagram. Our construction creates such a reciprocal diagram intrinsically on a polyhedral surface, so the method can apply to any polyhedral surfaces.

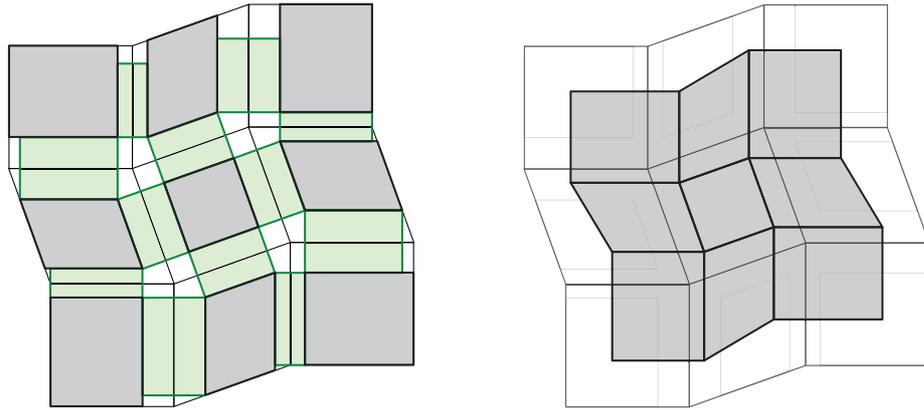


Figure 13: Developable tessellation with contracting cycles and regions before and after contraction.

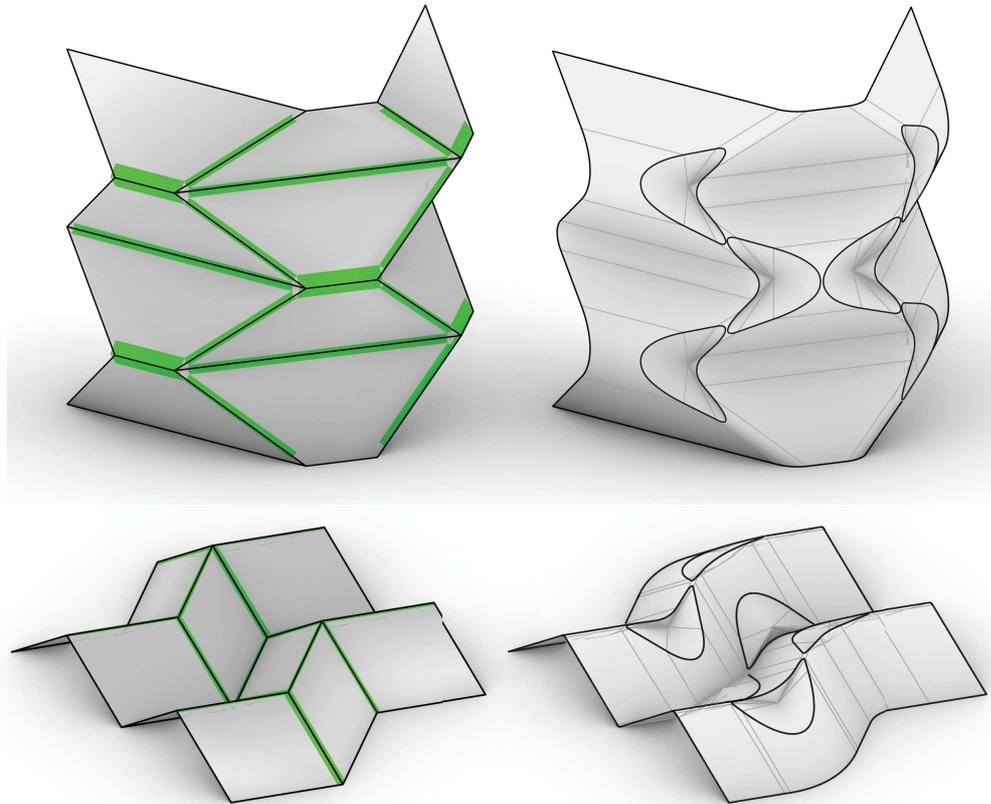


Figure 14: Computed material loss on the edges of a chicken wire and Miura-ori tessellation and the resulting rounding.

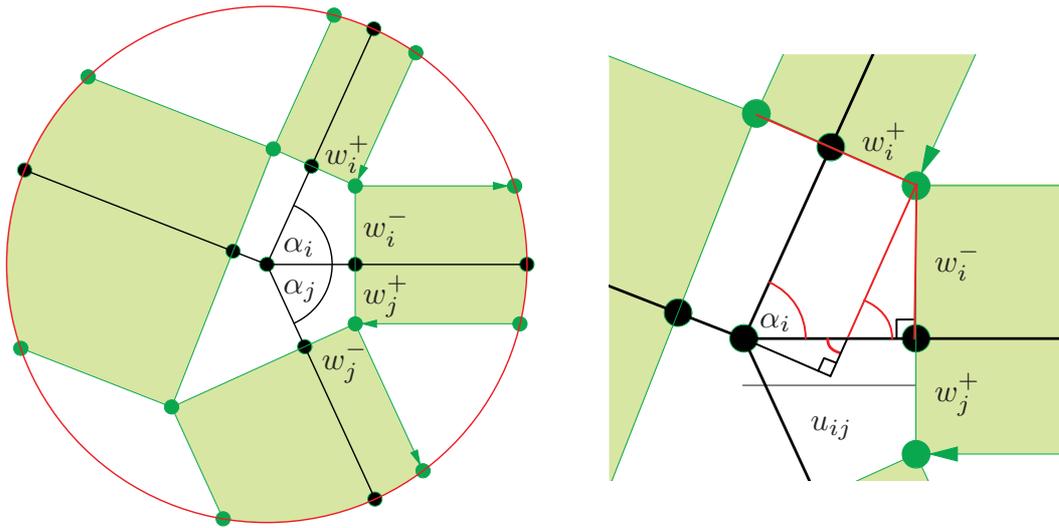
### 4.3 Optimisation set-up

The consistent material loss is given as the following linear programming problem.

**Variables:** We locate the contracting regions by identifying the cycle around the vertex. The corner of the cycle on sector  $i$  can be represented by the distances  $w_i^-$  and  $w_i^+$  from incident edges of the sector.

For every interior edge between sectors  $i$  and  $j$ , the two variables  $w_i^-$  and  $w_j^+$  correspond to the material loss on two sides of edges, summing up to the material loss of the edge  $w = w_i^- + w_j^+$ . Here, we used the notation based on half-edge data structure: for a sector angle  $\alpha_i$  of a vertex, we let  $w_i^+$  denote the variable associated with the incoming adjacent oriented edge and  $w_i^-$  the variable associated with the outgoing adjacent edge, see **fig. 15**.

Furthermore, we introduce a variable  $b$  for the lower bound that we are going to maximise.



**Figure 15:** Preliminary notation for constraints along vertex.

**Linear equality constraints:** Along every vertex, the perpendicular widths should close up. For every adjacent face, we set the widths  $u_{ij}$  along the quad to be the same, see **fig. 15**. This results in the linear equality constraint

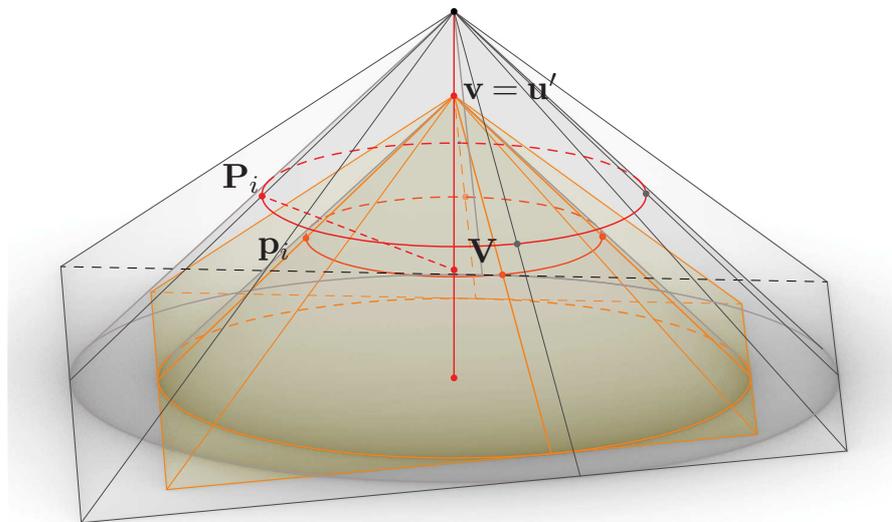
$$\left(w_i^+ + w_i^- \cos \alpha_i\right) \frac{1}{\sin \alpha_i} = \left(w_j^- + w_j^+ \cos \alpha_j\right) \frac{1}{\sin \alpha_j}.$$

**Linear inequality constraints:** Before we start the optimisation problem, we compute the maximal rounding width for every edge and upper bound the sum of the widths of two opposite half edges by this maximal rounding loss. Furthermore, we require the widths  $w_i > b$ .

**Objective function:** We maximise  $b$  subject to the above constraints. The solution is feasible, if  $b > 0$ .

## 4.4 Conical Convex Mesh

In the special case of conical convex meshes, we can construct the consistent material loss along vertices and location of apices, see [fig. 16](#). Conical meshes are characterised by having faces tangent to a cone of revolution whose axis is the intersection of the bisectors of two neighbouring faces. This in particular implies, that conical meshes have a family of constant face offsets whose vertices lie on the axes of the tangent cones. We use the offset mesh to construct the consistent material loss and the axis as the vertex normal to locate the apex.



**Figure 16:** Illustration of the construction of consistent material loss and apex location.

Suppose we fix an offset distance and project the faces perpendicular to the original faces. This locates the contracting regions, from which we compute the rounding applied to the original mesh, so the offset mesh is the piece of paper we fold.

For this material loss, we find good candidates for apices on the axis. By tangent continuity condition, the developed apex corresponds to the vertex of the offset mesh. If we intersect the common lines of the cone and the offset faces with a plane perpendicular to its axis, we find that the intersection points  $\mathbf{p}_i$  lie on a circle of radius  $r$  with center on the axis. If this plane is sufficiently far away from the apex, we can find a point on the axis whose distance to the corresponding points  $\mathbf{P}_i$  on the original surface is again  $r$ .

## 5 Design Examples

We implemented our construction method as a component for *Grasshopper* to create an interactive design system. The user specifies the input mesh, the preferred

rounding radii, preferred depth of apex, which are used as the initial conditions for the optimisation. We implemented the component in C# using *ALGLIB* (Bochkanov 2020) for optimisation using Augmented Lagrangian (AUL). Furthermore, we used the half edge data structure of *Plankton* (Piker and Pearson 2013) for mesh operations.

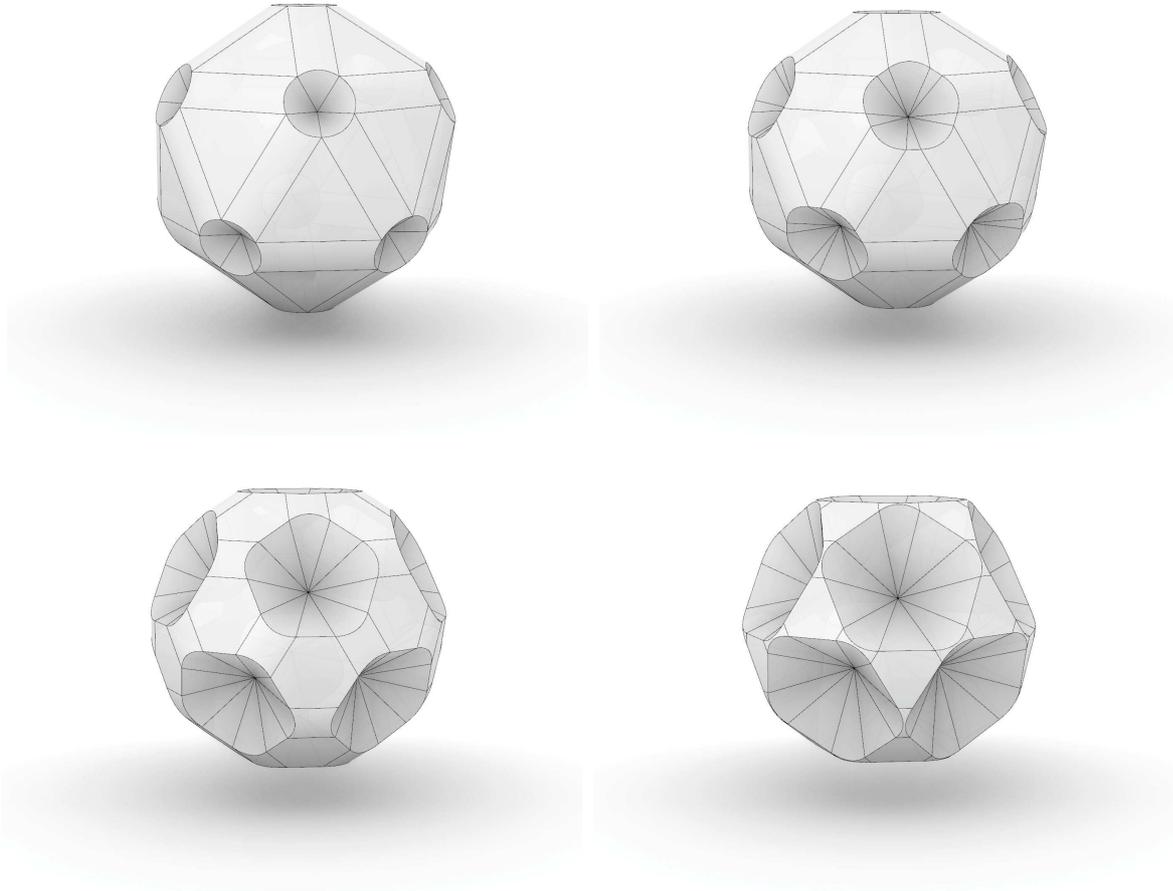
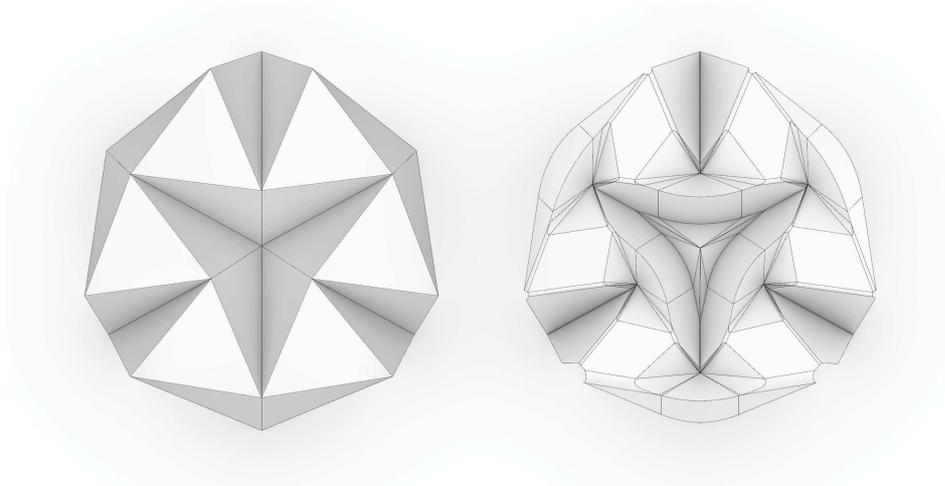
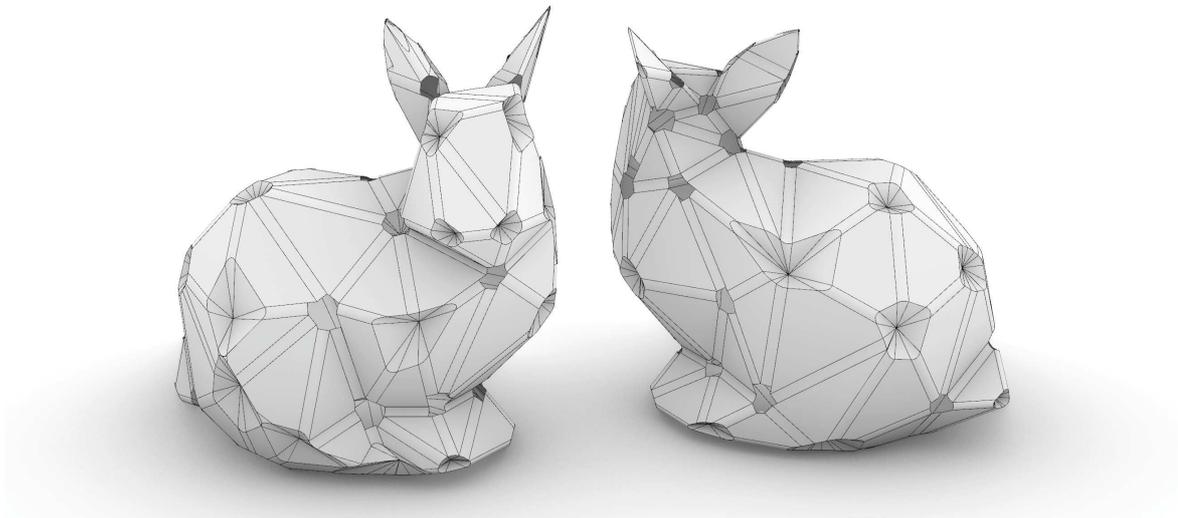


Figure 17: Rounded Icosahedron with different apex depths.

Figure 17 shows how different initial depth of apices affect the results for the curved crease models created from a regular icosahedron. Figure 19 shows the method applied to a low resolution version of Stanford bunny. Note that at the saddle points, it is not possible to locate the normal and thus the apices as described in sec. 2.5. Figure 18 shows a curved folded origami from one piece computed from a folded shape of a part of origami tessellation by Ron Resch. Finally, fig. 20 shows a design of a shelter using our system. We plan to investigate a fabrication process for realizing shell structures through bending sheet materials based on our scheme in future.



**Figure 18:** Rounded version of Ron Resch's tessellation.



**Figure 19:** Front and back of the Stanford bunny with computed with our Grasshopper component. Notice that only non-saddle shaped vertices can be turned into curved creases.

## Acknowledgements

This work was originally posed by the first author in Structural Origami Gathering 2017 and continuously discussed in IASS Structural Morphology Workshop 2018 and Structural Origami Gatherings 2018-2020. We thank other participants, in particular Erik Demaine, Duks Koschitz and Jason Ku, of the workshop and the gatherings for stimulating discussions. Tomohiro Tachi is supported by JST PRESTO JPMJPR1927. Rupert Maleczek is supported by the FFG Austria | Project Fold2Bend | Project Number 864731.

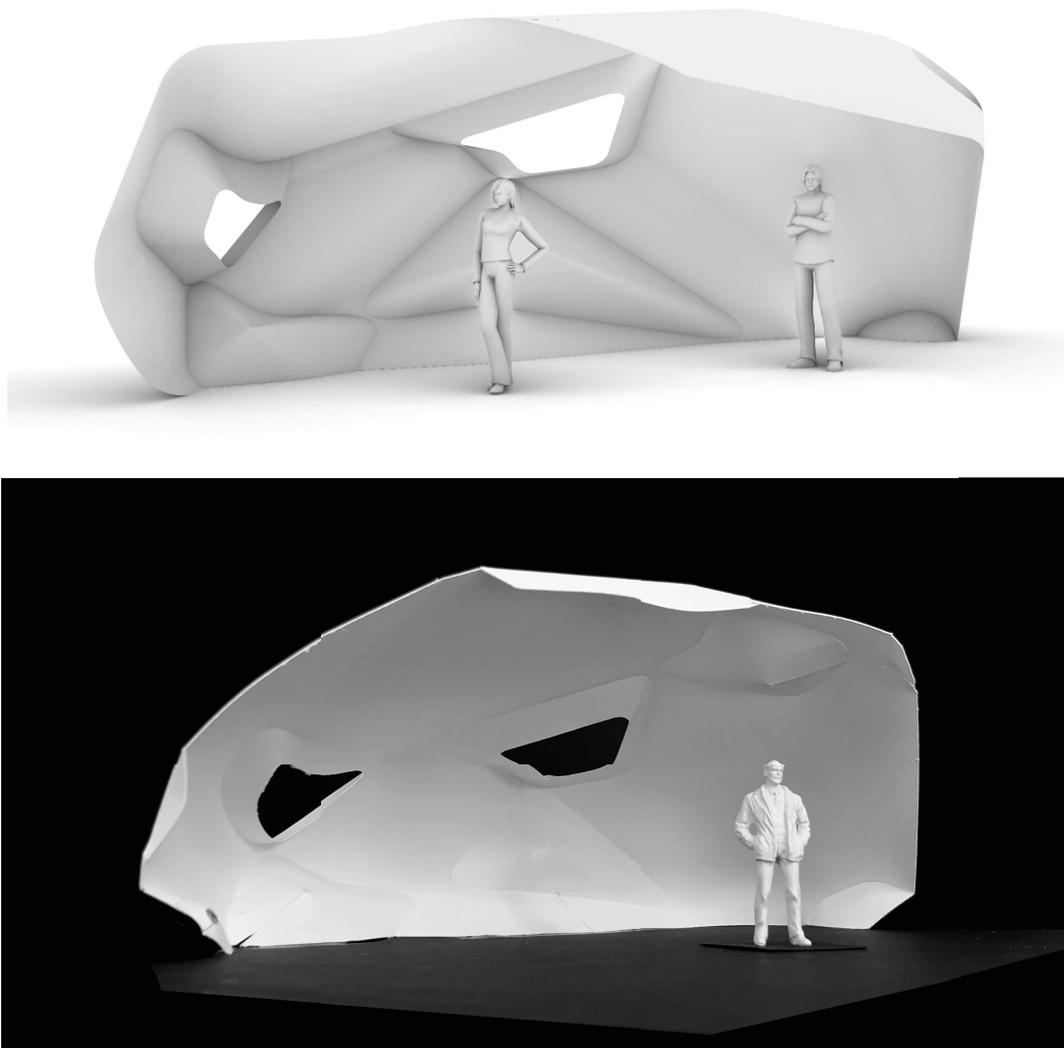


Figure 20: Curved origami based shell structure and folded model.

## References

- Abel, Z., J. Cantarella, E. D. Demaine, D. Eppstein, T. C. Hull, J. S. Ku, R. J. Lang, and T. Tachi (2016). Rigid origami vertices: Conditions and forcing sets. *Journal of Computational Geometry* 7(1), 171–184.
- Bochkanov, S. (1999–2020). ALGLIB. <https://www.alglib.net/>.
- Chandra, S., S. Bhooshan, and M. El-Sayed (2015). Curve-folding polyhedra skeletons through smoothing. In *Origami<sup>6</sup>*, pp. 231–240.
- Demaine, E., M. Demaine, D. Koschitz, and T. Tachi (2015). A review on curved creases in art, design and mathematics. *Symmetry: Culture and Science* 26(2), 145–161.
- Demaine, E. D., M. L. Demaine, D. A. Huffman, D. Koschitz, and T. Tachi (2015). Characterization of curved creases and rulings: Design and analysis of lens

- tessellations. In *Origami*<sup>6</sup>, pp. 209–230.
- Jiang, C., K. Mundilova, F. Rist, J. Wallner, and H. Pottmann (2019, November). Curve-pleated structures. *ACM Trans. Graph.* 38(6), 169:1–169:13.
- Koschitz, D. (2016). Designing with curved creases. In *Advances in Architectural Geometry*, pp. 82–103.
- Koschitz, D. (2019). Curved-crease paperfolding shell. In *Proceedings of IASS Annual Symposia*, pp. 1028–1035.
- Maleczek, R., G. Filz, C. Scheiber, M. Heimrath, and C. Preisinger (2016). Curved folded cone assemblies. In *Proceedings of IASS Annual Symposia*, pp. 268–276.
- Maleczek, R., G. Stern, A. Metzler, and C. Preisinger (2020). Large scale curved folding mechanisms. In *Impact: Design With All Senses*, pp. 539–553. Springer International Publishing.
- Mosely, J. (2002). The validity of the orb, an origami model. In *Origami*<sup>3</sup>, pp. 75–82.
- Mundilova, K. (2019a). Curved crease folds of spherical polyhedra with regular faces. In *Bridges 2019 Conference Proceedings*, pp. 423–426.
- Mundilova, K. (2019b). On mathematical folding of curved crease origami: Sliding developables and parametrizations of folds into cylinders and cones. *Computer-Aided Design* 115, 34–41.
- Piker, D. and W. Pearson (2013). Plankton. <https://github.com/meshmash/Plankton>.