Umbrella Mesh Supplementary Material

Yingying Ren*, Uday Kusupati*, Julian Panetta, Florin Isvoranu, Davide Pellis, Tian Chen, Mark Pauly

May 5, 2022

This supplementary document provides additional details on our fabrication and validation processes as well as the formulas needed to implement our generalized joint model and reduced design optimization.

1 Fabrication and Validation Details

1.1 Single-piece 3D printing

For small-scale prototypes, the easiest fabrication approach is monolithic 3D printing, which is made possible by the zero-energy rest states of umbrella meshes. Selective sintering technology provides adequate resolution to reproduce the umbrella cells' complex interior geometries while eliminating the need for support structures. The prototype in Figure 14 of the main text was printed in under 24 hours in a flexible material (Flexa Grey) using the Sinterit Lisa Pro. The layer thickness is set to 0.075mm. To aid replication, the CAD model is posted on `https://go.epfl.ch/umbrella`.

1.2 Assembling laser-cut and CNC-milled parts

Our larger-scale prototypes were created using CNC milling followed by manual assembly (Fig. 1). The triangular plates of all unit cells are identical and are cut from 6mm-thick PMMA (acrylic) sheets using a 3-axis CNC milling machine (the BZT PFU-S 2515G). The plate design has a protruding pin to accurately and consistently guide the arms into place (see Figure 1). This pin is milled to a depth of 3.1mm. A 1.2mm-high threshold enforces the angle constraint (blocking opening angles beyond 90°) by engaging with an arm's joint depression. Making 100 plates took roughly 5-6 hours of preparation and machining work.

Each pair of arms is milled as a single unit from 3mm-thick Polypropylene sheets. The CNC instructions are generated from the optimized elastic rod rest shapes using a post-processing script in Rhino3D. As each pair of arms may have different geometry, the script also outputs labels that are engraved or etched onto the ends of arms to guide the assembly process. The integral X-joint hinge is CNC milled with a 3mm diameter ballnose endmill to a depth of 2.6mm (leaving a thickness of 0.4mm for the flexure joint).

The full model is assembled manually by firmly clipping the arms onto the triangular plates. In order to enforce the target separation distance (which was a user-specified input to the design optimization), 3D printed rigid spacers are placed between the plates. In most models these can be held together passively by compression from the plates at equilibrium, but we used through-bolts in many umbrellas to tightly secure the plates and spacers together so that they do not fall out when the model is manipulated.

1.3 Model reconstruction

Once deployed, the structures are placed on a flat surface and are allowed to assume their equilibrated shapes under gravity. Calibration markers are placed around and under the structures, and photos are taken at different angles. The photos are imported into Metashape Pro, which reconstructs a volumetric mesh of the object. The meshes are then post-processed to calculate shape derivations (Euclidean distance) in Rhino3D.

2 Simulation

Umbrella meshes are simulated as a collection of discrete elastic rods coupled by our generalized joint model.

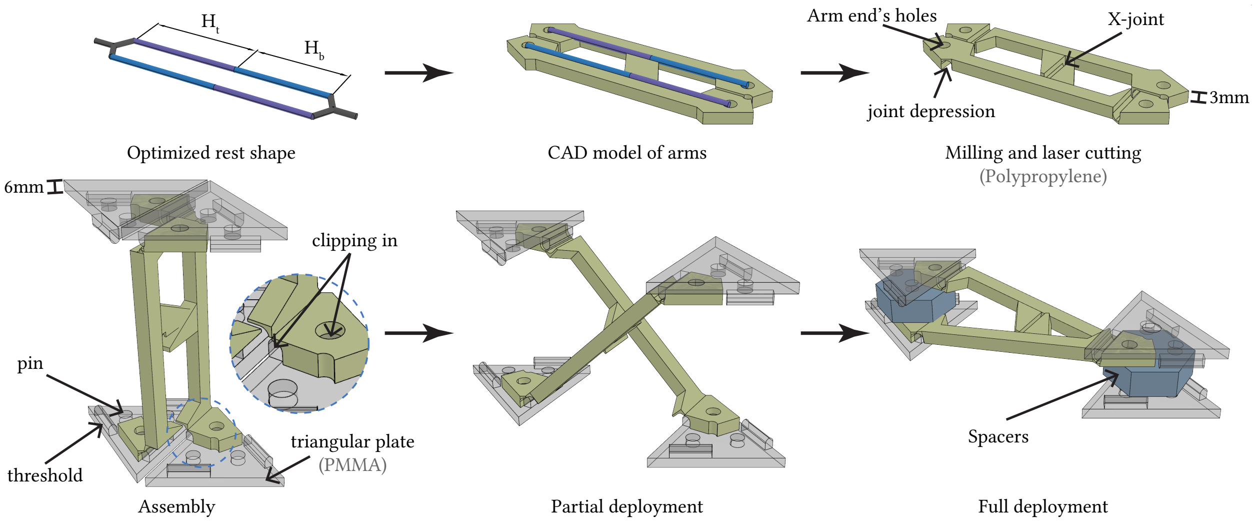

Figure 1 *Fabrication pipeline. The input to this workflow is the optimized rest shape consisting of the heights of the arms. From there, we create CAD models for CNC fabrication. The plates are milled and clipped onto the ends of the arms. Spacers are inserted in between the top and the bottom plates.*

2.1 Joint model

Our generalized joint model constrains the positions and orientations of an arbitrary number of discrete elastic rod edges bound to each hinge leaf (referred to as *terminal edges* in the paper) to be a rigid transformation of their rest positions/orientations specified as input. A joint is specified in the input by an initial position $\mathbf{q}_0$, hinge axis ("normal") $\mathbf{n}_0$, and an orthogonal vector $\mathbf{b}_0$ that together define the input joint *frame* $R_0 := \begin{bmatrix}\mathbf{b}_0 & \mathbf{n}_0 \times \mathbf{b}_0 & \mathbf{n}_0\end{bmatrix}$ (Figure 2). By convention, the opening angle α_0 in the input configuration is defined to be zero.

Let us consider the e^{th} edge bound to this joint, which corresponds to edge index j in the discrete elastic to which it belongs. (For umbrella meshes, j is always 0 or $\texttt{numEdges} - 1$ since only the first and last edges of rods are bound to joints.) This edge's tangent and normal (second material frame vector) in the input rest configuration is decomposed into the joint frame R_0 to obtain components $\tilde{\mathbf{t}}_e$ and $\tilde{\mathbf{n}}_e$, respectively. Furthermore, its midpoint may be offset away from $\mathbf{q}_0$ by some offset vector $\tilde{\mathbf{p}}_e$ decomposed in this frame. This bound edge is assigned an edge length simulation variable l_e to permit it to stretch, but the remainder of its state is completely determined by the joint's position $\mathbf{q}$, orientation $\boldsymbol{\omega}$, and opening angle variable α. Specifically, the endpoint positions $\mathbf{x}_j$ and $\mathbf{x}_{j+1}$ and the material frame angle θ^j are eliminated from the deformation variables via the formulas:

$$\begin{aligned}
\mathbf{x}_j &= \mathbf{q} + \mathbf{p}_e + \frac{l_e}{2}\mathbf{t}_e, \\
\mathbf{x}_{j+1} &= \mathbf{q} + \mathbf{p}_e - \frac{l_e}{2}\mathbf{t}_e, \\
\theta^j &= \angle(\underline{\mathbf{d}_2^j}, \mathbf{n}_e) = \operatorname{atan2}(-\mathbf{n}_e \cdot \underline{\mathbf{d}_1^j}, \mathbf{n}_e \cdot \underline{\mathbf{d}_2^j}),
\end{aligned}$$

where:

$$\begin{aligned}
\mathbf{t}_e &= R(\boldsymbol{\omega}) R_0 R_{\pm\alpha/2} \tilde{\mathbf{t}}_e, \\
\mathbf{p}_e &= R(\boldsymbol{\omega}) R_0 R_{\pm\alpha/2} \tilde{\mathbf{p}}_e, \\
\mathbf{n}_e &= R(\boldsymbol{\omega}) R_0 R_{\pm\alpha/2} \tilde{\mathbf{n}}_e.
\end{aligned}$$

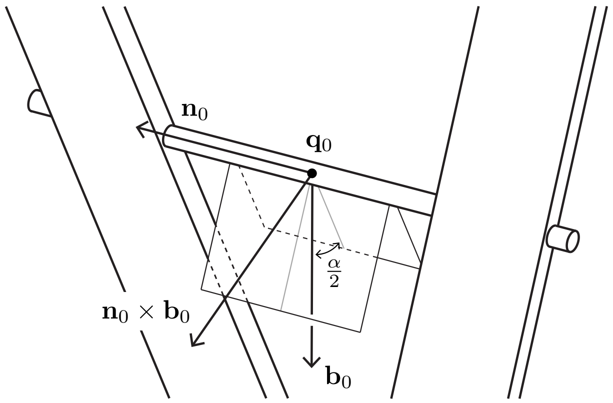

Figure 2 *Joint frame.*

Here, $R(\boldsymbol{\omega})$ is the rotation matrix specified by the angle-scaled-axis vector $\boldsymbol{\omega}$, and $R_{\pm\alpha/2}$ denotes a rotation by $\pm\alpha/2$ around the joint axis (the local Z axis in the joint frame):

$$R_{\pm\alpha/2} := \begin{bmatrix} \cos(\pm\alpha/2) & -\sin(\pm\alpha/2) & 0 \\ \sin(\pm\alpha/2) & \cos(\pm\alpha/2) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The angle is negative if e is bound to the first hinge leaf and positive if it is bound to the second.

These formulas define a nonlinear change of variables from the full set of discrete elastic rods variables to a reduced set of variables parametrizing only the space of valid umbrella mesh deformations admitted by the joint constraints. In order to run our simulation in this reduced space, we need Jacobians and Hessians of these formulas, which are presented in the following sections.

2.1.1 Jacobians

The derivatives of $\mathbf{x}_j$ and $\mathbf{x}_{j+1}$ with respect to $\mathbf{q}$, $\mathbf{p}_e$, $\mathbf{t}_e$, and l_e are trivial to determine, and the derivatives of $\mathbf{p}_e$ and $\mathbf{n}_e$ are completely analogous to the derivatives of $\mathbf{t}_e$. So we focus on the edge tangent derivatives $\frac{\partial \mathbf{t}_e}{\partial \boldsymbol{\omega}}$ and $\frac{\partial \mathbf{t}_e}{\partial \alpha}$ and the material frame angle derivatives $\frac{\partial \theta^j}{\partial \boldsymbol{\omega}}$ and $\frac{\partial \theta^j}{\partial \alpha}$. All of these derivatives can be computed easily with the chain rule:

$$\frac{\partial \mathbf{t}_e}{\partial \boldsymbol{\omega}} = \frac{\partial\big(R(\boldsymbol{\omega})\hat{\mathbf{t}}_e(\alpha)\big)}{\partial \boldsymbol{\omega}}, \qquad \frac{\partial \mathbf{t}_e}{\partial \alpha} = R(\boldsymbol{\omega}) R_0 \frac{\partial R_{\pm\alpha/2}}{\partial \alpha} \tilde{\mathbf{t}}_e, \qquad \frac{\partial R_{\pm\alpha/2}}{\partial \alpha} = \frac{1}{2} \begin{bmatrix} -\sin(\pm\alpha/2) & -\cos(\pm\alpha/2) & 0 \\ \cos(\pm\alpha/2) & -\sin(\pm\alpha/2) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\left(\frac{\partial \theta^j}{\partial \boldsymbol{\omega}}\right)^{\top} = -\mathbf{d}_1^j \cdot \frac{\partial \mathbf{n}_e}{\partial \boldsymbol{\omega}} + \underline{\mathbf{d}_1^j} \cdot \left(\frac{\partial P_{\widehat{\mathbf{t}^j}}^{\mathbf{t}_e} \underline{\widehat{\mathbf{d}_2^j}}}{\partial \mathbf{t}_e} \frac{\partial \mathbf{t}_e}{\partial \boldsymbol{\omega}} \right),$$

$$\frac{\partial \theta^j}{\partial \alpha} = -\mathbf{d}_1^j \cdot \frac{\partial \mathbf{n}_e}{\partial \alpha} + \underline{\mathbf{d}_1^j} \cdot \left(\frac{\partial P_{\widehat{\mathbf{t}^j}}^{\mathbf{t}_e} \underline{\widehat{\mathbf{d}_2^j}}}{\partial \mathbf{t}_e} \frac{\partial \mathbf{t}_e}{\partial \alpha} \right),$$

where we defined $\hat{\mathbf{t}}_e(\alpha) := R_0 R_{\pm\alpha/2} \tilde{\mathbf{t}}_e$, and the derivatives related to the parallel transport of edges and the rotation variable $\boldsymbol{\omega}$ that appear in these expressions, $\frac{\partial P_{\widehat{\mathbf{t}^j}}^{\mathbf{t}_e} \underline{\widehat{\mathbf{d}_2^j}}}{\partial \mathbf{t}_e}$ and $\frac{\partial (R(\boldsymbol{\omega})\mathbf{v})}{\partial \boldsymbol{\omega}}$ are provided in the supplementary material of [Panetta et al., 2019].

2.1.2 Hessians

Again, we give formulas only for the nontrivial terms of the Hessian:

$$
\begin{aligned}
\frac{\partial^2 \mathbf{t}_e}{\partial \boldsymbol{\omega}^2} &= \frac{\partial^2 (R(\boldsymbol{\omega})\hat{\mathbf{t}}_e(\alpha))}{\partial \boldsymbol{\omega}^2}, \qquad \frac{\partial^2 \mathbf{t}_e}{\partial \alpha \partial \boldsymbol{\omega}} = \frac{\partial (R(\boldsymbol{\omega}) R_0 \frac{\partial R_{\pm\alpha/2}}{\partial \alpha} \tilde{\mathbf{t}}_e)}{\partial \boldsymbol{\omega}}, \\
\frac{\partial^2 \mathbf{t}_e}{\partial \alpha^2} &= R(\boldsymbol{\omega}) R_0 \frac{\partial^2 R_{\pm\alpha/2}}{\partial \alpha^2} \tilde{\mathbf{t}}_e = -\frac{\mathbf{t}_e}{4}, \\
\frac{\partial^2 \theta^j}{\partial \boldsymbol{\omega}^2} &= -\mathrm{sym}\left(\left(\frac{\partial \mathbf{n}_e}{\partial \boldsymbol{\omega}} \right)^\top \frac{\partial \mathbf{d}_1^j}{\partial \boldsymbol{\omega}} \right) - \mathbf{d}_1^j \cdot \frac{\partial^2 \mathbf{n}_e}{\partial \boldsymbol{\omega}^2}, \\
\frac{\partial^2 \theta^j}{\partial \alpha \partial \boldsymbol{\omega}} &= -\left(\frac{\partial \mathbf{n}_e}{\partial \alpha} \right)^\top \frac{\partial \mathbf{d}_1^j}{\partial \boldsymbol{\omega}} - \mathbf{d}_1^j \cdot \frac{\partial^2 \mathbf{n}_e}{\partial \alpha \partial \boldsymbol{\omega}} - \frac{1}{2} \left(\frac{\partial \mathbf{t}_e}{\partial \boldsymbol{\omega}} \right)^\top \left(\mathbf{t}_e \times \frac{\partial \mathbf{t}_e}{\partial \alpha} \right), \\
\frac{\partial^2 \theta^j}{\partial \alpha \partial \alpha} &= -\left(\frac{\partial \mathbf{n}_e}{\partial \alpha} \right)^\top \frac{\partial \mathbf{d}_1^j}{\partial \alpha} - \left(\mathbf{d}_1^j \cdot \frac{\partial^2 \mathbf{n}_e}{\partial \alpha^2} \right), \\
\frac{\partial \mathbf{d}_1^j}{\partial \boldsymbol{\omega}} &= \frac{\partial \mathbf{n}_e}{\partial \boldsymbol{\omega}} \times \mathbf{t}_e + \mathbf{n}_e \times \frac{\partial \mathbf{t}_e}{\partial \boldsymbol{\omega}}, \qquad \frac{\partial \mathbf{d}_1^j}{\partial \alpha} = \frac{\partial \mathbf{n}}{\partial \alpha} \times \mathbf{t}_e + \mathbf{n}_e \times \frac{\partial \mathbf{t}_e}{\partial \alpha}.
\end{aligned}
$$

A formula for the rotated vector Hessian $\frac{\partial^2 (R(\boldsymbol{\omega})\mathbf{v})}{\partial \boldsymbol{\omega}^2}$ appearing in the expression for $\frac{\partial^2 \mathbf{t}_e}{\partial \boldsymbol{\omega}^2}$ is provided in the supplementary material of [Panetta et al., 2019]. The Hessians of $\mathbf{p}_e$ and $\mathbf{n}_e$ are obtained by replacing $\tilde{\mathbf{t}}_e$ in the Hessian of $\mathbf{t}_e$ with $\tilde{\mathbf{p}}_e$ and $\tilde{\mathbf{n}}_e$, respectively.

2.2 Deployment energy derivatives

Our simulation needs the gradients and Hessian of the linear actuator potential energy, which is a function of the position and orientation of the two joints representing the top and bottom plates:

$$
\begin{aligned}
\mathcal{L}(\mathbf{n}^t, \mathbf{n}^b, \mathbf{q}^t, \mathbf{q}^b) &= \frac{1}{2} w_1 \left\| \mathbf{n}^t + \mathbf{n}^b \right\|^2 + \frac{1}{2} w_2 \left\| (I - \mathbf{a} \otimes \mathbf{a})(\mathbf{q}^t - \mathbf{q}^b) \right\|^2 + \frac{1}{2} w_3 (\mathbf{a} \cdot (\mathbf{q}^t - \mathbf{q}^b) - s_{\text{target}})^2, \\
&= w_1 \mathbf{n}^t \cdot \mathbf{n}^b + \frac{1}{2} w_2 \left(\|\mathbf{q}^t - \mathbf{q}^b\|^2 - s^2 \right) + \frac{1}{2} w_3 (s - s_{\text{target}})^2,
\end{aligned}
$$

where $\mathbf{a} := \frac{\mathbf{n}^t - \mathbf{n}^b}{\|\mathbf{n}^t - \mathbf{n}^b\|}$ is the actuator's unit axis, and $s := \mathbf{a} \cdot (\mathbf{q}^t - \mathbf{q}^b)$ is the current plate separation.

The first derivatives of $\mathcal{L}$ with respect to its top-plate arguments are:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{n}^t} &= w_1 \mathbf{n}^b + \left(w_2 s + w_3 (s - s_{\text{target}}) \right) \frac{\partial s}{\partial \mathbf{n}^t}, \\
\frac{\partial \mathcal{L}}{\partial \mathbf{q}^t} &= w_2 (\mathbf{q}^t - \mathbf{q}^b) + w_3 (s - s_{\text{target}}) \frac{\partial s}{\partial \mathbf{q}^t}, \\
\frac{\partial s}{\partial \mathbf{n}^t} &= \left(\frac{\partial \mathbf{a}}{\partial \mathbf{n}^t} \right)^\top (\mathbf{q}^t - \mathbf{q}^b) = (I - \mathbf{a} \otimes \mathbf{a}) \frac{\mathbf{q}^t - \mathbf{q}^b}{\|\mathbf{n}^t - \mathbf{n}^b\|} = \frac{\mathbf{q}^t - \mathbf{q}^b - s\mathbf{a}}{\|\mathbf{n}^t - \mathbf{n}^b\|}, \\
\frac{\partial s}{\partial \mathbf{q}^t} &= \mathbf{a}.
\end{aligned}
$$

The bottom-plate formulas are analogous up to signs. The second derivatives are:

$$
\begin{aligned}
\frac{\partial^2 \mathcal{L}}{\partial \mathbf{n}^t \partial \mathbf{n}^t} &= \qquad (w_2 + w_3) \frac{\partial s}{\partial \mathbf{n}^t} \otimes \frac{\partial s}{\partial \mathbf{n}^t} + \left(w_2 s + w_3 (s - s_{\text{target}}) \right) \frac{\partial^2 s}{\partial \mathbf{n}^t \partial \mathbf{n}^t}, \\
\frac{\partial^2 \mathcal{L}}{\partial \mathbf{n}^t \partial \mathbf{n}^b} &= \quad w_1 I + (w_2 + w_3) \frac{\partial s}{\partial \mathbf{n}^t} \otimes \frac{\partial s}{\partial \mathbf{n}^b} + \left(w_2 s + w_3 (s - s_{\text{target}}) \right) \frac{\partial^2 s}{\partial \mathbf{n}^t \partial \mathbf{n}^b},
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 \mathcal{L}}{\partial \mathbf{q}^t \partial \mathbf{q}^t} &= \quad w_2 I + w_3 \frac{\partial s}{\partial \mathbf{q}^t} \otimes \frac{\partial s}{\partial \mathbf{q}^t}, \\
\frac{\partial^2 \mathcal{L}}{\partial \mathbf{q}^t \partial \mathbf{q}^b} &= -w_2 I - w_3 \frac{\partial s}{\partial \mathbf{q}^t} \otimes \frac{\partial s}{\partial \mathbf{q}^b}, \\
\frac{\partial^2 s}{\partial \mathbf{n}^t \partial \mathbf{n}^t} &= \frac{1}{\|\mathbf{n}^t - \mathbf{n}^b\|} \left(-\frac{\partial s}{\partial \mathbf{n}^t} \otimes \mathbf{a} - \mathbf{a} \otimes \frac{\partial s}{\partial \mathbf{n}^t} - \frac{s}{\|\mathbf{n}^t - \mathbf{n}^b\|} (I - \mathbf{a} \otimes \mathbf{a}) \right), \\
\frac{\partial^2 s}{\partial \mathbf{n}^t \partial \mathbf{n}^b} &= \frac{1}{\|\mathbf{n}^t - \mathbf{n}^b\|} \left(\frac{\partial s}{\partial \mathbf{n}^t} \otimes \mathbf{a} - \mathbf{a} \otimes \frac{\partial s}{\partial \mathbf{n}^b} + \frac{s}{\|\mathbf{n}^t - \mathbf{n}^b\|} (I - \mathbf{a} \otimes \mathbf{a}) \right).
\end{aligned}
$$

The top and bottom plate normals are given in terms of the corresponding joints' simulation variables as:

$$
\mathbf{n}^t = R(\boldsymbol{\omega}^t) R_0^t \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = R(\boldsymbol{\omega}^t) \mathbf{n}_0^t, \qquad \mathbf{n}^b = R(\boldsymbol{\omega}^b) \mathbf{n}_0^b,
$$

enabling the derivative of the linear actuator potential with respect to simulation variables $\boldsymbol{\omega}^t$ and $\boldsymbol{\omega}^b$ to be calculated using the chain rule and the formulas for the Jacobian and Hessian of rotated vectors from [Panetta et al., 2019].

Note that the linear actuator introduces additional nonzero blocks to the total potential energy Hessian that couple top- and bottom-plate variables.

2.3 Angle barrier term derivatives

For completeness, we provide the following first and second derivatives of the barrier function used to enforce the joint angle bounds in our simulation:

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}x} \left(\log \left(\frac{b-a}{b-x} \right) \right)_+^3 &= \frac{3}{b-x} (l(x))_+^2, \qquad l(x) := \log \left(\frac{b-a}{b-x} \right), \\
\frac{\mathrm{d}^2}{\mathrm{d}x^2} \left(\log \left(\frac{b-a}{b-x} \right) \right)_+^3 &= \frac{3\big(l(x) + 2\big)}{(b-x)^2} (l(x))_+.
\end{aligned}
$$

We note that the second derivative approaches 0 as $x \to a$, showing this barrier term to be C^2.

2.4 Simulation model normalization

To simplify the weight tuning process of the simulation problem and subsequent design optimization problem, we uniformly scale the input models such that their bounding box diagonals have length 1mm. At this scale, we find through experiments that to simulate the rigid plate behavior, it is sufficient to set the Young's modulus for the rods in the abstract plates to be 10 times the Young's modulus of the arm rods.

2.5 Deployment weight scheduling

For the deployment simulation, we set the weights for each term as follows:

1. Elastic energy: this term is left unweighted.

2. Deployment energy: three stiffness parameters w_1, w_2, w_3 discussed in the main text are assigned the same value. This value is either set immediately to 1 to achieve the target plate separation in a single equilibrium solve, or is increased more gradually to 1 in several simulation stages.

3. Attraction energy:

(a) This term is normalized by the square of the length of the bounding box diagonal.

(b) The weight ϵ is initially set to 10^{-3} to stabilize the deployment of the conformal-mapping-initialized design, where the top and bottom heights are symmetrical.

(c) During the subsequent design optimization, this weight is gradually reduced to 10^{-7} to ensure that the optimized model is in true equilibrium without the fictitious attraction forces.

(d) The diagonal weight matrices $\overline{W_i}$, W_i, and W_b offer the user direct control over the importance of fitting each midsurface point. However, for all our results, we set $\overline{W_i}$ to the identity matrix scaled so that all entries sum to w_{init} (placing equal importance fitting each interior points to its initial position). We assign the entries of diagonal matrices W_i and W_b such that the sum of all of these entries equals $1 - w_{\text{init}}$; normally we assign the same value to each of these entries, but we can scale up entries in W_b relative to W_i to place stronger emphasis on fitting to the boundary. To further stabilize the initial deployment of the conformal-mapping-initialized design, we put the main emphasis on the input joint position by setting $w_{\text{init}} = 1 - 10^{-4}$. However, we use $w_{\text{init}} = 0.1$ for the remainder of our pipeline.

4. Joint angle barrier energy: this term is left unweighted.

To simulate the undeployment of a structure, we can first set a higher target plate separation and minimize the total potential energy, then disable the linear actuators by setting the weight to 0. The reason that we need to first force the structure to open by pushing the plates apart is that all the models we have tested in this paper is bistable. So if we simply deactivate the deployment term and compute the local minimum, the structure will stay in the deployed state rather than retract back to the rest state.

3 Design Optimization

Our design optimization minimizes the following function of the design variables:

$$J(\mathbf{p}) = \mathcal{J}(\mathbf{x}^*(\mathbf{p}), \mathbf{p}), \qquad \mathcal{J}(\mathbf{x}, \mathbf{p}) := \mathcal{T}(\mathbf{x}) + \mathcal{E}(\mathbf{x}, \mathbf{p}) + \hat{\mathcal{D}}(\mathbf{x}, \mathbf{p}).$$

3.1 Gradients and Hessian-vector products

We perform analytical sensitivity analysis using the adjoint method. We first solve for the adjoint state $\mathbf{y}$:

$$\frac{\partial^2 \mathcal{U}}{\partial \mathbf{x}^2} \mathbf{y} = \frac{\partial \mathcal{J}}{\partial \mathbf{x}},$$

and use it to efficiently evaluate the gradient with respect to the design variables:

$$\frac{\partial J}{\partial \mathbf{p}} = \frac{\partial \mathcal{J}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}^*(\mathbf{p})}{\partial \mathbf{p}} + \frac{\partial \mathcal{J}}{\partial \mathbf{p}} = -\mathbf{y} \cdot \frac{\partial^2 \mathcal{U}}{\partial \mathbf{x} \partial \mathbf{p}} + \frac{\partial \mathcal{J}}{\partial \mathbf{p}}.$$

To accelerate design optimization, we use a Newton-CG solver which requires exact Hessian-vector products. We obtain these with second-order sensitivity analysis. The first step is to calculate the directional derivative of $\mathbf{x}^*(\mathbf{p})$ and $\mathbf{y}$ along a given design perturbation $\delta\mathbf{p}$:

$$\begin{aligned}
&\frac{\partial^2 \mathcal{U}}{\partial \mathbf{x}^2} \delta\mathbf{x}^* = -\frac{\partial^2 \mathcal{U}}{\partial \mathbf{x} \partial \mathbf{p}} \delta\mathbf{p}, \\
&\frac{\partial^2 \mathcal{U}}{\partial \mathbf{x}^2} \delta\mathbf{y} = -\frac{\partial^3 \mathcal{U}}{\partial \mathbf{x} \partial \mathbf{x} \partial \mathbf{p}} : (\mathbf{y} \otimes \delta\mathbf{p}) - \frac{\partial^3 \mathcal{U}}{\partial \mathbf{x} \partial \mathbf{x} \partial \mathbf{x}} : (\mathbf{y} \otimes \delta\mathbf{x}^*) + \frac{\partial^2 \mathcal{J}}{\partial \mathbf{x}^2} \delta\mathbf{x}^* + \frac{\partial^2 \mathcal{J}}{\partial \mathbf{x} \partial \mathbf{p}} \delta\mathbf{p}.
\end{aligned}$$

With these quantities in hand, we can evaluate the design optimization Hessian-vector product:

$$\frac{\partial^2 J}{\partial \mathbf{p}^2} \delta\mathbf{p} = -\delta\mathbf{y} \cdot \frac{\partial^2 \mathcal{U}}{\partial \mathbf{x} \partial \mathbf{p}} - \mathbf{y} \cdot \left(\frac{\partial^3 \mathcal{U}}{\partial \mathbf{x} \partial \mathbf{p} \partial \mathbf{x}} \delta\mathbf{x}^* + \frac{\partial^3 \mathcal{U}}{\partial \mathbf{x} \partial \mathbf{p} \partial \mathbf{p}} \delta\mathbf{p} \right) + \frac{\partial^2 \mathcal{J}}{\partial \mathbf{p} \partial \mathbf{x}} \delta\mathbf{x}^* + \frac{\partial^2 \mathcal{J}}{\partial \mathbf{p}^2} \delta\mathbf{p}.$$

While third derivatives appear in these expressions, only *directional* third derivatives are required along the $(\delta\mathbf{p}, \delta\mathbf{x}^*)$ directions; we compute these with (scalar) forward-mode automatic differentiation of a hand-coded Hessian-vector product routine for $\mathcal{U}$. We note that all linear systems solved in the course of sensitivity analysis have the same system matrix $\frac{\partial^2\mathcal{U}}{\partial\mathbf{x}^2}$ used in simulation (allowing matrix factorization reuse).

Our implementation uses hand-coded analytical derivatives for the following terms beyond those needed for the Newton-based equilibrium solver:

$$\frac{\partial^2\mathcal{U}}{\partial\mathbf{x}\partial\mathbf{p}}, \quad \frac{\partial\mathcal{J}}{\partial\mathbf{x}}, \quad \frac{\partial\mathcal{J}}{\partial\mathbf{p}}, \quad \frac{\partial^2\mathcal{J}}{\partial\mathbf{x}^2}\delta\mathbf{x}^*, \quad \frac{\partial^2\mathcal{J}}{\partial\mathbf{x}\partial\mathbf{p}}\delta\mathbf{p}, \quad \frac{\partial^2\mathcal{J}}{\partial\mathbf{p}^2}\delta\mathbf{p}, \quad \frac{\partial^2\mathcal{J}}{\partial\mathbf{x}^2}\mathbf{p}\delta\mathbf{x}^*.$$

The new terms added to $\mathcal{U}$ that were not considered in [Ren et al., 2021] (the linear actuator and barrier energies) are constant with respect to the design parameters (rest lengths), and so analogous derivative formulas given in [Ren et al., 2021] can be adapted, accounting for the slight differences in the change of variables used to enforce our generalized joint model's constraints. As for the derivatives of $\mathcal{J}$, objective term $\mathcal{E}$ is identical to the one used in [Ren et al., 2021], and $\mathcal{T}$ is the same apart from which points in the structure are fit to the target surface (here, these are the *midsurface* points described in the main paper) and the restriction of some closest point projections to the boundary curve; neither fundamentally changes the process of calculating derivatives. The first two subterms of the final deployment force objective term $\hat{\mathcal{D}}$ are fundamentally same as the contact force optimization terms of [Ren et al., 2021] (except now compressive forces are favored). The third torque term is new, but easy to differentiate:

$$\frac{\partial}{\partial\mathbf{x}}\frac{w_\tau}{2}\left\|\frac{\partial\mathcal{E}}{\partial\boldsymbol{\omega}_j}\right\|^2 = w_\tau\frac{\partial\mathcal{E}}{\partial\boldsymbol{\omega}_j}\cdot\frac{\partial^2\mathcal{E}}{\partial\boldsymbol{\omega}_j\partial\mathbf{x}}, \quad \frac{\partial}{\partial\mathbf{p}}\frac{w_\tau}{2}\left\|\frac{\partial\mathcal{E}}{\partial\boldsymbol{\omega}_j}\right\|^2 = w_\tau\frac{\partial\mathcal{E}}{\partial\boldsymbol{\omega}_j}\cdot\frac{\partial^2\mathcal{E}}{\partial\boldsymbol{\omega}_j\partial\mathbf{p}},$$

where $\frac{\partial^2\mathcal{E}}{\partial\boldsymbol{\omega}_j\partial\mathbf{x}}$ is one of the Hessian formulas implemented for the simulation, and $\frac{\partial^2\mathcal{E}}{\partial\boldsymbol{\omega}_j\partial\mathbf{p}}$ can be adapted from [Ren et al., 2021]. For the higher-order directional derivatives of this term, which would involve third-order derivatives of $\mathcal{E}$, we resort to forward-mode automatic differentiation of these first derivative formulas.

3.2 Design subspace derivatives

The design optimization derivative formulas we adopt from [Panetta et al., 2019] and [Ren et al., 2021] give the gradients with respect to the rest length of each "segment" (for umbrella meshes, these are simply the rest lengths of each distinct elastic rod in the model). However, to ensure a zero-energy rest state for our model, individual rod rest lengths cannot be changed arbitrarily. As described in the main paper, we optimize the top and bottom heights of the umbrella unit cell, which adjust the dimensions of all umbrella arms in a compatible way. Fortunately the map from these height variables to the corresponding umbrella arm lengths is linear, allowing us to translate the "segment rest length" derivative formulas into height derivative formulas. Representing the map from height variables $\mathbf{p}$ to segment rest lengths $\mathbf{p}_{\text{segment}}$ as $\mathbf{p}_{\text{segment}} = A\,\mathbf{p}$ with a sparse rectangular matrix A, we have the simple gradient and Hessian-vector product transformations from the chain rule:

$$\frac{\partial J}{\partial\mathbf{p}} = A^\top\frac{\partial J}{\partial\mathbf{p}_{\text{segment}}}, \qquad \frac{\partial^2 J}{\partial\mathbf{p}^2}\delta\mathbf{p} = A^\top\frac{\partial^2 J}{\partial{\mathbf{p}_{\text{segment}}}^2}A\,\delta\mathbf{p},$$

where we abused notation by using the same symbol J to represent the objective as a function of $\mathbf{p}$ or of $\mathbf{p}_{\text{segment}}$.

3.3 Design optimization weight scheduling

For the design optimization, we set the weights for each objective term as follows:

1. Elastic energy: this term is normalized by the initial elastic energy of the structure deployed from the conformal-mapping-initialized state.

2. Attraction energy:

   (a) This term is normalized by the square of the length of the bounding box diagonal.

   (b) For most model shown in the paper, we set the weight for this term to be 10^6. If there are still significant deviation from the target surface, we increase this weight further to improve the target fitting objective.

   (c) For some models, we find it helpful to recompute the target positions $\overline{\mathbf{m}}_i$ after some iterations to prevent the second term in surface fitting function defined in main text from fighting the other two terms in the function.

3. Deployment force:

   (a) This term is normalized by the square of the length of the bounding box diagonal.

   (b) We set c_{min} in the objective function as the 30^{th} percentile of the normal components of the forces at all unit cells.

   (c) We set w_c in the objective function to be between 10^8 and 10^{10} and w_t, w_τ to be 0 for all models shown in the paper. We disabled the tangential and torque terms because we do not have enough design freedom to lower these two terms using our current parameterization. In practice, we did not observe significant shearing in the structures resulted from the tangential forces neither in simulation nor in the physical models. We can further explore the effects of these two terms in future enriched design spaces.

References

[Panetta et al., 2019] Panetta, J., Konaković-Luković, M., Isvoranu, F., Bouleau, E., and Pauly, M. (2019). X-shells: A new class of deployable beam structures. *ACM Transactions on Graphics (TOG)*, 38(4):1–15.

[Ren et al., 2021] Ren, Y., Panetta, J., Chen, T., Isvoranu, F., Poincloux, S., Brandt, C., Martin, A., and Pauly, M. (2021). 3d weaving with curved ribbons. *ACM Transactions on Graphics (TOG)*, 40(4):1–15.