C-shells Supplementary Material

Quentin Becker, Seiichi Suzuki, Yinging Ren, Davide Pellis, Julian Panetta, Mark Pauly

October 3, 2023

This document provides additional technical details regarding our strategy for enforcing the average opening angle constraint during simulation, our energy nondimensionalization and stress calculation, our arc-length reparameterization of cubic splines, the terms used our planarization algorithm, the fairness regularization term used in the design optimization objective, and our process for calculating gradients for the design optimization.

1 Average Angle Constraint Enforcement

As discussed in Section 4.1 of the main paper, the approach that Panetta et al. (2019) use for imposing the equality constraint on the average opening angles during the deployment simulation is problematic in situations where the deployment path constitutes a direction of negative curvature; in these cases, the elastic energy Hessian H they attempt to factorize at each Newton step is indefinite even at minimizers of the constrained optimization problem. As a result, the solver applies unnecessary modifications to the Hessian (severely damping steps and slowing progress) and refuses to converge, falsely believing the minimum to be a saddle point.

We address this issue by applying a sparse linear change of variables to make the average angle (i.e., the deployment path parameter) an explicit variable of the optimization.

1.1 Change of Simulation Variables

A naïve choice for this change of variables can easily make the transformed Hessian dense. For instance, solving for the last joint opening angle variable in terms of the average angle and the other $n-1$ opening angles corresponds to the change of variables:

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n_{\text{joints}}-2} \\ \alpha_{n_{\text{joints}}-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ -1 & -1 & -1 & \dots & -1 & n \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n_{\text{joints}}-2} \\ \bar{\alpha}_{\mathbf{x}} \end{bmatrix},$$

the last dense row of which would contribute a dense $n \times n$ block to the transformed Hessian. However, other choices are possible: the important property is that the left $n-1$ columns must average to zero (i.e., be orthogonal to the constraint and not influence the average deployment angle), while the last column must average to one.

We propose a different symmetric formula that achieves this goal while minimizing the number of nonzero entries in any row, thereby *minimizing the fill-in of the transformed Hessian:*

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n_{\text{joints}}-2} \\ \alpha_{n_{\text{joints}}-1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 1 \\ -1 & 1 & 0 & \dots & 0 & 1 \\ 0 & -1 & 1 & \dots & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 1 \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix}}_{A_\alpha} \begin{bmatrix} \bar{\alpha}_0 \\ \bar{\alpha}_1 \\ \bar{\alpha}_2 \\ \vdots \\ \bar{\alpha}_{n_{\text{joints}}-2} \\ \bar{\alpha}_{\mathbf{x}} \end{bmatrix}.$$

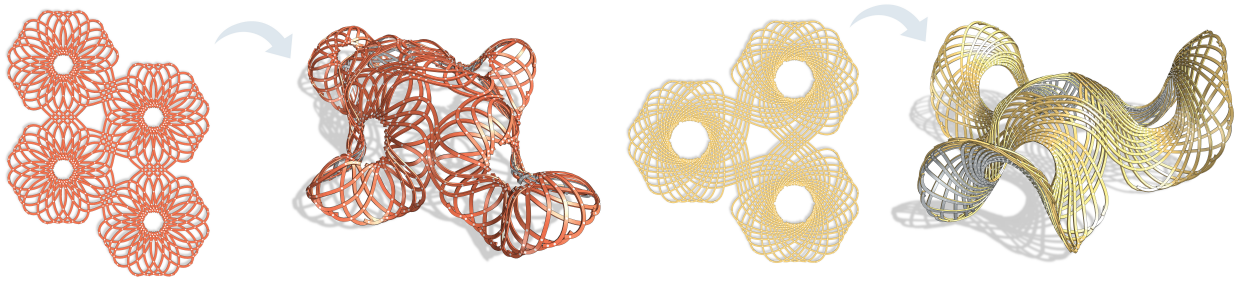

Figure 1 *Some patterns can be repeated and stitched in the flat layout during the design conception phase to generate interesting deployed states.*

The new joint angle variables α_0 now correspond to simultaneously opening and closing a pair of joints at unit speed (preserving the average angle).

Here A_α is the joint-angle sub-block of the full sparse matrix A that implements the change of variables $\mathbf{x} = A\bar{\mathbf{x}}$ (expressing the original simulation variables $\mathbf{x}$ in terms of the new variables $\bar{\mathbf{x}}$). The rest of the rows/columns of A are taken from the $n_\mathbf{p} \times n_\mathbf{p}$ identity matrix so that only the joint angles are affected by the change of variables.

Our sparse change of variables can be adapted for the case that only a subset I of the joints are actuated. We then construct a smaller $|I| \times |I|$ matrix A_α using the same approach above but restricted to the angle variables of actuated joints. The remaining rows/columns of A are once more taken from the identity matrix.

We use these new simulation variables $\bar{\mathbf{x}}$ throughout our pipeline, but refer to them as $\mathbf{x}$ in the main paper to avoid cluttered notation.

1.2 Impact on the Gradients and Hessians

The gradient of the total elastic energy $\frac{\partial}{\partial \bar{\mathbf{x}}} E(A\bar{\mathbf{x}}, \mathbf{p})$ can be obtained via the chain rule as $\frac{\partial E}{\partial \mathbf{x}} A$. Similarly, the Hessian of the total elastic energy $\frac{\partial^2}{\partial \bar{\mathbf{x}}^2} E(A\bar{\mathbf{x}}, \mathbf{p})$ is given by

$$\frac{\partial}{\partial \bar{\mathbf{x}}} \left(\frac{\partial E}{\partial \mathbf{x}} A \right) = A^\mathsf{T} \frac{\partial^2 E}{\partial \mathbf{x}^2} A.$$

To study the impact of this change of variables on the Hessian sparsity pattern, we note that:

$$A^\mathsf{T} \frac{\partial^2 E}{\partial \mathbf{x}^2} A = \sum_{k,l} h_{k,l} \mathbf{a}_k^\top \mathbf{a}_l,$$

where $h_{k,l}$ is an element of the original Hessian $\frac{\partial^2 E}{\partial \mathbf{x}^2}$, and *row vector* $\mathbf{a}_k$ denotes the k^{th} row of A, making the expression $\mathbf{a}_k^\top \mathbf{a}_l$ an outer product. Since these rows all contain a 1 in the location associated with average angle variable $\bar{\alpha}_\mathbf{x}$, the sum of outer products will fill in the row and column of $\frac{\partial^2 E}{\partial \bar{\mathbf{x}}^2}$ associated with $\bar{\alpha}_\mathbf{x}$. However, this variable is pinned during deployment, *removing the highly dense row and column of the transformed Hessian.* After this removal, our change of variables leaves every row/column of the transformed Hessian highly sparse. In total, it introduces $O(n_{\text{joints}})$ new nonzero entries, far fewer than the $O(n_{\text{joints}}^2)$ entries introduced by the naïve approach.

1.3 Impact on the Deployment

In Panetta et al. (2019), the deployed equilibrium state for a given opening angle $\bar{\alpha}^{\text{tgt}}$ was found by solving a constrained optimization problem:

$$\mathbf{x}^*_{3\mathrm{D}}(\mathbf{p}, \bar{\alpha}^{\mathrm{tgt}}) := \underset{\mathbf{x}}{\operatorname{argmin}}\, E(\mathbf{x}, \mathbf{p}) \quad \text{s.t.} \quad \mathbf{a}^{\mathsf{T}}\mathbf{x} = \bar{\alpha}^{\mathrm{tgt}},$$

where the inner product with vector $\mathbf{a}$ extracts and averages the X-shell's opening angle variables. The associated KKT system,

$$\begin{bmatrix} \dfrac{\partial^2 E}{\partial \mathbf{x}^2} & \mathbf{a} \\ \mathbf{a}^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\tilde{\lambda} \end{bmatrix} = \begin{bmatrix} -\left(\dfrac{\partial E}{\partial \mathbf{x}}\right)^{\mathsf{T}} \\ \delta\bar{\alpha} \end{bmatrix},$$

determines the infinitesimal change $\delta\mathbf{x}$ in state variable $\mathbf{x}^*_{3\mathrm{D}}(\bar{\mathbf{p}})$ induced by incrementing the average opening angle by $\delta\bar{\alpha}$, where $\delta\tilde{\lambda}$ is the Lagrange multiplier associated to the minimization problem. The Hessian of the energy evaluated at the current intermediate deployed state $\mathbf{x}^*_{3\mathrm{D}}(\bar{\mathbf{p}})$ may not be positive definite. Formerly, the Hessian would be modified in this case to be positive definite.

After our change of variables, the KKT system becomes

$$\begin{cases} \dfrac{\partial^2 E}{\partial \bar{\mathbf{x}}_{\setminus\bar{\alpha}}{}^2}\delta\bar{\mathbf{x}}_{\setminus\bar{\alpha}} & = -\left(\dfrac{\partial E}{\partial \bar{\mathbf{x}}_{\setminus\bar{\alpha}}}\right)^{\mathsf{T}} \\ \delta\bar{\alpha}_{\mathbf{x}} & = \delta\bar{\alpha}, \end{cases}$$

where we dropped the row and column of the left hand side of the KKT system that are associated to the pinned average angle variable. The transformed Hessian of the total elastic energy $\frac{\partial^2 E}{\partial \bar{\mathbf{x}}_{\setminus\bar{\alpha}}{}^2}$ is now ensured to be positive definite at the intermediate deployed equilibrium state $\bar{\mathbf{x}}^*_{3\mathrm{D}}(\bar{\mathbf{p}})$. This allows factorizing the left hand side using an efficient sparse Cholesky factorization.

The total torque applied at the actuated joints to hold the deployed configuration can now be obtained by looking at the component of the gradient associated to the average opening angle

$$\sum_{i=0}^{n_{\mathrm{joints}}-1} \frac{\partial E}{\partial \alpha_i} = \sum_{i=0}^{n_{\mathrm{joints}}-1} \frac{\partial E}{\partial \bar{\alpha}_{\mathbf{x}}} \frac{\partial \bar{\alpha}_{\mathbf{x}}}{\partial \alpha_i} = \frac{\partial E}{\partial \bar{\alpha}_{\mathbf{x}}},$$

where all the gradients are evaluated at $\bar{\mathbf{x}}^*_{3\mathrm{D}}$, the deployed state.

2 Non-dimensionalization of the Energy During Deployment

Using the new parameterization, the deployed state is given by

$$\bar{\mathbf{x}}^*_{3\mathrm{D}}(\mathbf{p}, \bar{\alpha}^{\mathrm{tgt}}) := \underset{\bar{\mathbf{x}}}{\operatorname{argmin}}\, \frac{1}{YV_0} E(\bar{\mathbf{x}}, \mathbf{p}) + T(\bar{\mathbf{x}}) \quad \text{s.t.} \quad \bar{\alpha}(\bar{\mathbf{x}}) = \bar{\alpha}^{\mathrm{tgt}},$$

where E is now the total elastic energy of the system using the new parameterization, T is the non-dimensionalized target attraction term, $\bar{\alpha}(\bar{\mathbf{x}})$ extracts the average angle variable from $\bar{\mathbf{x}}$, and $\bar{\alpha}^{\mathrm{tgt}}$ is the target average opening angle in the deployed state. The target attraction term is used here only to factor out rigid motion and is scaled by a small weight. To ease parameter tuning, we nondimensionalize the energy term by dividing it by both the Young's modulus Y of the fabrication material and the rest volume V_0 of the C-shell. This normalization factor is motivated by the fact that the bending, twisting, and stretching energies stored in the rods of a C-shell can be derived by plugging strain tensor fields $\boldsymbol{\epsilon}$ induced these deformation modes into the linear elasticity energy $\frac{1}{2}\int_\Omega \boldsymbol{\epsilon} : \mathbf{C} : \boldsymbol{\epsilon}$, where C is the fabrication material's elasticity tensor. For an isotropic material, $\mathbf{C} = Y\mathbf{C_0}(\nu)$, where $\mathbf{C_0}(\nu)$ depends only on the Poisson's ratio ν; for example $\mathbf{C_0}(0)$ is the fourth-order identity tensor. Since strain $\boldsymbol{\epsilon}$ is nondimensional, this elastic energy clearly is proportional to YV_0.

3 Stress Computation

We use a similar approach to Megaro et al. (2017) to compute stresses for Discrete Elastic Rods (DERs). Introducing a 2D coordinate system (x, y) for the material cross-section with the centerline at its origin, we seek to approximate the stress tensor $\boldsymbol{\sigma}(x, y)$. Following the kinematic assumptions of Kirchhoff rods, the state of stress in a deformed rod with torsion τ, *material curvature vector* $\boldsymbol{\kappa}$, and uniaxial stretching $\frac{\Delta l}{l}$ is:

$$\boldsymbol{\sigma}(x,y) = \tau\mu \begin{bmatrix} 0 & 0 & \dfrac{\partial\psi}{\partial x} - y \\ 0 & 0 & \dfrac{\partial\psi}{\partial y} + x \\ \dfrac{\partial\psi}{\partial x} - y & \dfrac{\partial\psi}{\partial y} + x & \dfrac{\sigma_{zz}}{\tau\mu} \end{bmatrix}, \qquad \sigma_{zz} := Y\left(\frac{\Delta l}{l} + \boldsymbol{\kappa}\cdot\begin{bmatrix} x \\ y \end{bmatrix}\right),$$

where the scalar field $\psi(x, y)$ describes how the initially planar cross-section warps out of plane under torsion to relax into equilibrium; it is determined by solving a Laplace equation Landau et al. (1986); Megaro et al. (2017); Panetta et al. (2019).

We use von Mises stress $\sigma_v = \sqrt{\frac{3}{2}}\|\boldsymbol{\sigma}_d\|_F$ to evaluate our structures' robustness, where $\boldsymbol{\sigma}_d := \boldsymbol{\sigma} - \frac{1}{3}\mathrm{tr}(\boldsymbol{\sigma})\mathbf{I}$ is the deviatoric stress. This von Mises stress simplifies to:

$$\sigma_v^2 = 3(\mu\tau)^2 \left\| \nabla\psi + \begin{bmatrix} -y \\ x \end{bmatrix} \right\|^2 + \sigma_{zz}^2,$$

and it can be shown that each of these two terms is subharmonic. This means σ_v^2 itself is subharmonic and therefore σ_v satisfies a maximum principle: the maximum von Mises stress occurs on the boundary. In other words, to evaluate failure likelihood, we can restrict our consideration to the cross-section boundary.

One notable departure of our formulation from Megaro et al. (2017) is in its discretization for DERs. We recall that the material curvature $\boldsymbol{\kappa}$ is defined by decomposing the curvature normal into its components with respect to the orthonormal material frame. Unfortunately, for DERs, discrete curvature is defined at the vertices, while the material frames are attached to edges. Megaro et al. (2017) resolve this misalignment by simply averaging the material frame vectors onto the vertices using an arithmetic mean, which is nonphysical. Instead, when evaluating stress at a sample point within the cross-section at a rod vertex, we compute two separate stress values: one according to each incident edge's material frame. Then we average these two values using the incident edge lengths as weights to obtain a single stress sample. We sample stress at every boundary node of the same finite element mesh of the cross section that we use to solve for ψ, guaranteeing that the maximum is sampled.

4 Curve Parameterization

The rest shapes of the beams composing C-shells are parameterized using interpolating cubic splines. For joints and intermediate control points along a curve, we fit a natural cubic spline to join them. The interpolation property is important for preserving the topology of the linkage (so that curve intersections occur precisely at the joints where they are intended).

4.1 Curve Discretization

Our curves are split into segments at the joints, and each segment is discretized into edges. The joint model we use requires the segments to overlap at the joints, and share an edge. That model distributes the total rest length of a segment evenly across the edges with the exception of the boundary edges. Such shared edges have a length equal to the minimum length of edges from either one of the two consecutive segments.

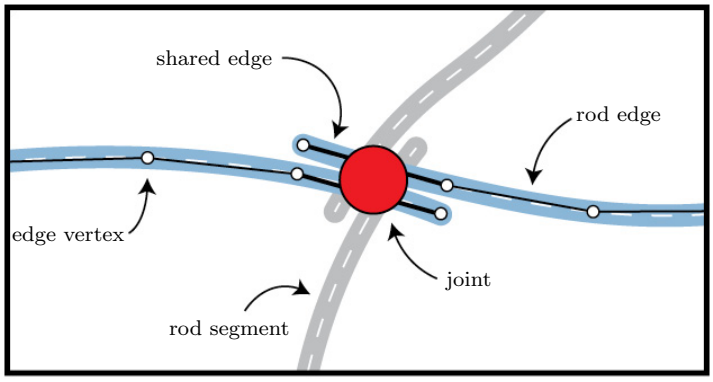

Figure 2 *Terminology of the different elements involved in our C-shell discretization. Consecutive rod segments originate from the same cubic spline, ensuring C^2 smoothness at the joints.*

In order for the discretized linkage to have an energy-free rest state, the rest curvatures measured between two consecutive edges should be carefully picked. Discretizing the curves requires a special care regarding the edge lengths, which we handle by using a arc-length parameterization of our splines. We describe our reparameterization method in the next section. Discrete points can be sampled along the curve so that they respect the above rules on the rest length distribution. Afterwards, we calculate the discrete rest curvature from the turning angle between consecutive edges as explained in Bergou et al. (2008, 2010).

This entire process, going from the design variables consisting of joint position and perpendicular offsets of interior control points to the DER rest-state quantities, is differentiable.

4.2 Constant Speed Reparameterization of Interpolating Splines

Arc-length parameterization for cubic spline cannot be expressed analytically, and must be approached numerically. Our reparameterization method consists in fitting an interpolating cubic spline through more interpolated points. We now describe our method and explain why it produces an arc-length parameterized approximation of a given $\mathcal{C}^0$ curve $\gamma : [0,1] \to \mathbb{R}^2$ of length $L(\gamma)$.

A curve $\bar{\gamma} : [0,1] \to \mathbb{R}^2$ is constant speed if and only if $\forall s \in [0,1], \quad \|\bar{\gamma}'(s)\| = L(\bar{\gamma})$.

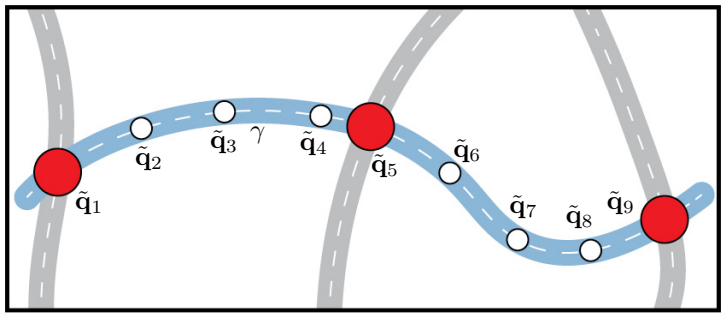

Figure 3 *Notation used.*

We start by sampling $\gamma : [0,1] \to \mathbb{R}^2$ at some locations $(t_i)_{i \in [\![1,n]\!]} \in [0,1]^n$, and call the sampled points *refinement points*: $\tilde{\mathbf{q}}_i := \gamma(t_i)$. We assume that the locations are such that t_1=0,

t_n=1, and $(t_i)_{i\in[\![1,n]\!]}$ is sorted in an increasing fashion. We define the length of the polyline joining refinement points up to i as $\tilde{L}_i := \sum_{k=2}^{i} \|\tilde{\mathbf{q}}_k - \tilde{\mathbf{q}}_{k-1}\|$, and $\tilde{s}_i := \tilde{L}_i/\tilde{L}_n$. We fit an interpolating cubic spline $\tilde{\gamma}_n$ (length $L(\tilde{\gamma}_n)$) to the set of refinement points using the knots $\tilde{s}_i$. Hence $\tilde{\gamma}_n(\tilde{s}_i) = \tilde{\mathbf{q}}_i$, and we may compute

$$\frac{\tilde{\gamma}_n(\tilde{s}_{i+1}) - \tilde{\gamma}_n(\tilde{s}_i)}{\tilde{s}_{i+1} - \tilde{s}_i} = \frac{\tilde{\mathbf{q}}_{i+1} - \tilde{\mathbf{q}}_i}{\|\tilde{\mathbf{q}}_{i+1} - \tilde{\mathbf{q}}_i\|} \sum_{k=2}^{n} \|\tilde{\mathbf{q}}_k - \tilde{\mathbf{q}}_{k-1}\|,$$

so that the norm of the above quantity equals $\tilde{L}_n$. In the limit of thiner increments $\Delta t := \max_{i\in[\![2,n]\!]}(t_i - t_{i-1})$, and large number of samples n, the length of the polyline $\tilde{L}_n$ converges to the length of the cubic spline $L(\tilde{\gamma}_n)$, which converges to the length of the initial curve $L(\gamma)$. Indeed, the cubic spline $\tilde{\gamma}_n$ approximates γ up to the fourth order on $\Delta\tilde{s} := \max_{i\in[\![2,n]\!]}(\tilde{s}_i - \tilde{s}_{i-1})$ Hall and Meyer (1976), and, by continuity of γ, $\Delta\tilde{s}$ converges to 0 as Δt goes to 0.

Therefore, for any $\epsilon > 0$, we may find a fine subdivision $(t_i)_{i\in[\![1,n]\!]}$ for a large enough number of samples n such that for any $\tilde{s}_i \leq \tilde{s} < \tilde{s}_{i+1}$

$$\begin{aligned} |\|\tilde{\gamma}_n'(s)\| - L| &\leq \left| \|\tilde{\gamma}_n'(s)\| - \frac{\|\tilde{\gamma}_n(\tilde{s}_{i+1}) - \tilde{\gamma}_n(\tilde{s}_i)\|}{\tilde{s}_{i+1} - \tilde{s}_i} \right| + \left| \frac{\|\tilde{\gamma}_n(\tilde{s}_{i+1}) - \tilde{\gamma}_n(\tilde{s}_i)\|}{\tilde{s}_{i+1} - \tilde{s}_i} - \tilde{L}_n \right| + |\tilde{L}_n - L| \\ &\leq \epsilon + 0 + \epsilon. \end{aligned}$$

The first upper bound comes from the $\mathcal{C}^1$ continuity of cubic splines on their domain, while the last upper bound comes from our previous argument about polyline length convergence. This proves that, in the limit of large refinements, our strategy produces an arc-length parameterized cubic spline that approximates γ. Increasing the total number of refinement points n improves the reparameterization at the expense of higher computational complexity.

5 Planarization

We describe our planarization algorithm and support our observations on the deployment kinematics on an actual C-shell design. The algorithm takes a B-spline surface then jointly optimizes the joints positions in the flat state and on the surface under feasibility constraints.

5.1 Variables

In this part, a C-shell is represented by its joints $\mathbf{c}_i \in \mathbb{R}^2$ and each of its constitutive rod is simplified into a polyline connecting the joints. To each joint in the flat state, we associate a position on the target surface given by the surface parameters $(u_i, v_i) \in [0,1]^2$. The surface is given by the differentiable function $S : [0,1]^2 \to \mathbb{R}^3$. We jointly optimize the joints positions in the flat state $\mathbf{c}_i$ and the corresponding target joint positions in the deployed state (u_i, v_i) so that it minimizes deviations to a set of kinematic rules we observed in some designs.

5.2 Objective

Our total planarization objective $\mathcal{E}(\mathbf{c}_1, \ldots, \mathbf{c}_n, u_1, v_1, \ldots, u_n, v_n)$ is composed of multiple terms we describe later

$$\mathcal{E} = \mathcal{E}_{\text{len}} + \mathcal{E}_{\text{eq}} + \mathcal{E}_{\alpha} + \mathcal{E}_{\text{bnd}},$$

where we omit dependencies of the different terms to the variables for clarity.

5.2.1 Segments Lengths Preservation

We first observe that stretching and twisting are often negligible compared to the bending energy in the deployed state for the cross sections used. We hence assume the polyline edges to be inextensible. The distances between neighboring joints should be preserved in the flat and deployed states. We hence define the segments lengths preservation term as

$$\mathcal{E}_{\text{len}}(\mathbf{c}_1, \dots, \mathbf{c}_n, u_1, v_1, \dots, u_n, v_n) := \frac{w_{\text{len}}}{2\,\bar{l}_0^{\,2}} \sum_c \sum_i \left(\|\mathbf{c}_{c,i+1} - \mathbf{c}_{c,i}\| - \|S(u_{c,i+1}, v_{c,i+1}) - S(u_{c,i}, v_{c,i})\| \right)^2,$$

where $\bar{l}_0$ is the surface diameter divided by the number of edges in the linkage, and w_{len} is a weight that controls the importance given to that term. The first sum is taken over each curve c in the linkage, and the second is taken over the joints connected by the polylines. We denote by $\mathbf{c}_{c,i}$ the i-th joint along curve c, similar for $(u_{c,i}, v_{c,i})$.

5.2.2 Deployed State Equilibrium (Straight Linkage)

The deployed state we obtain from the parameters (u_i, v_i) may not be at equilibrium given the rest quantities extracted from the flat joints positions $\mathbf{c}_i$. We simulate our linkage made of polylines as an ensemble of DERs connected at the joints. Each curve of the C-shell is approximated by a single DER whose vertices are given by the interpolated joints, and edges are the rod segments.

The simplified linkage has rest quantities given by $\mathbf{p}(\mathbf{c})$, where $\mathbf{c}$ holds all the flat joints positions, and deformed joints positions given by $\bar{\mathbf{c}}(u_1, v_1, \dots, u_n, v_n)$. For some material frame angles $\boldsymbol{\theta}$, the total energy of the deployed linkage is given by $E(\bar{\mathbf{c}}, \boldsymbol{\theta}, \mathbf{p}) := \sum_c E_c(\bar{\mathbf{c}}_c, \boldsymbol{\theta}_c, \mathbf{p}_c)$, where we sum over all DERs c. The average opening angle in the deployed state can be computed thanks to the function $\bar{\alpha}(\bar{\mathbf{c}})$. We set the material frame angles such that the structure is as equilibrium when holding the deployed joints in place, and solve

$$\boldsymbol{\theta}^*(\bar{\mathbf{c}}, \mathbf{p}) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, E(\bar{\mathbf{c}}, \boldsymbol{\theta}, \mathbf{p}).$$

Sensitivities of the material frame angles at equilibrium to the deployed joints positions and rest quantities can be obtained as

$$\frac{\partial^2 E}{\partial \boldsymbol{\theta}^2} \begin{bmatrix} \dfrac{\partial \boldsymbol{\theta}^*}{\partial \bar{\mathbf{c}}} & \dfrac{\partial \boldsymbol{\theta}^*}{\partial \mathbf{p}} \end{bmatrix} = - \begin{bmatrix} \dfrac{\partial^2 E}{\partial \boldsymbol{\theta} \partial \bar{\mathbf{c}}} & \dfrac{\partial^2 E}{\partial \boldsymbol{\theta} \partial \mathbf{p}} \end{bmatrix}.$$

Denoting by λ the torque applied uniformly at the joints of the structure, we measure equilbrium based on the force balance equation

$$\hat{\mathcal{E}}_{\text{eq}}(\bar{\mathbf{c}}, \boldsymbol{\theta}, \lambda, \mathbf{p}) := \frac{w_{\text{eq}}}{2(YA_0)^2} \left\| \frac{\partial E}{\partial \bar{\mathbf{c}}} - \lambda \frac{\partial \bar{\alpha}}{\partial \bar{\mathbf{c}}} \right\|^2,$$

where w_{eq} controls the importance granted to that criterion compared to other terms in the planarization objective, and the Young modulus Y and the cross section area A_0 scale the force values. The optimal torque can be computed analytically as

$$\lambda^*(\bar{\mathbf{c}}, \boldsymbol{\theta}, \mathbf{p}) = \frac{\dfrac{\partial E}{\partial \bar{\mathbf{c}}} \cdot \dfrac{\partial \bar{\alpha}}{\partial \bar{\mathbf{c}}}^{\mathsf{T}}}{\left\| \dfrac{\partial \bar{\alpha}}{\partial \bar{\mathbf{c}}} \right\|^2}.$$

We hence define the equilibrium criterion as

$$\mathcal{E}_{\text{eq}}(\bar{\mathbf{c}}, \mathbf{p}) := \hat{\mathcal{E}}_{\text{eq}}\left(\bar{\mathbf{c}}, \boldsymbol{\theta}^*(\bar{\mathbf{c}}, \mathbf{p}), \lambda^*(\bar{\mathbf{c}}, \boldsymbol{\theta}^*(\bar{\mathbf{c}}, \mathbf{p}), \mathbf{p}), \mathbf{p}\right).$$

The gradients with respect to the deployed positions can be obtained as

$$\begin{aligned}
\frac{\partial \mathcal{E}_{\text{eq}}}{\partial \bar{\mathbf{c}}} &= \frac{\partial \hat{\mathcal{E}}_{\text{eq}}}{\partial \bar{\mathbf{c}}} + \frac{\partial \hat{\mathcal{E}}_{\text{eq}}}{\partial \boldsymbol{\theta}}\frac{\partial \boldsymbol{\theta}^*}{\partial \bar{\mathbf{c}}} + \cancelto{0}{\frac{\partial \hat{\mathcal{E}}_{\text{eq}}}{\partial \lambda}}\left(\frac{\partial \lambda^*}{\partial \bar{\mathbf{c}}} + \frac{\partial \lambda^*}{\partial \boldsymbol{\theta}}\frac{\partial \boldsymbol{\theta}^*}{\partial \bar{\mathbf{c}}}\right) \\
&= \frac{w_{\text{eq}}}{2(YA_0)^2}\left(\frac{\partial E}{\partial \bar{\mathbf{c}}} - \lambda^*\frac{\partial \bar{\alpha}}{\partial \bar{\mathbf{c}}}\right)\left(\frac{\partial^2 E}{\partial \bar{\mathbf{c}}^2} - \lambda^*\frac{\partial^2 \bar{\alpha}}{\partial \bar{\mathbf{c}}^2} + \frac{\partial^2 E}{\partial \bar{\mathbf{c}}\partial \boldsymbol{\theta}}\frac{\partial \boldsymbol{\theta}^*}{\partial \bar{\mathbf{c}}}\right) \\
&= \frac{w_{\text{eq}}}{2(YA_0)^2}\left[\left(\frac{\partial E}{\partial \bar{\mathbf{c}}} - \lambda^*\frac{\partial \bar{\alpha}}{\partial \bar{\mathbf{c}}}\right)\left(\frac{\partial^2 E}{\partial \bar{\mathbf{c}}^2} - \lambda^*\frac{\partial^2 \bar{\alpha}}{\partial \bar{\mathbf{c}}^2}\right) + \mathbf{y}_{\text{eq}}^{\mathsf{T}}\frac{\partial^2 E}{\partial \boldsymbol{\theta}\partial \bar{\mathbf{c}}}\right],
\end{aligned}$$

where the adjoint state vector $\mathbf{y}_{\text{eq}}$ satisfies the following linear system

$$\frac{\partial^2 E}{\partial \boldsymbol{\theta}^2}\mathbf{y}_{\text{eq}} = -\left(\frac{\partial E}{\partial \bar{\mathbf{c}}} - \lambda^*\frac{\partial \bar{\alpha}}{\partial \bar{\mathbf{c}}}\right)^{\mathsf{T}}.$$

Similarly, the gradient with respect to the rest quantities reads

$$\begin{aligned}
\frac{\partial \mathcal{E}_{\text{eq}}}{\partial \mathbf{p}} &= \frac{\partial \hat{\mathcal{E}}_{\text{eq}}}{\partial \mathbf{p}} + \frac{\partial \hat{\mathcal{E}}_{\text{eq}}}{\partial \boldsymbol{\theta}}\frac{\partial \boldsymbol{\theta}^*}{\partial \mathbf{p}} + \cancelto{0}{\frac{\partial \hat{\mathcal{E}}_{\text{eq}}}{\partial \lambda}}\left(\frac{\partial \lambda^*}{\partial \mathbf{p}} + \frac{\partial \lambda^*}{\partial \boldsymbol{\theta}}\frac{\partial \boldsymbol{\theta}^*}{\partial \mathbf{p}}\right) \\
&= \frac{w_{\text{eq}}}{2(YA_0)^2}\left[\left(\frac{\partial E}{\partial \mathbf{p}} - \lambda^*\frac{\partial \bar{\alpha}}{\partial \mathbf{p}}\right)\frac{\partial^2 E}{\partial \mathbf{p}^2} + \mathbf{y}_{\text{eq}}^{\mathsf{T}}\frac{\partial^2 E}{\partial \boldsymbol{\theta}\partial \mathbf{p}}\right].
\end{aligned}$$

The gradient with respect to the flat joints positions are then obtained through autodifferentiation using PyTorch. We use the analytical B-spline surface derivatives to backpropagate $\frac{\partial \mathcal{E}}{\partial \bar{\mathbf{c}}}$ to the gradient with respect the surface parameters (u_i, v_i).

5.2.3 Opening Angles Increment Spread

In our designs, we also observed that the opening angles were all either opening or closing as the linkage deploys. We prevent opening angles to open and close at the same time in the linkage by reducing the variance of the opening angle increments. We define the signed opening angle of a quadrilateral q as $\alpha_q := \angle(\mathbf{c}_{q,2} - \mathbf{c}_{q,1}, \mathbf{c}_{q,4} - \mathbf{c}_{q,1})$, where the 4 vertices are numbered consistently accross the quadrilaterals in the linkage. Similar for the opening angles in the deployed states $\bar{\alpha}_q$. We define the opening angle increments as $\Delta\alpha_q = \bar{\alpha}_q - \alpha_q$, and our term reads

$$\mathcal{E}_{\alpha}(\mathbf{c}_1, \ldots, \mathbf{c}_n, u_1, v_1, \ldots, u_n, v_n) := \frac{w_{\alpha}}{2}\mathbb{V}\text{ar}_q\left[\Delta\alpha_q\right].$$

5.2.4 Soft Boundary Joint Pinning

The last term softly enforces pinning constraints to boundary joints $(u_{b,i}, v_{b,i})$ to some user-defined positions $\bar{\mathbf{c}}_{b,i}^{(\text{tar})} \in \mathbb{R}^3$ as

$$\mathcal{E}_{\text{bnd}}(u_1, v_1, \ldots, u_n, v_n) := \frac{w_{\text{bnd}}}{2\,\bar{l}_0^{\,2}}\sum_i \left\| S(u_{b,i}, v_{b,i}) - \bar{\mathbf{c}}_{b,i}^{(\text{tar})} \right\|^2,$$

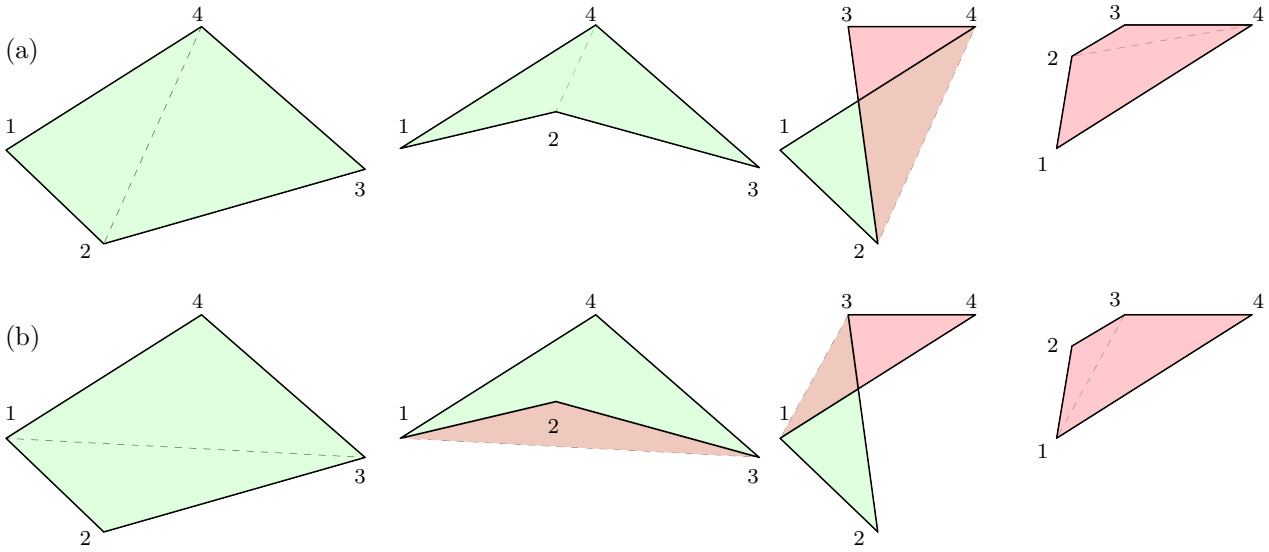

Figure 4 *Self-intersection detection for quadrilaterals. We triangulate the quadrilateral in two ways (rows (a) and (b)) and compute the signed area of the triangles. The dashed lines represent the quadrilateral's triangulation. Checking that at least one triangulation produces two positively oriented triangles ensures the transformed quadrilateral to be non self-intersecting.*

where the sum is taken over the boundary joints, w_{bnd} enables tuning the importance given to that term, and $\bar{l}_0$ is the previously defined length scale.

5.3 Constraints

We first constrain the deployed joints variables (u_i, v_i) to lie within the range $[0, 1]$. Second, the quadrilaterals should not self-intersect in the flat state. For that, we triangulate the quadrilaterals in two ways and compute the signed areas of the triangles, see Figure 4. We requires that at least one of the two triangulations of a quadrilateral q has two positively oriented triangles:

$$\max(\min(\text{SA}_{q,412}, \text{SA}_{q,234}), \min(\text{SA}_{q,123}, \text{SA}_{q,341})) > \epsilon_{\text{area}}|S|, \tag{A1}$$

where $\text{SA}_{q,ijk}$ computes the signed area of the triangle of vertices i, j, k in quadrilateral q, and $|S|$ is the area of the target surface. ϵ_{area} determines how far we want the quadrilaterals from being self-intersecting.

6 Design Optimization

Our design optimization consists in minimizing the function $\bar{J}$ of the curves degrees of freedom and average opening angle $\bar{\mathbf{q}}$

$$\bar{J}(\bar{\mathbf{q}}) := J\left(\mathbf{q}_{\text{ip}}(\mathbf{q}), \bar{\mathbf{p}}(\bar{\mathbf{q}})\right), \quad J(\mathbf{q}_{\text{ip}}, \bar{\mathbf{p}}) := \frac{1}{E_0} E(\bar{\mathbf{x}}^*_{\text{3D}}(\bar{\mathbf{p}}), \mathbf{p}) + T_t(\bar{\mathbf{x}}^*_{\text{3D}}(\bar{\mathbf{p}})) + R(\mathbf{q}_{\text{ip}}, \mathbf{p}), \tag{A2}$$

where $\mathbf{q}_{\text{ip}}$ are the intermediate interpolated points constructed from the joints positiions and orthogonal offsets stored in $\mathbf{q}$.

6.1 Fairness Regularization

The last term in the definition of J is a regularization that we split into two parts

$$R(\mathbf{q}_{\text{ip}}, \mathbf{p}) := \underbrace{\frac{w_{\text{ip}}}{2l_0^2}\|\mathbf{L}_{\text{ip}}\mathbf{q}_{\text{ip}} - \mathbf{q}_{\text{ip,f}}\|_2^2}_{w_{\text{ip}}R_{\text{ip}}(\mathbf{q}_{\text{ip}})} + \underbrace{\frac{w_\kappa}{2\bar{\kappa}_0}\mathbf{p}_\kappa^\mathsf{T}\mathbf{L}\mathbf{p}_\kappa}_{w_\kappa R_\kappa(\mathbf{p})}. \tag{A3}$$

The first term is a fairness term that measures the Dirichlet energy of the directed graph of interpolated points. We build this graph by first connecting the interpolated points $\mathbf{q}_{\text{ip}}$ and their neighbors along each curve from the curve linkage. Then, we remove the edge going from inner interpolated points to either one of each curve endpoints. This prevents the boundary from shrinking. Interpolated points with no inward pointing edges are labeled as *fixed* and are stored in the constant vector $\mathbf{q}_{\text{ip,f}} \in \mathbb{R}^{2n_{\text{ip}}}$. This vector also contains 0s wherever the associated interpolated point is not fixed. The user may remove edges to fix additional interpolated points. $\mathbf{L}_{\text{ip}}$ is the uniform Laplacian matrix associated to the interpolated points graph.

The second term penalizes the deviation of the rest curvature at a vertex to the average of its two neighboring vertices along each rod. We use the average of the rest lengths of the rod segments $\bar{\kappa}_0$ to scale the regularization term. We define $\mathbf{p}_\kappa$ to extract the rest curvatures from the vector of rest quantities $\mathbf{p}$, and $\mathbf{L}$ as the 1D uniform Laplacian matrix.

The coefficients w_{ip} and w_κ control the importance given to each of the two terms.

6.2 Gradients and Hessian-Vector Products

The objective $\bar{J}$ in Equation (A2) is minimized using an off-the-shelf Sequential Linear-Quadratic Programming (SLQP) algorithm provided by Knitro Waltz and Nocedal (2004). The EQP part is solved using a Newton-CG like approach, which requires computing gradients and Hessian-Vector Products (HVPs) of the objective with respect to the curves degrees of freedom $\bar{\mathbf{q}}$. This vector also includes the target average opening angle $\bar{\alpha}_{\mathbf{q}}$. We first expose the gradients and HVPs with respect to the rest quantities $\mathbf{p}$, and $\bar{\alpha}_{\mathbf{p}}$. Then, we explain how they relate to the same quantities expressed with respect to design parameters.

6.2.1 Rest Quantities

We define an objective function that depends on the interpolated points, the deployed degrees of freedom, and the rest quantities as

$$\tilde{J}(\mathbf{q}_{\text{ip}}, \bar{\mathbf{x}}_{\text{3D}}, \bar{\mathbf{p}}) := \frac{1}{E_0}E(\bar{\mathbf{x}}_{\text{3D}}, \mathbf{p}) + T_t(\mathbf{x}_{\text{3D}}) + R(\mathbf{q}_{\text{ip}}, \mathbf{p}).$$

The previously defined objective as a function of the rest quantities can be written as $J(\mathbf{q}_{\text{ip}}, \bar{\mathbf{p}}) = \tilde{J}(\mathbf{q}_{\text{ip}}, \bar{\mathbf{x}}^*_{\text{3D}}(\bar{\mathbf{p}}), \bar{\mathbf{p}})$. We express the first-order KKT conditions for the deployed equilibrium problem $\bar{\mathbf{x}}^*_{\text{3D}}(\bar{\mathbf{p}}) := \text{argmin}_{\bar{\mathbf{x}}}\, \bar{E}(\bar{\mathbf{x}}, \mathbf{p})$ subject to $\bar{\alpha}_{\mathbf{x}} = \bar{\alpha}_{\mathbf{p}}$

$$\begin{cases} \left.\dfrac{\partial \bar{E}}{\partial \bar{\mathbf{x}}}\right|_{\bar{\mathbf{x}}^*_{\text{3D}}(\bar{\mathbf{p}}), \bar{\mathbf{p}}} + \lambda(\bar{\mathbf{p}})\mathbf{e}_{\bar{\alpha}}^\mathsf{T} = \mathbf{0} \\ \bar{\alpha}^*_{\mathbf{x}}(\bar{\mathbf{p}}) = \bar{\alpha}_{\mathbf{p}}, \end{cases}$$

where $\lambda(\bar{\mathbf{p}})$ is the Lagrange multiplier of the equality constrained minimization problem. Differentiating gives the KKT system shown below

$$\begin{cases} \mathbf{H}_{3\mathrm{D}} \dfrac{\partial \mathbf{x}^*_{3\mathrm{D}}}{\partial \bar{\mathbf{p}}} = -\begin{bmatrix} \dfrac{\partial^2 \bar{E}}{\partial \mathbf{x} \partial \mathbf{p}} & \dfrac{\partial^2 \bar{E}}{\partial \mathbf{x} \partial \bar{\alpha}_{\mathbf{x}}} \end{bmatrix} \\ \dfrac{\partial^2 \bar{E}}{\partial \bar{\alpha}_{\mathbf{x}} \partial \mathbf{x}} \dfrac{\partial \mathbf{x}^*_{3\mathrm{D}}}{\partial \bar{\mathbf{p}}} + \begin{bmatrix} \dfrac{\partial^2 \bar{E}}{\partial \bar{\alpha}_{\mathbf{x}} \partial \mathbf{p}} & \dfrac{\partial^2 \bar{E}}{\partial \bar{\alpha}_{\mathbf{x}} \partial \bar{\alpha}_{\mathbf{x}}} \end{bmatrix} = -\dfrac{\partial \lambda}{\partial \bar{\mathbf{p}}} \\ \dfrac{\partial \bar{\alpha}^*_{\mathbf{x}}}{\partial \mathbf{p}} = \mathbf{0} \\ \dfrac{\partial \bar{\alpha}^*_{\mathbf{x}}}{\partial \bar{\alpha}_{\mathbf{p}}} = 1, \end{cases} \tag{A4}$$

where $\mathbf{H}_{3\mathrm{D}} := \frac{\partial^2 \bar{E}}{\partial \mathbf{x}^2}(\bar{\mathbf{x}}^*_{3\mathrm{D}}(\bar{\mathbf{p}}), \mathbf{p})$ is the Hessian of the elastic energy and surface attraction term of the deployed linkage with respect to the rod linkage degrees of freedom, and $\bar{\mathbf{p}} := [\mathbf{p}^\mathsf{T} \ \bar{\alpha}_t]^\mathsf{T}$. We then define the adjoint state vector $\mathbf{w}$ as

$$\mathbf{H}_{3\mathrm{D}} \mathbf{w} = \frac{\partial \tilde{J}}{\partial \mathbf{x}}^\mathsf{T},$$

where the gradient $\frac{\partial \tilde{J}}{\partial \mathbf{x}}$ can be computed analytically using the elastic rods model, see Panetta et al. (2019) for details. The overall gradient of the objective J with respect to the rest quantities can be written as

$$\frac{\partial J}{\partial \bar{\mathbf{p}}} = \mathbf{w}^\mathsf{T} \begin{bmatrix} \dfrac{\partial^2 \bar{E}}{\partial \mathbf{x} \partial \mathbf{p}} & \dfrac{\partial^2 \bar{E}}{\partial \mathbf{x} \partial \bar{\alpha}_{\mathbf{x}}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \dfrac{\partial \tilde{J}}{\partial \bar{\alpha}_{\mathbf{x}}} \end{bmatrix} + \frac{\partial \tilde{J}}{\partial \bar{\mathbf{p}}}.$$

The derivative of the objective J with respect to the interpolated points is computed analytically as $\frac{\partial J}{\partial \mathbf{q}_{\mathrm{ip}}} = \beta_{\mathrm{ip}} \frac{\partial R_{\mathrm{ip}}}{\partial \mathbf{q}_{\mathrm{ip}}}$.

We now compute the HVP for J along a given direction $\delta\bar{\mathbf{p}}$. First, we write the KKT system in Equation (A4) along that direction to obtain

$$\delta \bar{\mathbf{x}}^*_{3\mathrm{D}} = \begin{bmatrix} -\mathbf{H}^{-1}_{3\mathrm{D}} \left(\dfrac{\partial^2 \bar{E}}{\partial \mathbf{x} \partial \mathbf{p}} \delta \mathbf{p} + \dfrac{\partial^2 \bar{E}}{\partial \mathbf{x} \partial \bar{\alpha}_{\mathbf{x}}} \delta \bar{\alpha}_{\mathbf{p}} \right) \\ \delta \bar{\alpha}_{\mathbf{p}} \end{bmatrix}, \tag{A5}$$

where we define for short $\delta \bar{\mathbf{x}}^*_{3\mathrm{D}} := \frac{\partial \bar{\mathbf{x}}^*_{3\mathrm{D}}}{\partial \bar{\mathbf{p}}} \delta \bar{\mathbf{p}}$. Similarly, the perturbation of the adjoint state vector $\delta \mathbf{w} := \frac{\partial \mathbf{w}}{\partial \bar{\mathbf{p}}} \delta \bar{\mathbf{p}}$ must satisfy the linear system given by

$$\mathbf{H}_{3\mathrm{D}} \delta \mathbf{w} = \frac{\partial^2 \tilde{J}}{\partial \mathbf{x} \partial \bar{\mathbf{x}}} \delta \bar{\mathbf{x}}^*_{3\mathrm{D}} + \frac{\partial^2 \tilde{J}}{\partial \mathbf{x} \partial \mathbf{p}} \delta \mathbf{p} - \left(\frac{\partial^3 \bar{E}}{\partial \mathbf{p} \partial \mathbf{x} \partial \bar{\mathbf{x}}} \delta \bar{\mathbf{x}}^*_{3\mathrm{D}} + \frac{\partial^3 \bar{E}}{\partial \mathbf{p} \partial \mathbf{x} \partial \mathbf{p}} \delta \mathbf{p} \right) \mathbf{w}.$$

The full HVP can then be written as

$$\begin{aligned} \frac{\partial^2 J}{\partial \bar{\mathbf{p}}^2} \delta \bar{\mathbf{p}} = &- \begin{bmatrix} \dfrac{\partial^2 \bar{E}}{\partial \mathbf{p} \partial \mathbf{x}} \\ \dfrac{\partial^2 \bar{E}}{\partial \bar{\alpha}_{\mathbf{x}} \partial \mathbf{x}} \end{bmatrix} \delta \mathbf{w} - \begin{bmatrix} \dfrac{\partial^3 \bar{E}}{\partial \mathbf{p} \partial \mathbf{x} \partial \bar{\mathbf{x}}} \delta \bar{\mathbf{x}}^*_{3\mathrm{D}} + \dfrac{\partial^3 \bar{E}}{\partial \mathbf{p} \partial \mathbf{x} \partial \mathbf{p}} \delta \mathbf{p} \\ \dfrac{\partial^3 \bar{E}}{\partial \bar{\alpha}_{\mathbf{x}} \partial \mathbf{x} \partial \bar{\mathbf{x}}} \delta \bar{\mathbf{x}}^*_{3\mathrm{D}} + \dfrac{\partial^3 \bar{E}}{\partial \bar{\alpha}_{\mathbf{x}} \partial \mathbf{x} \partial \mathbf{p}} \delta \mathbf{p} \end{bmatrix} \mathbf{w} \\ &+ \begin{bmatrix} \mathbf{0} \\ \dfrac{\partial^2 \tilde{J}}{\partial \bar{\alpha}_{\mathbf{x}} \partial \bar{\mathbf{x}}} \delta \bar{\mathbf{x}}^*_{3\mathrm{D}} + \dfrac{\partial^2 \tilde{J}}{\partial \bar{\alpha}_{\mathbf{x}} \partial \bar{\mathbf{p}}} \delta \bar{\mathbf{p}} \end{bmatrix} \end{aligned} \tag{A6}$$

$$+ \frac{\partial^2 \tilde{J}}{\partial \bar{\mathbf{p}} \partial \bar{\mathbf{x}}} \delta \bar{\mathbf{x}}^*_{3\mathrm{D}} + \frac{\partial^2 \tilde{J}}{\partial \bar{\mathbf{p}} \partial \bar{\mathbf{p}}} \delta \bar{\mathbf{p}}.$$

The Hessian of J with respect to the control points is simply given by $\beta_{\mathrm{ip}}/l_0^2 \ \mathbf{L}_{\mathrm{ip}}^{\mathsf{T}} \mathbf{L}_{\mathrm{ip}}$. Third order derivatives of the energy are obtained through automatic differentiation as explained in Panetta et al. (2019).

6.2.2 Design Parameters

Gradients of $\bar{J}(\bar{\mathbf{q}})$ can be obtained by backpropagating the gradients of $J(\mathbf{q}_{\mathrm{cp},\bar{\mathbf{p}}})$. In practice, this consists in computing the Vector-Jacobian Product (VJP) and it is performed efficiently using reverse-mode automatic differentiation.

Second order derivatives are more expensive to obtain using automatic differentiation since the computation graph must be traversed multiple times. While constructing the full Hessian $\frac{\partial^2 \bar{J}}{\partial \bar{\mathbf{q}}^2}$ is prohibited, HVPs can be computed using the previously introduced HVPs as

$$\begin{aligned} \frac{\partial^2 \bar{J}}{\partial \bar{\mathbf{q}}^2} \delta \bar{\mathbf{q}} = \frac{\partial \bar{\mathbf{p}}}{\partial \bar{\mathbf{q}}}^{\mathsf{T}} \frac{\partial^2 J}{\partial \bar{\mathbf{p}}^2} \frac{\partial \bar{\mathbf{p}}}{\partial \bar{\mathbf{q}}} \delta \bar{\mathbf{q}} + \frac{\partial}{\partial \bar{\mathbf{q}}} \left(\frac{\partial}{\partial \bar{\mathbf{q}}} \left(\mathbf{g}_{\bar{\mathbf{p}}}^{\mathsf{T}} \cdot \bar{\mathbf{p}} \right) \cdot \delta \bar{\mathbf{q}} \right) + \frac{\partial \mathbf{q}_{\mathrm{ip}}}{\partial \bar{\mathbf{q}}}^{\mathsf{T}} \frac{\partial^2 J}{\partial \mathbf{q}_{\mathrm{ip}}^2} \frac{\partial \mathbf{q}_{\mathrm{ip}}}{\partial \bar{\mathbf{q}}} \delta \bar{\mathbf{q}} \\ + \frac{\partial}{\partial \bar{\mathbf{q}}} \left(\frac{\partial}{\partial \bar{\mathbf{q}}} \left(\mathbf{g}_{\mathbf{q}_{\mathrm{ip}}}^{\mathsf{T}} \cdot \mathbf{q}_{\mathrm{ip}} \right) \cdot \delta \bar{\mathbf{q}} \right), \end{aligned} \tag{A7}$$

where we define the **constant** vectors equal to the current gradients of the objective $\mathbf{g}_{\bar{\mathbf{p}}}^{\mathsf{T}} := \partial J / \partial \bar{\mathbf{p}}$ and $\mathbf{g}_{\mathbf{q}_{\mathrm{ip}}}^{\mathsf{T}} := \partial J / \partial \mathbf{q}_{\mathrm{ip}}$.

6.3 Extended Rest Quantities Update

When updating the extended rest quantities by an increment $\delta \bar{\mathbf{p}}$, we estimate the change in the deployed state configuration by truncating the following Taylor expansion up to the second order

$$\bar{\mathbf{x}}^*_{3\mathrm{D}}(\bar{\mathbf{p}} + \delta \bar{\mathbf{p}}) = \bar{\mathbf{x}}^*_{3\mathrm{D}}(\bar{\mathbf{p}}) + \frac{\partial \bar{\mathbf{x}}^*_{3\mathrm{D}}}{\partial \bar{\mathbf{p}}} \delta \bar{\mathbf{p}} + \delta \bar{\mathbf{p}}^{\mathsf{T}} \frac{\partial^2 \bar{\mathbf{x}}^*_{3\mathrm{D}}}{\partial \bar{\mathbf{p}}^2} \delta \bar{\mathbf{p}} + \mathcal{O}(\|\delta \bar{\mathbf{p}}\|^3).$$

The first-order modification of the degrees of freedom can be obtained by solving the system shown in Equation (A5). The second order perturbation of the deployed degrees of freedom at equilibrium can be obtained by differentiating Equation (A5) as shown below:

$$\begin{cases} \mathbf{H}_{3\mathrm{D}} \delta \bar{\mathbf{p}} \dfrac{\partial^2 \mathbf{x}^*_{3\mathrm{D}}}{\partial \bar{\mathbf{p}}^2} \delta \bar{\mathbf{p}} &= -\left[\dfrac{\partial^3 \bar{E}}{\partial \mathbf{x} \partial \bar{\mathbf{x}} \partial \bar{\mathbf{x}}} \delta \bar{\mathbf{x}}^*_{3\mathrm{D}} + \dfrac{\partial^3 \bar{E}}{\partial \mathbf{x} \partial \bar{\mathbf{x}} \partial \mathbf{p}} \delta \mathbf{p} \right] \delta \bar{\mathbf{x}}^*_{3\mathrm{D}} \quad - \left[\dfrac{\partial^3 \bar{E}}{\partial \mathbf{x} \partial \mathbf{p} \partial \bar{\mathbf{x}}} \delta \bar{\mathbf{x}}^*_{3\mathrm{D}} + \dfrac{\partial^3 \bar{E}}{\partial \mathbf{x} \partial \mathbf{p} \partial \mathbf{p}} \delta \mathbf{p} \right] \delta \mathbf{p} \\ \delta \bar{\mathbf{p}} \dfrac{\partial^2 \bar{\alpha}^*_{\mathbf{x}}}{\partial \bar{\mathbf{p}}^2} \delta \bar{\mathbf{p}} &= 0. \end{cases}$$

References

Bergou, M., Audoly, B., Vouga, E., Wardetzky, M., and Grinspun, E. (2010). Discrete viscous threads. *ACM Transactions on graphics (TOG)*, 29(4):1–10.

Bergou, M., Wardetzky, M., Robinson, S., Audoly, B., and Grinspun, E. (2008). Discrete elastic rods. In *ACM SIGGRAPH 2008 papers*, pages 1–12.

Hall, C. A. and Meyer, W. W. (1976). Optimal error bounds for cubic spline interpolation. *Journal of Approximation Theory*, 16(2):105–122.

Landau, L., Lifshitï¸ sï¸¡, E., Kosevich, A., Lifshitz, E., and Pitaevskii, L. (1986). *Theory of Elasticity: Volume 7*. Course of theoretical physics. Elsevier Science.

Megaro, V., Zehnder, J., Bächer, M., Coros, S., Gross, M. H., and Thomaszewski, B. (2017). A computational design tool for compliant mechanisms. *ACM Trans. Graph.*, 36(4):82–1.

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.

Waltz, R. A. and Nocedal, J. (2004). Knitro 2.0 user's manual. *Ziena Optimization, Inc.[en ligne] disponible sur http://www. ziena. com (September, 2010)*, 7:33–34.