Supplementary Material for "Curl Quantization for Automatic Placement of Knit Singularities"

1 AUXILIARY RESULTS

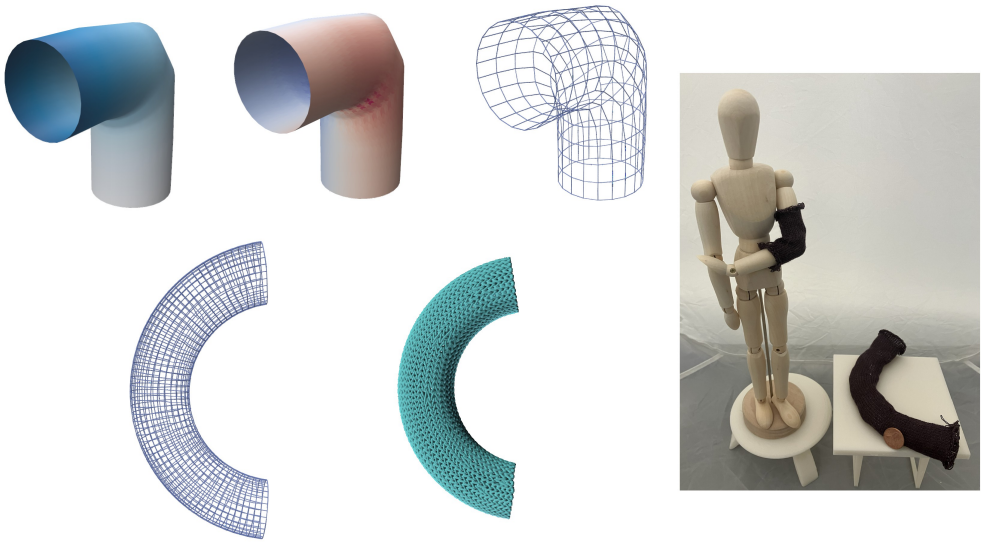

Fig. 1. Top (left to right): Elbow sleeve with time function, curl signal and knit graph. Bottom (left to right): Knit graph of bent cylinder from Fig. 3 and render. Right: Fabricated sleeve (on 12 in mannequin) and bent cylinder (with 0.75 inch coin).

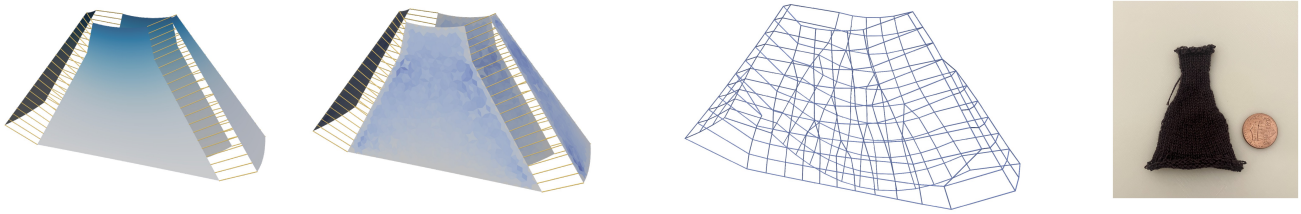

Fig. 2. Four-panel skirt. From left to right: Time function, wale curl signal, knit graph visualized with vertices on the panels. Fabricated result (with 0.75 inch coin).

2 LINEAR CONSTRAINTS

Recall that the primary problem we aim to solve is given by Eq. (2) in the main text, subject to several additional linear constraints briefly described in §4.3 of the main text. As noted there, all of the path integrals below are discretized as simple finite sums of oriented 1-form values along edges. Here we detail these constraints as they relate to the machine-knittability of knit graphs, and the notation is borrowed from the main text.

•

(1) Eq. (4a), Helix-free condition:

$$\int_{\gamma_i} \sigma_{c,s} = 0 \qquad i = 1, \ldots, n_{\text{pairs}}$$

Knit graphs are required to be helix-free, meaning that vertices making up a course row cannot have a sequence of wale edges going between them. This is a bit paradoxical at first glance, as the yarn path in a machine knit will ultimately have to helix, so that stitches may be made on top of existing course rows. However, it is challenging to specify the exact amount of helicing needed. This is solved by the Autoknit scheduler [Narayanan et al. 2018] via a post-processing *tracing* step, which applies a sequence of rules and visits every node of the knit graph twice to generate such an appropriate yarn path. The helix-free knit graph condition is a strong manufacturing constraint that ensures that the scheduler achieves this behavior, even in the case of short-row ends.

(2) Eq. (4b) Morse decomposition:

$$\int_{\mu_i} \sigma_{c,s} = 0 \qquad i = 1, \ldots, n_{\text{Morse}}$$

Eq. (6) Homology generators:

$$\int_{g_l} \sigma = \mathbf{k}_g \qquad l = 1, \ldots, 2g + n_{\text{bdy}} - 1$$

To handle richer topologies (greater than 2 boundaries), we decompose the model, in a Morse sense, into cylinders along critical level sets of the time function. Just as in Mitra et al. [2024], we constrain the integral of the 1-form σ to be 0 along these levels sets and only generate singularity pairs within each cylindrical component. We also ask that the integral of σ along homology generators be an integer multiple of the period. We construct these generators using the tree-cotree method of Dłotko [2012]. These generator constraints are required to guarantee that we have a well-defined striping function $M \to \mathbb{S}^1$ even in the nontrivial homology. Fig. 3 below highlights the import of these constraints.

(3) Eq. (4c), Boundary Alignment:

$$\sigma_{c,s}|_{\partial M} = 0$$

This ensures that the boundary cycles of the knit graph align with the boundaries of the input model. This is especially useful when the input model may contain jagged or non-uniform boundaries, such as Fig. 6(2) of the main text.

(4) Eq. (4d), User-specified alignment:

$$\sigma_s|_{l_j} = 0 \qquad j = 1, \ldots, n_{\text{align}}$$

This simply generalizes constraint (4c) to arbitrary edges in the model that may not be in the starting / ending cycles of knitting. As shown in Fig. 9 from the main text, this finds use in the cut-and-sew setting where a designer may want wale edges of the knit graph to line up with the patch boundaries.

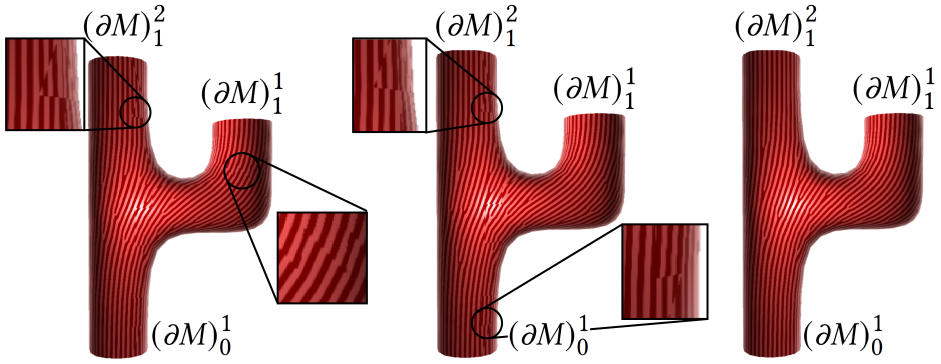

Fig. 3. Denote ∂M to be the boundary of the manifold. Let us use $(\partial M)_0 = \sqcup_{i=1}^{N_0} (\partial M)_0^i$ and $(\partial M)_1 = \sqcup_{i=1}^{N_1} (\partial M)_1^i$ to label those boundaries and components on which the time function is set to 0 and 1, respectively. The total number of boundary components is $N = N_0 + N_1$. Left: No homology constraints cause stripe discontinuities around all boundaries. Middle: Only constraining the generator around $(\partial M)_1^1$ cause discontinuities near $(\partial M)_1^2$ and $(\partial M)_0^1$. Right: Enforcing the homology constraints on $n_{bdy} - 1$ boundaries ensures no stripe discontinuities. Figure from Fig. 8 [Mitra et al. 2023] (borrowed with author's permission).

3 EDGE LENGTH ERROR HISTOGRAMS

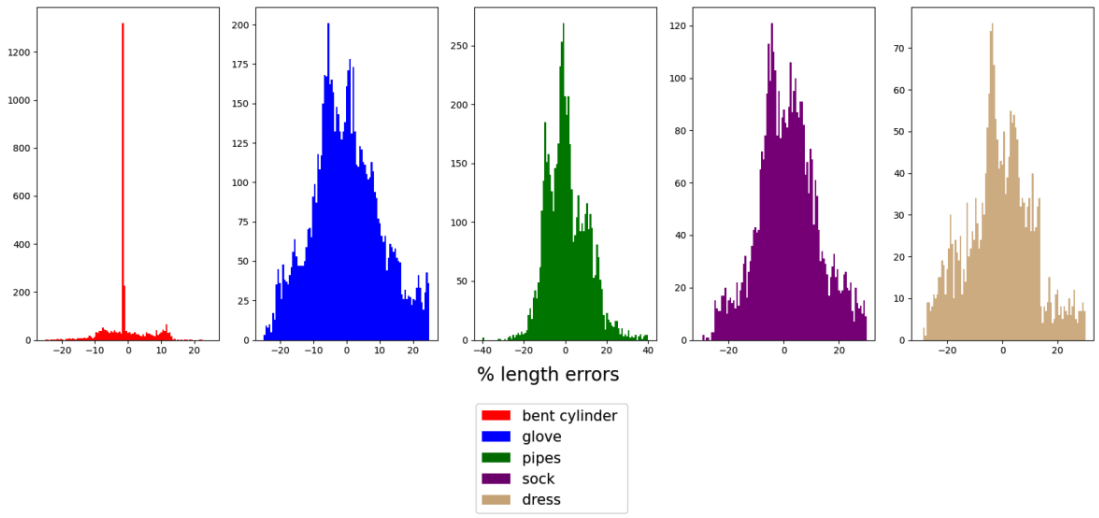

Fig. 4. Edge length histograms for fabricated models. As is the case with Autoknit [Narayanan et al. 2018], our edge length errors are $< 10\%$. While our histograms don't peak near 0, we have fewer outliers than Narayanan et al. [2018] due to the use our L^2 objective.

.

4 STRIPE PATTERN TRACING AND KNIT GRAPH GENERATION

Having solved for both course and wale 1-forms, σ_c and σ_w respectively, we trace the generated stripe patterns to produce the knit graph. Our 1-forms are integrated along a spanning tree of mesh edges and the integrated values, α and β are stored modulo the period, P at every corner i.e., base of a halfedge, in a face. Crucially, because our singularities lie on edges and not on faces, σ_c and σ_w are nonsingular at all faces. Thus, for face f_{ijk}, the stripe texturing coordinates are simply linearly interpolated to find course and wale intersections. In particular, we solve the linear system,

$$\begin{aligned} \alpha_i b_i + \alpha_j b_j + \alpha_k b_k &= Z_\alpha \\ \beta_i b_i + \beta_j b_j + \beta_k b_k &= Z_\beta \end{aligned} \tag{1}$$

where b_i, b_j and b_k are barycentric coordinates of f_{ijk} and Z_α and Z_β are the stripe level sets passing through f_{ijk}. This allows for more robust tracing and obviates the need for the effective interpolant construction of Mitra et al. [2024] or the use of a non-linear interpolant [Knöppel et al. 2015]. Additionally, we simplify tracing of the course rows by using a mesh with average edge length smaller than the period. This ensures that there is only one choice of level set born at a singular edge. Much like in Mitra et al. [2024, 2023], to connect vertices across faces, we create virtual vertices on triangle borders by computing the intersection of the stripe level set with the triangle border. These virtual vertices are then connected to the interior knit graph vertices.

5 FOLIATION GUARANTEES

As in Mitra et al. [2024], we are able to guarantee non-helicing of all level sets of our course stripe texturing function (and not just the level sets that are used to trace course rows). For detailed definitions of foliations concepts we refer to their paper. This non-helicing of all level sets is done by joining saddle points of the resulting foliation with our path constraints. These ensure that certain separatrices join the saddles, and that all level sets starting at one short row end, end up at the other short row end that it is joined to via a non-helicing path constraint. Figure 5 (also in the main text) illustrates the similarity in foliation structure near a +1 singularity placed on a triangle face, and on an edge, or bigon face in the lens complex.

Fig. 5. Reproduction of Fig. 4 from main text. Foliation behavior near a +1 singularity, showing a source (red) and saddle (yellow) in close proximity, with level sets emanating to the right. In the left and middle images, separatrices are orange. Left: Singular foliation structure as in Knöppel et al. [2015] (from Mitra et al. [2024]). Middle: We place singularities on edges. Right: Non-helicing path constraint visualized in blue for a positive singular edge.

We characterize the foliation structure in the vicinity of a course singular edge. For this we recall the definition of *index* of a discrete 1-form σ on a polyhedral mesh defined in Gortler et al. [2006]. At a vertex v:

$$\mathrm{ind}_\sigma(v) := \frac{2 - \mathrm{sc}(v)}{2}$$

In the above, $\mathrm{sc}(v)$ denotes the number of sign changes in the sign of σ as one traverses the outgoing half-edges in cyclic order. An analogous definition of index for faces is specified, with $\mathrm{sc}(f)$ denoting the number of times the sign of σ changes as one traverses the cycle of half-edges circulating CCW about the face.

Let us consider the neighborhood of a singular edge and for simplicity, we assume that we are looking at a singular edge with index +1, reflected in the schematic figure below. As in the text, we let $e_{i^+ j^+}$ denote the

singular edge and assume $i^+ j^+$ is the direction more aligned with ∇h. Let k denote the vertex opposite halfedge $j^+ i^+$, and we use N^+ to denote the illustrated neighborhood of $e_{i^+ j^+}$.

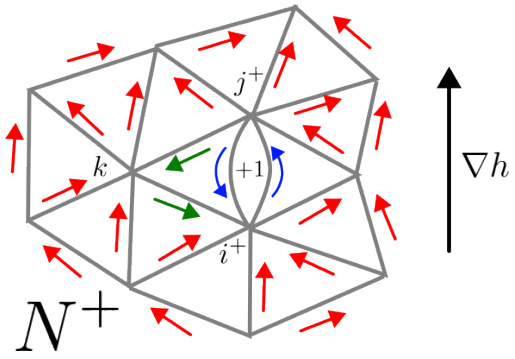

Fig. 6. Notation, and visual proof of Prop. 1. Colored arrows denote the direction of increasing σ over the edges and halfedges. See proof text for more.

Proposition 1. *In a neighborhood N^+ of a +1 singular edge $e_{i^+ j^+}$, let us assume that*

$$\text{sign}(\sigma_{lm}) = \text{sign}(h(m) - h(l)) \qquad \text{for all halfedges } lm \text{ not in triangle } i^+ j^+ k \tag{2}$$

Then three inequality constraints:

$$\text{sign}(\sigma_{i^+ j^+}) = \text{sign}(\sigma_{j^+ i^+}) \tag{3a}$$

$$\text{sign}(\sigma_{j^+ k}) > 0 \tag{3b}$$

$$\text{sign}(\sigma_{k i^+}) > 0 \tag{3c}$$

imply $\text{ind}_\sigma(k) = -1$, $\text{ind}_\sigma(i^+) = 0$, *and* $\text{ind}_\sigma(j^+) = 0$.

Proof. The argument is made visually via the arrows in Fig. 6, which denote directions of positive σ on edges and halfedges. Eq. (2) implies the red arrows. Eq. (3a) along with the placement of a singular edge on $e^{i^+ j^+}$ implies the blue arrows. Eqs. (3b), (3c) imply the green arrows. Indices of i^+, j^+, and k can be inferred via simple counting and the index definition. □

Lastly, we have the following:

Lemma 1. *If* $\text{ind}_\sigma(v) = -1$ *and* $d_1^\Delta \sigma = 0$ *on all faces containing* v, *then* v *is a saddle vertex of the foliation induced by* σ.

Proof. We again argue by sketch. See Fig. 7 for an illustration of a index -1 vertex. Due to the fact that $d_1^\Delta \sigma = 0$, the level sets of σ, i.e., the leaves of the foliation, are linear over the nearby triangles. Thus, whenever one has a sign change there is a particular level set which forms a separatrix coming into or out of v. With four sign changes, we can see that the 1-ring of v is split into four hyperbolic sectors. Thus, v is a saddle vertex of the foliation. □

The above arguments justify our use of the particular path constraints that we select, i.e., those that connect k with the analogous vertex near the -1 singularity. Analogous results hold near the -1 singularity. Within our optimization framework, $d_1^\Delta \sigma = 0$ on all triangular faces, and assumption (2) is strongly motivated by our objective, which aims to align $\delta\sigma$ with $\overline{\nabla h}$. Lastly, we motivate satisfaction of (3a), (3b), and (3c) by choosing

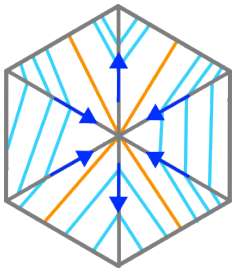

Fig. 7. Visual proof of Lemma 1. Blue arrows denote direction of positive σ over halfedges, while cyan denotes level sets of σ, and orange lines denote separatrices. See proof text for more.

singular edges that are well aligned with ∇h. Empirically, we found that we are able to robustly join separatrices and prevent helicing of *any* level sets via these path constraints. If guaranteed robustness is desired, these inequality constraints can be incorporated into the optimizations, at the cost of some computational speed.

Lastly, we note the practical improvement upon the corresponding constraints with face-based singularities. These required us to fix the form over singular triangles in a post-processing optimization, and to match particular points on the edges of singular triangles. This was due to the chicken-and-egg problem of not being able to know separatrices without knowing the form that one is optimizing for (and which must be held to the path constraints). Here, we may simply match vertices, and can avoid any freezing of the 1-form σ, and any similar circular issues.

REFERENCES

Paweł Dłotko. 2012. A fast algorithm to compute cohomology group generators of orientable 2-manifolds. *Pattern Recognition Letters* 33, 11 (2012), 1468–1476.

Steven J. Gortler, Craig Gotsman, and Dylan Thurston. 2006. Discrete one-forms on meshes and applications to 3D mesh parameterization. *Computer Aided Geometric Design* 23, 2 (2006), 83–112.

Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. 2015. Stripe patterns on surfaces. *ACM Trans. Graph.* 34, 4, Article 39 (July 2015), 11 pages.

Rahul Mitra, Erick Jimenez Berumen, Megan Hofmann, and Edward Chien. 2024. Singular Foliations for Knit Graph Design. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) *(SIGGRAPH '24)*. Association for Computing Machinery, New York, NY, USA, Article 38, 11 pages.

Rahul Mitra, Liane Makatura, Emily Whiting, and Edward Chien. 2023. Helix-Free Stripes for Knit Graph Design. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 75, 9 pages.

Vidya Narayanan, Lea Albaugh, Jessica Hodgins, Stelian Coros, and James Mccann. 2018. Automatic Machine Knitting of 3D Meshes. *ACM Trans. Graph.* 37, 3, Article 35 (Aug. 2018), 15 pages.