

# Medial Sphere Preconditioning for Knot Untangling and Volume-Filling Curves

YUTA NOMA, University of Toronto, Canada

ALEC JACOBSON, University of Toronto and Adobe Research, Canada

KARAN SINGH, University of Toronto, Canada

Repulsive Curves [Yu et al. 2021]

230 seconds  
505 iterations

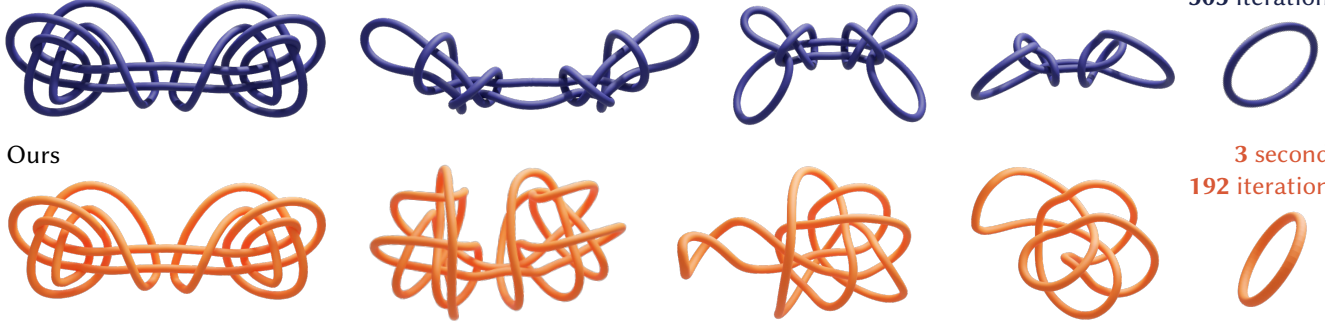


Fig. 1. Untangling the Freedman-He-Wang knot [1994] in 3 seconds. Our method uses a sparse volume-filling energy and a specialized preconditioner, leading to a 77x speedup compared to the Repulsive Curves [Yu et al. 2021b] algorithm that takes 230 seconds to untangle this curve.

We propose a fast, robust, and user-controllable algorithm for knot untangling and volume-filling curves. We extend prior work on surface-filling curves to the more challenging case of 3D volumes, equipped with a specialized gradient preconditioner that allows larger step sizes. Our method exhibits orders of magnitude faster runtime than existing methods. Our framework provides a whole new set of parameters to guide the shape of the curve, making it ideal for interactive design applications.

CCS Concepts: • **Computing methodologies** → **Mesh models**; **Mesh geometry models**; • **Human-centered computing** → Interactive systems and tools.

## ACM Reference Format:

Yuta Noma, Alec Jacobson, and Karan Singh. 2025. Medial Sphere Preconditioning for Knot Untangling and Volume-Filling Curves. In *SIGGRAPH Asia 2025 Conference Papers (SA Conference Papers '25)*, December 15–18, 2025, Hong Kong, Hong Kong. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3757377.3763856>

## 1 INTRODUCTION

Repulsion and attraction can play an essential role in computational design of curves. The *Repulsive Curves* algorithm [Yu et al. 2021b] recently demonstrated its power to pack a curve or collection of curves into a given volume, resulting in continuous space-filling curves

which are useful for trajectory planning and digital fabrication (Figure 2). The same flow can be utilized to untangle knots (Figure 1), suitable for visual understanding and computational design.

Unfortunately, the Repulsive Curves method is slow due to its dense linear solves per iteration, and often fails altogether. Meanwhile, Noma et al. [2024] defined an alternative curve flow to pack curves on a given surface mesh, but their method does not trivially extend to *volumetric* space-filling curves, and the optimization can be fragile (Figure 8).

We propose a fast, robust, and user-controllable geometric flow for knot untangling and volume-filling curves. We built on the recent *surface-filling* curve flow of Noma et al. [2024] and extended their energy to the more challenging *volume-filling* curve flows. We begin by reformulating their flow using the language of Sobolev spaces, and reveal that a 5x faster convergence can be achieved using our novel gradient preconditioning.

Each iteration of our flow only requires cheap operations like closest point queries and *sparse* linear solves. This leads to a massive speed up compared to Repulsive Curves, which requires *dense* linear solves (Figures 1, 3). Furthermore, our knot untangling algorithm is

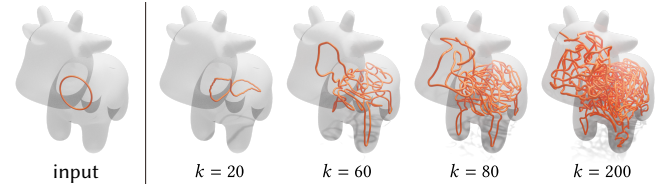


Fig. 2. We compute volume-filling curves through a geometric flow that finds an equilibrium of two sparse energies.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SA Conference Papers '25, December 15–18, 2025, Hong Kong, Hong Kong

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-2137-3/2025/12

<https://doi.org/10.1145/3757377.3763856>

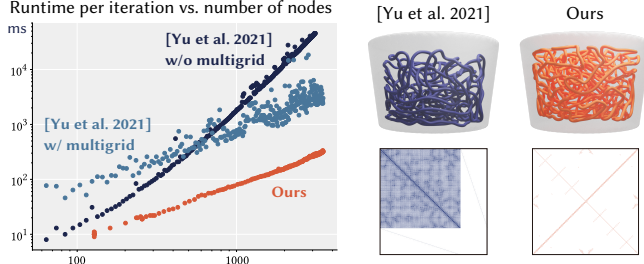


Fig. 3. Our method is orders of magnitude faster than Repulsive Curves [Yu et al. 2021b] (left) while creating qualitatively similar results (right). The difference is that their preconditioner is *dense* while ours is *sparse* (right bottom).

robust, showing a 100% success rate in untangling all 228 knots in the knot dataset [Yu et al. 2021b], in contrast to 87.7% (200/228) of Repulsive Curves. Not only is our method fast and robust, it also provides additional user control, such as the curve thickness (Figure 6) and the global shape of the curve (Figure 7).

We showcase the performance of our method by testing it on curves from various contexts, and extensively evaluate our algorithmic choices through ablation studies on different parameters.

## 2 RELATED WORK

Our method addresses two main applications: knot untangling and continuous space-filling curves. These two applications share similar requirements to keep the curve apart with a certain thickness (Figures 2, 4). Here, we cover relevant prior works regarding them, alongside a summary of works using gradient preconditioning in computer graphics.

### 2.1 Knot Untangling

Finding the simplest configuration of a closed curve without passing through itself is called the *knot untangling* problem. Solving this problem on a discrete graph cannot be done in polynomial time [de Mesmay et al. 2018], and the operations used in such literature (e.g., Reidemeister moves) are non-intuitive for non-mathematicians. Instead, continuously morphing a curve based on an energy minimization flow can be a good starting point for visual understanding and computational design.

Early methods employ repulsive potentials such as the so-called *symmetric energy* [Scharein 1998] or the *Möbius energy* [Freedman et al. 1994; Kusner and Sullivan 1998], but these energies require integration over every pair of points on a curve, leading to  $O(n^2)$  complexity to just integrate them. Alternatively, the *SONO* algorithm [Pieranski 1998] seeks convergence by iteratively tightening the knot, but this simple approach takes hours to converge [Yu et al. 2021b, Fig. 15]. Other lines of work simulate the knot-tying process [Brown et al. 2004; Harmon et al. 2012; Kubiak et al. 2007], untangle knots without optimizing the shape [Ladd and Kavraki 2004], or train robots to untangle open knots [Sundaresan et al. 2021], but none of them solve the problem of morphing closed curves to untangle them.

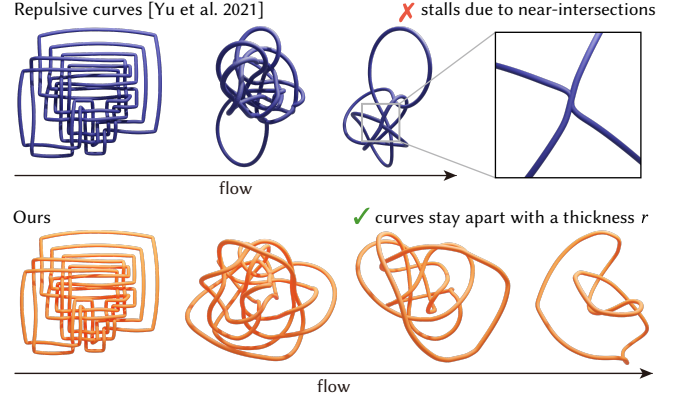


Fig. 4. Repulsive curves [Yu et al. 2021b] often stall when different parts of the curve get too close to one another. Our method naturally avoids this because the energy keeps the curve apart with a constant thickness  $r$ .

Recently, methods based on the *tangent point energy* [Bartels et al. 2017; Walker 2016; Yu et al. 2021a,b] exhibited state-of-the-art performance in this task. Especially, the Repulsive Curves algorithm by Yu et al. [2021b] achieved faster optimization of this energy, thanks to their fractional Sobolev preconditioner and Barnes-Hut approximation. However, the tangent point energy still needs to consider every pair of points within geometric proximity and is inherently dense. Although Yu et al. provide a hierarchical multigrid solver that reduces the degrees of freedom, they cannot escape from the fundamental need of dense linear solves per iteration.

We instead rely on a sparse and simple two-term energy that is fast to evaluate, leading to orders of magnitudes faster runtime.

### 2.2 Continuous Space-Filling Curves

Space-filling curves fill out a  $d$ -dimensional space with a single one-dimensional trajectory. While *discrete* space-filling curves only allow nodes to lie on the grid (e.g., [Hilbert 1891; Peano 1890]), we consider the problem of designing *continuous* space-filling curves where curve nodes can be placed arbitrarily and curves have smooth curvature (Figure 5). These patterns appear in many real-world applications, such as infill patterns for 3D printing [Liu et al. 2024; Zhang et al. 2022], knitting [Hirose et al. 2024], sensor placement [Sakura and Kakehi 2025], or robotic path planning [Yan and Mostofi

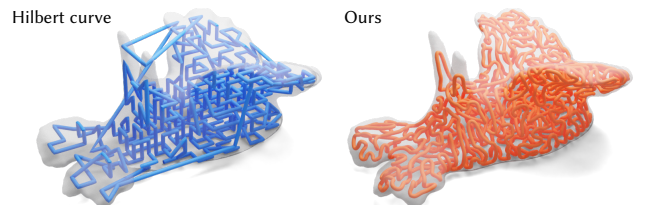


Fig. 5. Unlike Hilbert curves [Hilbert 1891], our curve fills out the space with a smooth, singly connected trajectory without protrusions beyond the target volume.

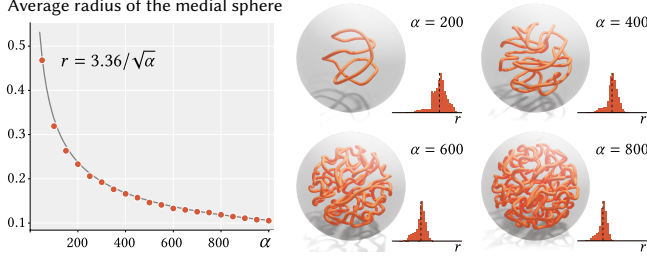


Fig. 6. The energy weight  $\alpha$  controls the thickness of the curve  $r$ , here computed as the average radius of the medial sphere. We find an empirical relation of  $r = 3.36/\sqrt{\alpha}$  through experiments.

2016]. These curves are also largely used for artistic creations [Pedersen and Singh 2006] and mathematical visualizations [Ashton et al. 2010; Pieranski 1998; Scharein 1998].

Most widely used algorithms compute such curves by initializing the curve based on a discrete algorithm and then relaxing it to a smooth one. For example, traditional slicer software (e.g., Cura [Ultimaker B.V. 2023]) slices the 3D volume into planes and places discrete space-filling curves on it. The recent advances in multi-axis 3D printing [Dai et al. 2018; Etienne et al. 2019; Fang et al. 2020; Liu et al. 2024; Zhang et al. 2022] have even allowed *curved* layers while satisfying the fabrication constraints. Other works find an initial Hamiltonian path from a finite set of points [Giannatsis et al. 2015], clusters [Yan and Mostofi 2016], grid cells [Cheng et al. 2020; Joshi et al. 2019], or polygonal elements [Bedel et al. 2022; Lin et al. 2019]. While these works exhibit excellent performance for specific tasks (e.g., 3D printing), they require complicated curve surgery to connect the curves together, and little can be said about the topology of the curve, such as connectivity and the knot group.

In contrast, we opt for employing a geometric flow that grows the curve from arbitrary initializations. A geometric flow preserves the local connectivity by definition (Figure 7), and it preserves the knot group robustly (Figure 12).

Despite their benefit, existing geometric flows that compute space-filling curves suffer from slow computation time [Yu et al. 2021b] or only support 2-manifolds [Noma et al. 2024; Sharp and Crane 2018]. We build on the existing geometric flow of *surface-filling* curves [Noma et al. 2024] and extend them to serve the case of *volume-filling* curves. Furthermore, we provide additional parameters that can guide the design of the curve, like the curve thickness (Figure 6) or the locality pattern (Figure 7).

### 2.3 Preconditioners for Geometric Optimization

In order to accelerate gradient-based geometric optimization, evaluating gradients in the Sobolev space instead of the normal  $L^2$  space dramatically improves speed and robustness. In computer graphics, Sobolev preconditioners have been used for repulsive shape optimization [Yu et al. 2021a,b], inverse rendering [Ling et al. 2022; Nicolet et al. 2021], geometric data interpolation [Wang and Solomon 2021], cloth simulation [Wu et al. 2022], and designing wire art [Tojo et al. 2024]. Gradient preconditioning is also showing excellent performance in optimizing neural fields [Chng et al. 2024] or Gaussian splats [Pehlivan et al. 2025]. Although it is well known

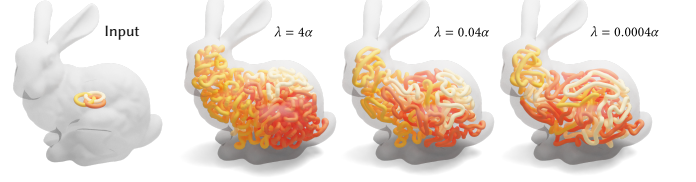


Fig. 7. The preconditioner weight  $\lambda$  controls the effect of higher order spectral components of the curve. Higher  $\lambda$  leads to pronounced wiggles with local entanglements, while lower  $\lambda$  leads to straighter curves with global entanglements.

that the preconditioner must match the order of the energy [Yu et al. 2021a,b], we reveal that matching the preconditioner’s *sparsity pattern* to the energy is also important for performance.

## 3 BACKGROUND

In what follows, let us assume that we are given an arclength parametrized curve  $\gamma : [0, l] \mapsto \mathbb{R}^3$  inside a domain  $\Omega \subset \mathbb{R}^3$ . Our goal will be to find another curve  $\gamma^*$  that fills the interior of  $\Omega$  with some radius  $r$ ; that is, for any point  $p \in \Omega$ , there exists a point  $q \in \gamma^*$  such that  $\|p - q\| \leq r$ . We will do this by constructing an energy functional  $E[\gamma]$  and iteratively minimizing it with respect to  $\gamma$  starting from some initial curve  $\gamma^0$ . We discretize these curves with a set of straight segments  $\gamma_1, \dots, \gamma_n \in \mathbb{R}^3$  connecting  $n$  nodes, whose coordinates are stored in a vector  $\mathbf{y} \in \mathbb{R}^{3n}$ .

Before we dive into our method, let us first briefly cover the ideas from existing work that we rely on.

### 3.1 2D Filling Curves [Noma et al. 2024]

Noma et al. [2024] formulated the problem of computing a space-filling curve  $\gamma : [0, l] \mapsto \mathbb{R}^2$  on a 2D plane by following the gradient flow of a two-term energy

$$E[\gamma] = L_\gamma + \underbrace{\alpha \int_\gamma \|\gamma(s) - m^+(s)\|^2 + \|\gamma(s) - m^-(s)\|^2 ds}_{E_M[\gamma]}, \quad (1)$$

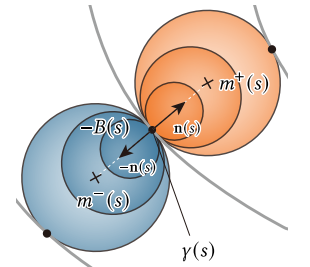
where  $L_\gamma$  is the length of the curve,  $m^+(s), m^-(s)$  are the closest points from  $\gamma(s)$  to the medial axis of  $\gamma$ , and  $\alpha$  is a balancing coefficient. The two points  $m^+(s), m^-(s)$  are found by “growing” a sphere along the principal normal direction  $\mathbf{n}(s)$  and its opposite direction  $-\mathbf{n}(s)$  until it hits another point on the curve (see inset from Noma et al. [2024]), which can be denoted as

$$m^+(s) = \gamma(s) + R^+(s)\mathbf{n}(s), \quad (2)$$

$$m^-(s) = \gamma(s) - R^-(s)\mathbf{n}(s), \quad (3)$$

where  $R^+(s), R^-(s)$  are the sizes of the spheres.

Noma et al.’s gradient flow essentially finds an equilibrium state of this two-term energy where the variational derivative equals to



zero, i.e.,

$$\frac{\delta E}{\delta \gamma}(s) = \mathbf{0} \quad \forall s \in [0, l]. \quad (4)$$

Although they have never explicitly derived the gradient in their paper, let us derive it here. The first variation of the curve length  $L_\gamma$  is known to be

$$\frac{\delta L_\gamma}{\delta \gamma}(s) = -\frac{1}{r(s)} \mathbf{n}(s), \quad (5)$$

where  $r(s)$  is the radius of curvature of  $\gamma$  and  $\mathbf{n}(s)$  is the principal normal direction of  $\gamma$  [Devadoss and O'Rourke 2011, Section 5.5]. Regarding the second term  $E_M$ ,  $m^+(s)$  and  $m^-(s)$  do not vary when we move  $\gamma$  because their method momentarily fixes them during optimization. Thus, the variational derivative of  $E_M$  becomes

$$\frac{\delta E_M}{\delta \gamma}(s) = 2(\gamma(s) - m^+(s)) + 2(\gamma(s) - m^-(s)) \quad (6)$$

$$= 2(R^+(s) - R^-(s)) \mathbf{n}(s). \quad (7)$$

By adding Eq. 5 and 6, we have

$$\frac{\delta E}{\delta \gamma}(s) = \left( 2R^+(s) - 2R^-(s) - \frac{\alpha}{r(s)} \right) \mathbf{n}(s). \quad (8)$$

### 3.2 Sobolev Preconditioning

In order to find the equilibrium of Eq. 8, a naive choice would be to follow the gradient descent flow

$$\gamma^{k+1}(s) = \gamma^k(s) - \tau \underbrace{\frac{\delta E}{\delta \gamma}(s)}_{:=g_{L^2}}, \quad (9)$$

where  $\gamma^k$  is the curve at the  $k$ -th iteration and  $\tau$  is the timestep. This is also called  $L^2$  gradient descent because the gradient  $g_{L^2}$  satisfies the following equation:

$$\langle g_{L^2}, \eta \rangle_{L^2} = \int_\gamma \frac{\delta E}{\delta \gamma} \cdot \eta ds \quad (10)$$

for any  $\eta : [0, l] \mapsto \mathbb{R}^3$ , where  $\langle \cdot, \cdot \rangle_{L^2}$  is the  $L^2$  inner product.

However, this only allows small steps due to the instability for higher order spectral components. Instead, a reasonable choice is to use the  $H^p$  inner product evaluated in Sobolev space:

$$\langle g_{H^p}, \eta \rangle_{H^p} = \langle (\Delta + \lambda I)^p g_{H^p}, \eta \rangle_{L^2} = \int_\gamma \frac{\delta E}{\delta \gamma} \cdot \eta ds, \quad (11)$$

where  $\lambda$  is a coefficient that controls the higher-order components. This leads to the *preconditioned* gradient descent flow

$$\gamma^{k+1}(s) = \gamma^k(s) - \tau(\Delta + \lambda I)^{-p} \frac{\delta E}{\delta \gamma}(s), \quad (12)$$

which is typically discretized as

$$\gamma^{k+1} = \gamma^k - \tau(\mathbf{L} + \lambda \mathbf{M})^{-p} \mathbf{g}, \quad (13)$$

where  $\mathbf{M}$  is the mass matrix,  $\mathbf{L}$  is the discrete Laplacian, and  $\mathbf{g} = \nabla E(\gamma)$  is the pointwise gradient of the energy.

In fact, the quasi-Newton preconditioner used in Noma et al. [2024] takes exactly the same form, where  $p = 1$ ,  $\lambda = \alpha$ ,  $\mathbf{M} \in \mathbb{R}^{2n \times 2n}$

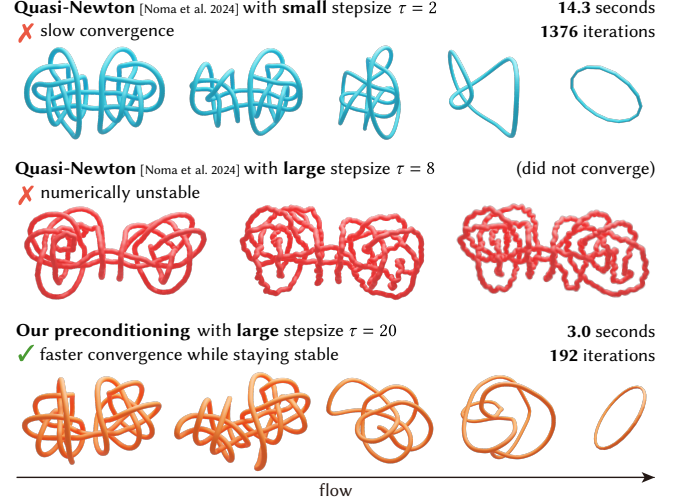


Fig. 8. The quasi-Newton scheme used in Noma et al. [2024] is slow with a small step size, but unstable with a large one. Our *medial sphere preconditioning* stays stable even at larger  $\tau$ , leading to faster convergence.

is a diagonal matrix that stores the average length of the incident edges, and

$$\mathbf{L} = \begin{bmatrix} \tilde{\mathbf{L}} & \\ & \tilde{\mathbf{L}} \end{bmatrix} \in \mathbb{R}^{2n \times 2n} \quad (14)$$

is the curve Laplacian that has

$$\tilde{L}_{ij} = \begin{cases} \sum_{k \in \mathcal{N}(i)} 1/l_{ik} & \text{if } i = j \\ -1/l_{ij} & \text{if } j \in \mathcal{N}(i) \\ 0 & \text{else} \end{cases} \quad (15)$$

where  $\mathcal{N}(i)$  is the neighborhood of the node  $i$  and  $l_{ij}$  is the length of the segment that connects node  $i$  and  $j$ .

These  $H^p$  preconditioners are most effective when their order  $2p$  exactly matches the energy's spatial derivatives. For example, Nicolet et al. [2021] use  $p = 2$  to account for their diffusion term, while Yu et al. [2021a; 2021b] use a fractional differential operator to match the order of their fractional energy function.

In this regard, having  $p = 1$  in the quasi-Newton preconditioner of Noma et al. [2024] sounds reasonable because the spatial derivative of the 2D-filling energy in Eq. 1 is second order, due to the appearance of curvature in its gradient in Eq. 8. However, we reveal in Section 4.2 that there is another condition that is forgotten, hindering the performance (Figure 8).

## 4 METHOD

We now explain our method to find the volume-filling curve  $\gamma^*$ . We extend the surface-filling energy in Eq. 1 to 3D, and propose the *medial sphere preconditioner* that allows faster convergence. We also describe the parameter choices specifically tailored for knot untangling.

### 4.1 Energy

In order to extend the 2D filling curve energy in Eq. 1 to 3D, one problem is how to extend the the medial axis energy  $E_M[\gamma]$  to 3D.



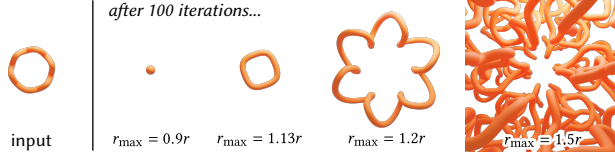


Fig. 9. Smaller  $r_{\max}$  leads the curve to shrink into a single point, while larger  $r_{\max}$  leads the curve to grow outside faster.

The medial axis  $M_\gamma$  of a curve  $\gamma$  is the set of points having more than one closest point on the curve. In 2D it is a nonmanifold *curve*, while in 3D it is a nonmanifold *surface*.  $M_\gamma$  is the local maxima of the minimum distance function  $d(p, \gamma) = \min_{q \in \gamma} \|p - q\|$ , and thus we may impose that if  $d(p, \gamma)$  is below  $r$  for every  $p \in M_\gamma$ , so it is for every  $p \in \Omega$ . In other words,  $\gamma$  is a space-filling curve if  $d(p, \gamma)$  is below  $r$  for all  $p$  on the medial axis  $M_\gamma$ .

The 2D medial axis energy is based on the observation that one may create a surjective mapping from  $[0, l]$  to  $M_\gamma$  represented as  $m^+$  and  $m^-$ , based on the medial sphere grown from  $\gamma(s)$  along  $\mathbf{n}(s)$  and  $-\mathbf{n}(s)$ . However, in 3D, the sphere can be grown along all directions on the unit circle on the bitangent plane  $N(s)$  (see inset). Let  $m(\gamma(s), x)$  be the center position of the sphere grown from  $\gamma(s)$  along  $x \in S^1$  on  $N(s)$ , the function  $m : \gamma \times S^1 \rightarrow M_\gamma$  is a *surjective* function where every point on the medial axis has one or more pairs of  $\gamma(s) \in \gamma$  and  $x \in S^1$ . This allows us to redefine the space-filling condition to have  $\|\gamma(s) - m(\gamma(s), x)\| < r$  for all pairs of  $(s, x)$ , leading to our *volume-filling* energy as

$$E_M[\gamma] = \int_{\gamma} \int_{x \in S^1} \|\gamma(s) - m(\gamma(s), x)\|^2 dx ds. \quad (16)$$

The total energy thus becomes

$$E[\gamma] = L_\gamma + \alpha E_M[\gamma], \quad (17)$$

where the variational derivative is

$$\frac{\delta E}{\delta \gamma}(s) = -\frac{1}{r(s)} \mathbf{n}(s) + \alpha \int_{x \in S^1} (\gamma(s) - m(\gamma(s), x)) dx. \quad (18)$$

**Discretization.** We discretize the gradient of the energy by uniformly sampling  $N$  rays on the unit circle  $S^1$ , which we use  $N = 16$  as our default parameter. The unit circle is laid on the bitangent plane at  $\gamma_i$ , which we define by averaging the vector along the incoming and outgoing segments. Let  $\mathbf{x}_i^l$  be the  $l$ -th direction that is uniformly sampled on the unit circle,  $\kappa_i$  be the discrete curvature at  $\gamma_i$ , and  $\mathbf{n}_i$  be the curve normal direction at  $\gamma_i$ . With these in hand, we discretize the gradient of the energy as

$$\frac{\partial E}{\partial \gamma_i} = -\kappa_i \mathbf{n}_i + \alpha \frac{2\pi}{N} \sum_{l=1}^N (\gamma_i - m(\gamma_i, \mathbf{x}_i^l)). \quad (19)$$

To find out the sphere center  $m(\gamma_i, \mathbf{x}_i^l)$ , we use the same strategy as Noma et al. [2024] to run a binary search on the ray shot from  $\gamma_i$  along  $\mathbf{x}_i^l$ . Starting from a large enough step  $r_{\max}$  (Figure 9), we iteratively halve the search range until it finds the largest sphere

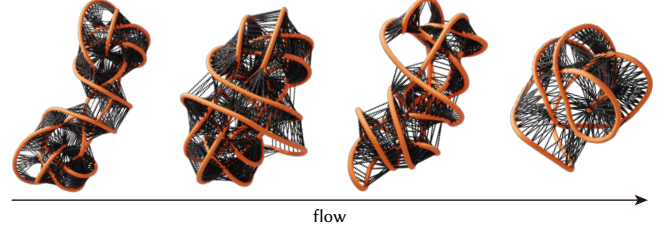


Fig. 10. Our preconditioner considers the connectivity that shares the same medial sphere (black) in addition to the local connectivity of the curve (orange).

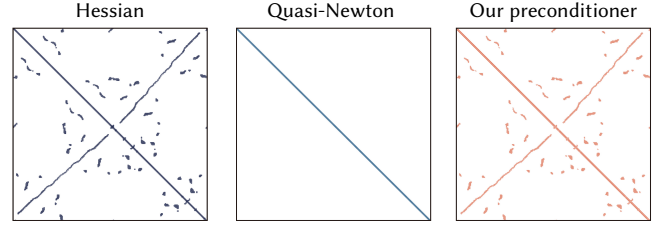


Fig. 11. Our medial sphere preconditioner matches the sparsity pattern of the Hessian, unlike the Quasi-Newton preconditioner of Noma et al. [2024]. The sparsity pattern is measured on the Freedman-He-Wang knot in Figure 1.

where no other nodes on the curve (and the input surface if exists) are included inside it. This procedure can be done by a closest point query, where off-the-shelf implementations are fast and robust enough.

We note that  $m(\gamma_i, \mathbf{x}_i^l)$  depends not only on  $\gamma_i$  but also on the node  $\gamma_j$  that the medial sphere passes through. Thus, the Hessian of  $E_M$  includes not only diagonal terms  $\partial^2 E / \partial \gamma_i^2$  but also cross terms  $\partial^2 E / \partial \gamma_i \partial \gamma_j$  (see Figure 11 left).

## 4.2 Medial Sphere Preconditioner

We now explain our medial sphere preconditioner. The key intuition is that the preconditioner must respect the sparsity pattern of the Hessian of the energy (Figure 11) to achieve fast yet robust convergence.

Recall that the preconditioned gradient descent is discretized as

$$\gamma^{k+1} = \gamma^k - \tau \mathbf{P}^{-1} \mathbf{g}, \quad (20)$$

where  $\mathbf{P} = \mathbf{L} + \lambda \mathbf{M}$  is the preconditioner where  $\lambda$  controls the magnitude of higher-order components (Figure 7). Instead of using the curve Laplacian  $\mathbf{L}$  that only considers the *local* connectivity, we use a modified Laplacian  $\mathbf{L}^*$  that considers *global* interactions (Figure 10). Let  $\mathcal{M}(i)$  be the set of curve nodes that share the same medial sphere as  $i$ , i.e.,  $j \in \mathcal{M}(i)$  if and only if there exists a vector  $x \in S^1$  where  $\|\gamma_i - m_i(x)\| = \|\gamma_j - m_i(x)\|$  or  $\|\gamma_i - m_j(x)\| = \|\gamma_j - m_j(x)\|$ . Then we have the *modified* Laplacian

$$\mathbf{L}^* = \begin{bmatrix} \tilde{\mathbf{L}}^* & & \\ & \mathbf{L}^* & \\ & & \tilde{\mathbf{L}}^* \end{bmatrix} \in \mathbb{R}^{3n \times 3n}, \quad (21)$$

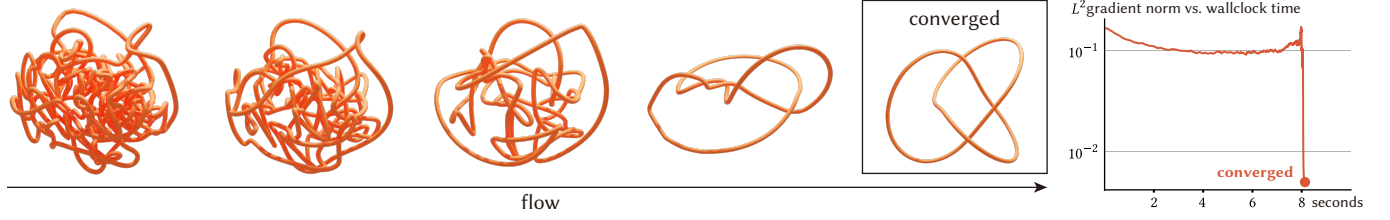


Fig. 12. Our method can untangle a knot from the TREFOIL100 dataset in 8 seconds without changing the knot group.

where

$$\tilde{\mathbf{L}}^*_{ij} = \begin{cases} \sum_{k \in \mathcal{N}(i)} 1/l_{ik} + \sum_{k \in \mathcal{M}(i)} \beta/l_{ik} & \text{if } i = j \\ -1/l_{ij} & \text{if } j \in \mathcal{N}(i) \\ -\beta/l_{ij} & \text{if } j \in \mathcal{M}(i) \\ 0 & \text{else} \end{cases} \quad (22)$$

and  $\beta$  is a coefficient which we set to  $\beta = 0.01$  (Figure 18).

*Why is this faster?* Let the second-order component of the local Taylor expansion of  $E(\boldsymbol{\gamma})$  around  $\boldsymbol{\gamma} = \boldsymbol{\gamma}_k$  be

$$\tilde{E}(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x}, \quad (23)$$

where  $\mathbf{x} = \boldsymbol{\gamma} - \boldsymbol{\gamma}_k$  and  $\mathbf{H} = D^2 E(\boldsymbol{\gamma})$  is the Hessian of  $E$ . In order to achieve second-order convergence, it is important for every step to effectively reduce this energy. As  $\nabla \tilde{E}(\mathbf{x}) = \mathbf{H} \mathbf{x}$ , the steepest preconditioned gradient step would be

$$\mathbf{x}^{k+1} = (\mathbf{I} - \tau \mathbf{P}^{-1} \mathbf{H}) \mathbf{x}^k. \quad (24)$$

Given that  $\mathbf{A} = \mathbf{P}^{-1} \mathbf{H}$  is a symmetric positive semi-definite matrix, we may expand  $\mathbf{x}^k$  and  $\mathbf{x}^{k+1}$  into its eigenvectors, i.e.,

$$\mathbf{x}^k \approx \sum_i v_i^k \boldsymbol{\psi}_i \quad (25)$$

where  $v_i^k$  is the component along the  $i$ -th eigenvector  $\boldsymbol{\psi}_i$  whose eigenvalue is denoted as  $\mu_i$ . By considering only the  $i$ -th eigendirection in Eq. 24 we have

$$v_i^{k+1} = (1 - \tau \mu_i) v_i^k. \quad (26)$$

To reduce the quadratic energy along every  $i$ ,  $|1 - \tau \mu_i| < 1$  must be satisfied for every  $i$ . This is sufficient when the largest eigenvalue  $\mu_{\max}$  satisfies this condition, which can be rewritten as

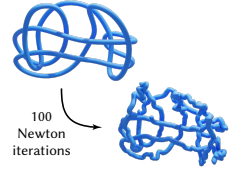
$$0 < \tau < \frac{2}{\mu_{\max}} := \tau_{\max}. \quad (27)$$

This indicates that the largest admissible stepsize is determined by the largest eigenvalue of  $\mathbf{A} = \mathbf{P}^{-1} \mathbf{H}$ . As larger eigenvalues correspond to high-frequency spectral components, it is important for  $\mathbf{P}$  to preserve the same sparsity of  $\mathbf{H}$ . Our preconditioner has an identical sparsity pattern to the Hessian (Figure 11), and this allows larger timesteps compared to the Quasi-Newton scheme (Figure 8).

*Why not use the actual Hessian?* Another tempting approach is to use Newton's method and use the inverse of the Hessian as the preconditioner. The Hessian is obtained by differentiating Eq. 19, which contains a term regarding the medial sphere grown from  $\gamma_i$  along  $\mathbf{x}_i^l$ :

$$R_i^l = \frac{\|\gamma_j - \gamma_i\|^2}{2 \mathbf{x}_i^l \cdot (\gamma_j - \gamma_i)}. \quad (28)$$

However, we observed that Newton's method produces spikes on the curve due to the fractional term approaching infinity, and thus leading the flow to stall (see inset).



### 4.3 Timestepping

For each timestep, we apply a backtracking line search equipped with a continuous collision detection (CCD) to ensure that the curve does not change its topology each step. We do this by triangulating the swept surface of the curve with the descent direction  $\mathbf{d}$  and checking if this mesh does not cause any global self-intersection using a standard bounding volume hierarchy. This check guarantees that the topology is preserved, which we show a proof in the supplemental material.

Following the lead of previous approaches [Noma et al. 2024], we remesh the curve each timestep so that the curve has equally spaced segments. We do this by removing segments shorter than a certain threshold  $h$  and subdividing segments longer than  $2h$ . We empirically found that  $h < r/5$  is sufficient to keep the flow robust.

### 4.4 Knot Untangling

Based on the volume-filling curve flow, we build an algorithm that efficiently untangles knots. As our flow quickly finds an equilibrium where the curve  $\gamma$  has equal spacing  $r$ , it can untangle knots while maintaining this equal spacing throughout the flow (see Figure 1).

We first compute the average medial sphere size  $r_{\text{avg}}$ . Next, we run a single step of our volume-filling flow with  $r = r_{\text{avg}}$  and  $r_{\text{max}} = 0.9 r_{\text{avg}}$ . Setting  $r_{\text{max}}$  to a smaller value than  $r$  produces an external tightening force (Figure 9), leading the curve to shrink while maintaining an equal spacing (Figure 12). We also use a small  $\lambda = 1$  to smoothly untangle the curve without suffering from higher-order components. Although the flow shrinks the curve into a small one, for visualization purposes, we may simply scale it so that it retains the original length of the curve.

We terminate the flow if (1) the  $L^2$  norm of the gradient per node becomes lower than  $5 \times 10^{-3} r$  or (2) the number of nodes becomes

Table 1. We report the average runtime and iterations to untangle the dataset. See the supplemental material for full details.

Dataset	#Curves	Success Rate	Avg. Runtime	Avg. #Iters
KNOT128	128	100 %	8.7 s	720
TREFOIL100	100	100 %	9.1 s	555

no larger than 10 due to remeshing. Non-trivial knots (e.g., trefoils) terminate because they reach the simplest state (see Figure 12), while trivial knots terminate because they contract into a single point due to small  $r_{\max}$  and the number of nodes becomes sufficiently small (see Figure 9). In the case of (2), we remap the curve onto a planar circle for visualization.

## 5 RESULTS

We implemented our algorithm in C++ using GEOMETRY CENTRAL [Sharp et al. 2019] for standard geometric operations, EIGEN [Guennebaud et al. 2010] for sparse linear solves, TINYAD [Schmidt et al. 2022] for automatic differentiation, and KNN-CPP [Meyer 2019] for closest point queries. We measured the runtime on a 2021 M1 MacBook Pro with 32 GB of RAM.

### 5.1 Experiments

We mainly compare the performance of our algorithm to Repulsive Curves [Yu et al. 2021b], the state-of-the-art algorithm in this task. While several other algorithms can untangle knots (e.g., KnotPlot [Scharein 1998] or SONO [Pieranski 1998]), Repulsive Curves is orders of magnitude faster (see [Yu et al. 2021b, Figure 15]), and we thus do not perform a full comparison with them. In the following experiments, we used the default parameters explained in the previous sections.

In Figure 3, we measured the runtime of our method compared to Repulsive Curves [Yu et al. 2021b]. Even when using their multigrid scheme, the runtime per iteration of our method is orders of magnitude faster at the same number of nodes. We also confirmed that the computational complexity per iteration scales with  $O(n^{1+f})$ , where  $f > 0$  accounts for the Laplacian-like sparse system solve.

This speedup results in faster untangling of the Freedman-He-Wang knot [1994] (Figure 1), a well-known challenging knot. With a large timestep  $\tau = 20$ , our method can untangle this knot 77x faster (3 seconds) than Repulsive Curves (230 seconds).

We also ran our algorithm on the knot dataset (KNOT 128 and TREFOIL 100) in the Repulsive Curves paper [Yu et al. 2021b]. Among the 228 models in the dataset, our method showed a 100% success rate using our termination condition (Figure 13 and Table 1), while Yu et al. reported a 87.7% (200/228) success rate. This improved robustness is thanks to our energy that can avoid curves to get closer than  $r$  (Figure 4). We also note that our termination condition in Section 4.4 works robustly, where all the non-trivial knots terminated with condition (1). The only knot terminated with condition (2) was the only trivial knot in the dataset (KNOT0001). See the supplemental material for details on the runtime and termination conditions per model.

Additionally, our method can untangle Ochiai knots [1990], another set of difficult unknots even in discrete knot theory literature

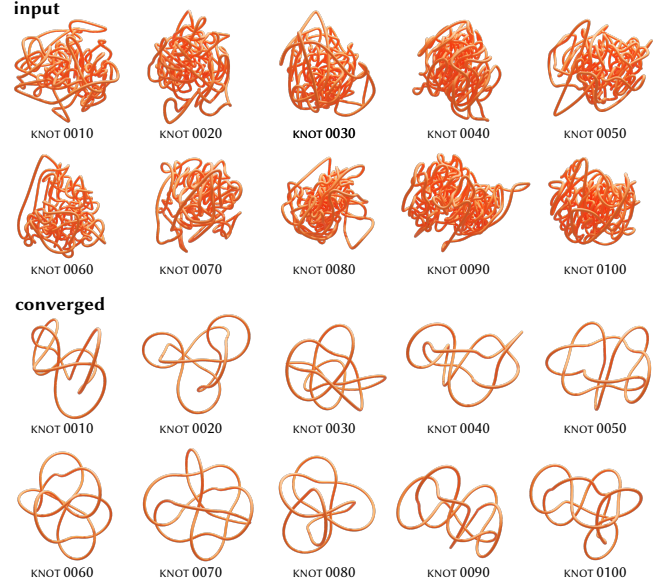


Fig. 13. Our method achieved 100% success rate to untangle all the 128 knots from the KNOT128 dataset [Yu et al. 2021b].

[Burton et al. 2021]. While Repulsive Curves failed to untangle OCHIAI III and OCHIAI IV knots (Figure 4), our method succeeded in untangling all four trivial knots in the order of seconds (Figure 15).

We measured the effect of setting  $\tau$  to a larger value, compared to the quasi-Newton scheme of Noma et al. [2024]. As shown in Figure 14, quasi-Newton, while showing faster runtime at  $\tau < 2$ , exhibits a significant increase in runtime with larger  $\tau$  due to numerical instability (see Figure 8). In comparison, our preconditioner steadily decreases the runtime as we increase the step size.

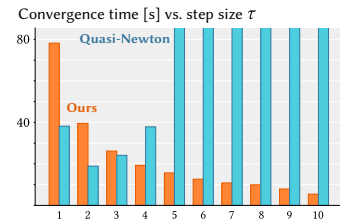


Fig. 14. Unlike quasi-Newton [Noma et al. 2024], our preconditioner steadily decreases the convergence time as we increase the step size  $\tau$ . Measured on the Freedman-He-Wang knot.

To justify the choice of the parameter  $\beta = 0.01$  used in our modified Laplacian  $L^*$ , we also measured the effect of changing it (Figure 18). We observed that  $\beta = 0$ , which means not using our medial sphere preconditioning at all, results in early stalls due to self-intersections. On the other hand, a larger  $\beta = 1$  leads to numerical instabilities.

### 5.2 User Control and Ablations

We measured the effect of our tunable parameters, which allow additional user control.

The energy weight  $\alpha$  can control the curve thickness  $r$  (Figure 6). Similar to Noma et al. [2024], we observed an empirical correlation

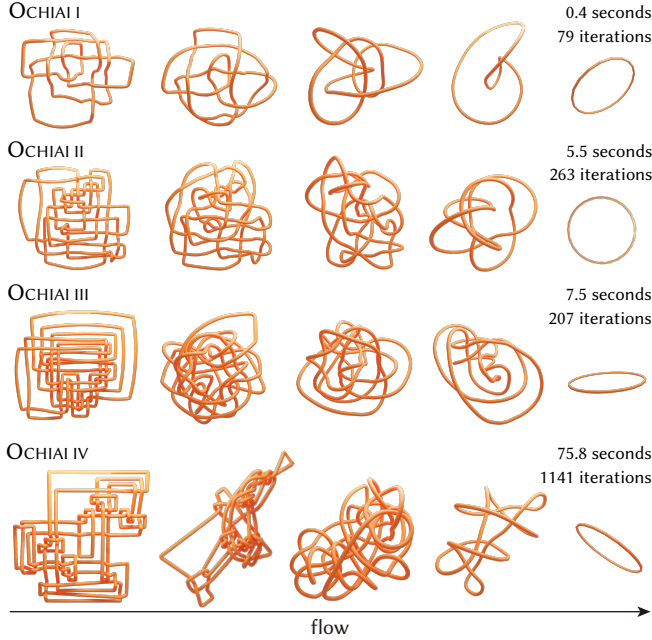


Fig. 15. Our method can untangle the Ochiai unknots [1990] that are known to be challenging.

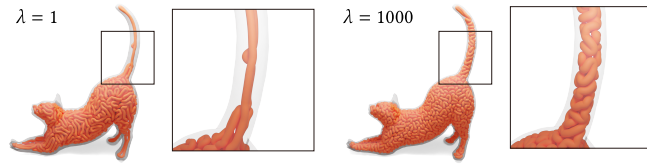


Fig. 16. Higher  $\lambda$  achieves higher coverage of the curve at thin regions, at the cost of having pronounced wriggles.

of  $r \approx 3.36/\sqrt{\alpha}$ , allowing the user to obtain a curve with a desired curve thickness.

The preconditioner weight  $\lambda$  can control the magnitude of the higher-order spectral components that lead to pronounced wriggles (Figure 7). As a result, the user can have a higher  $\lambda$  to generate a curve with local entanglement, while a lower  $\lambda$  to generate global entanglement. We also observed that lower  $\lambda$  leads to lower coverage due to the Laplacian’s smoothing effect on the gradient directions (Figure 16).

The binary search radius  $r_{\max}$  can control the growing speed of the curve (Figure 9), which could be useful for creating animations.

Finally, the number of ray directions  $N$  is crucial for runtime, as the number of closest point queries we need to run is proportional to  $N$  (Figure 17). We observed that as we reduce  $N$ , the runtime gets shortened at the cost of convergence issues, leading to a lower success rate in the knot dataset (226/228 with  $N = 8$  and 136/228 with  $N = 4$ ) than the full flow (228/228 with  $N = 16$ , see the supplemental material). This suggests that a smaller  $N$  could be used for a preview suitable for trial and error of tweaking the parameters, while a larger  $N$  could be used later for the full flow.

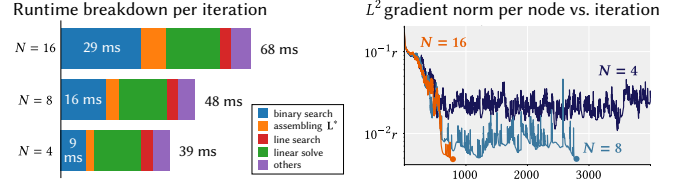


Fig. 17. Reducing the number of ray directions  $N$  can reduce the runtime per iteration, at the cost of causing convergence issues.

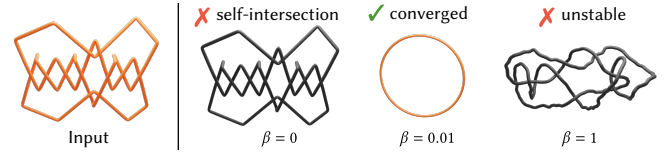


Fig. 18. We empirically find that  $\beta = 0.01$  keeps the flow stable, without causing self-intersections or numerical instability at larger time steps.

## 6 CONCLUSION AND DISCUSSION

We proposed a volume-filling curve energy and a knot untangling algorithm that serves as a drop-in replacement of Repulsive Curves [Yu et al. 2021b], showing significant improvements in speed and robustness. We extensively showed that our method can untangle a wide variety of knots quickly and robustly. Our method offers several tunable parameters, allowing users to guide the design of the curve.

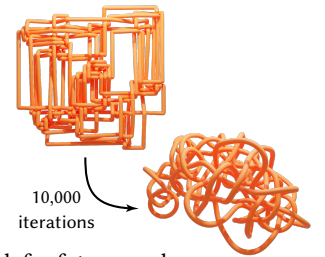
Our fast, robust, and controllable algorithm could open up a new design space in motion path planning and computational design. Our radius control can change the spacing between the curves when, e.g., visualizing vascular networks (see [Yu et al. 2021b, Fig. 28]). Another useful scenario would be analyzing protein knots [Hsu et al. 2024] with our fast and robust flow.

A fundamental theoretical inquiry persists regarding the empirical relationship between the curve thickness  $r$  and the weighting parameter  $\alpha$ .

Our flow did not terminate on Haken’s Gordian knot [Petronio and Zanellati 2016] even after a sufficient runtime (see inset). We thus leave this task for future work.

Although we did not resort to any multigrid or parallelization schemes, applying them could make our algorithm even faster. Especially, finding the medial axis positions via binary search was the main bottleneck of our method (Figure 17). Thus, parallelizing the closest point query on the GPU could potentially lead to a speedup.

Finally, we hope that our volume-filling energy and knot untangling algorithm will help the computer graphics community and artists to design volume-filling curves interactively and robustly.





## ACKNOWLEDGMENTS

We thank Thea Yu for early discussions, Abhishek Madan and Karran Pandey for proofreading, and Esther Lin for video narration. Our research is funded in part by NSERC Discovery (RGPIN–2022–04680), the Ontario Early Research Award program, the Canada Research Chairs Program, a Sloan Research Fellowship, the DSI Catalyst Grant program and gifts by Adobe Inc.

## REFERENCES

- Ted Ashton, Jason Cantarella, Michael Piatek, and Eric Rawdon. 2010. Knot Tightening By Constrained Gradient Descent. arXiv:1002.1723 [math.DG] <https://arxiv.org/abs/1002.1723>
- Sören Bartels, Philipp Reiter, and Johannes Riege. 2017. A simple scheme for the approximation of self-avoiding inextensible curves. *IMA J. Numer. Anal.* 38, 2 (05 2017), 543–565. <https://doi.org/10.1093/imanum/drx021> arXiv:<https://academic.oup.com/imanj/article-pdf/38/2/543/24654947/drx021.pdf>
- A. Bedel, Y. Coudert-Osmont, J. Martinez, R. I. Nishat, S. Whitesides, and S. Lefebvre. 2022. Closed space-filling curves with controlled orientation for 3D printing. *Computer Graphics Forum* 41, 2 (2022), 473–492. <https://doi.org/10.1111/cgf.14488> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14488>
- Joel Brown, Jean-Claude Latombe, and Kevin Montgomery. 2004. Real-time knot-tying simulation. *Vis. Comput.* 20, 2 (May 2004), 165–179. <https://doi.org/10.1007/s00371-003-0226-y>
- Benjamin A. Burton, Hsien-Chih Chang, Maarten Löffler, Arnaud de Mesmay, Clément Maria, Saul Schleimer, Eric Sedgwick, and Jonathan Spreer. 2021. Hard Diagrams of the Unknot. arXiv:2104.14076 [math.GT] <https://arxiv.org/abs/2104.14076>
- Puikui Cheng, Wing Kam Liu, Kornel Ehmann, and Jian Cao. 2020. Enumeration of additive manufacturing toolpaths using Hamiltonian paths. *Manufacturing Letters* 26 (2020), 29–32.
- Shin-Fang Chng, Hemanth Saratchandran, and Simon Lucey. 2024. Preconditioners for the Stochastic Training of Implicit Neural Representations. arXiv:2402.08784 [cs.CV] <https://arxiv.org/abs/2402.08784>
- Chengkai Dai, Charlie C. L. Wang, Chenming Wu, Sylvain Lefebvre, Guoxin Fang, and Yong-Jin Liu. 2018. Support-free volume printing by multi-axis motion. *ACM Trans. Graph.* 37, 4, Article 134 (July 2018), 14 pages. <https://doi.org/10.1145/3197517.3201342>
- Arnaud de Mesmay, Yo’av Rieck, Eric Sedgwick, and Martin Tancer. 2018. The unbearable hardness of unknotting. arXiv:1810.03502 [math.GT] <https://arxiv.org/abs/1810.03502>
- Satyan L. Devadoss and Joseph O’Rourke. 2011. *Discrete and Computational Geometry*. Princeton University Press.
- Jimmy Etienne, Nicolas Ray, Daniele Panozzo, Samuel Hornus, Charlie C. L. Wang, Jonàs Martínez, Sara McMains, Marc Alexa, Brian Wyvill, and Sylvain Lefebvre. 2019. CurviSlicer: slightly curved slicing for 3-axis printers. *ACM Trans. Graph.* 38, 4, Article 81 (July 2019), 11 pages. <https://doi.org/10.1145/3306346.3323022>
- Guoxin Fang, Tianyu Zhang, Sikai Zhong, Xiangjia Chen, Zichun Zhong, and Charlie C. L. Wang. 2020. Reinforced FDM: multi-axis filament alignment with controlled anisotropic strength. *ACM Trans. Graph.* 39, 6, Article 204 (Nov. 2020), 15 pages. <https://doi.org/10.1145/3414685.3417834>
- Michael H Freedman, Zheng-Xu He, and Zhenghan Wang. 1994. Möbius energy of knots and unknots. *Annals of mathematics* 139, 1 (1994), 1–50.
- John Giannatsis, Aggelos Vassilakos, Vassilios Canellidis, and Vassilis Dedoussis. 2015. Fabrication of graded structures by extrusion 3D Printing. In *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE, 175–179.
- Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. <http://eigen.tuxfamily.org>.
- David Harmon, Etienne Vouga, Breannan Smith, Rasmus Tamstorf, and Eitan Grinspun. 2012. Asynchronous contact mechanics. *Commun. ACM* 55, 4 (April 2012), 102–109. <https://doi.org/10.1145/2133806.2133828>
- David Hilbert. 1891. Ueber die stetige Abbildung einer Linie auf ein Flächenstück. *Math. Ann.* (1891).
- Yuichi Hirose, Mark Gillespie, Angelica M. Bonilla Fominaya, and James McCann. 2024. Solid Knitting. *ACM Trans. Graph.* 43, 4, Article 88 (July 2024), 15 pages. <https://doi.org/10.1145/3658123>
- Min-Feng Hsu, Manoj Kumar Sriramoju, Chih-Hsuan Lai, Yun-Ru Chen, Jing-Siou Huang, Tzu-Ping Ko, Kai-Fa Huang, and Shang-Te Danny Hsu. 2024. Structure, dynamics, and stability of the smallest and most complex 71 protein knot. *Journal of Biological Chemistry* 300, 1 (2024), 105553. <https://doi.org/10.1016/j.jbc.2023.105553>
- Anant A Joshi, Maulik C Bhatt, and Arpita Sinha. 2019. Modification of Hilbert’s space-filling curve to avoid obstacles: a robotic path-planning strategy. In *2019 Sixth Indian Control Conference (ICC)*. IEEE, 338–343.
- Blazej Kubiak, Nico Pietroni, Fabio Ganovelli, and Marco Fratarcangeli. 2007. A robust method for real-time thread simulation. In *Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Technology* (Newport Beach, California) (VRST ’07). Association for Computing Machinery, New York, NY, USA, 85–88. <https://doi.org/10.1145/1315184.1315198>
- R. B. Kusner and J. M. Sullivan. 1998. *Möbius-Invariant Knot Energies*. 315–352. [https://doi.org/10.1142/9789812796073\\_0017](https://doi.org/10.1142/9789812796073_0017)
- Andrew M. Ladd and Lydia E. Kavraki. 2004. *Motion Planning for Knot Untangling*. Springer Berlin Heidelberg, Berlin, Heidelberg, 7–23. [https://doi.org/10.1007/978-3-540-45058-0\\_2](https://doi.org/10.1007/978-3-540-45058-0_2)
- Sen Lin, Lingwei Xia, Guowei Ma, Shiwei Zhou, and Yi Min Xie. 2019. A maze-like path generation scheme for fused deposition modeling. *The International Journal of Advanced Manufacturing Technology* 104 (2019), 1509–1519.
- Selena Ling, Nicholas Sharp, and Alec Jacobson. 2022. VectorAdam for rotation equivariant geometry optimization. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) (NIPS ’22). Curran Associates Inc., Red Hook, NY, USA, Article 297, 12 pages.
- Tao Liu, Tianyu Zhang, Yongxue Chen, Yuming Huang, and Charlie C. L. Wang. 2024. Neural Slicer for Multi-Axis 3D Printing. *ACM Trans. Graph.* 43, 4, Article 85 (July 2024), 15 pages. <https://doi.org/10.1145/3658212>
- Fabian Meyer. 2019. knn-cpp. <https://github.com/Rookfighter/knn-cpp>.
- Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. 2021. Large steps in inverse rendering of geometry. *ACM Trans. Graph.* 40, 6, Article 248 (Dec. 2021), 13 pages. <https://doi.org/10.1145/3478513.3480501>
- Yuta Noma, Silvia Sellán, Nicholas Sharp, Karan Singh, and Alec Jacobson. 2024. Surface-Filling Curve Flows via Implicit Medial Axes. *ACM Trans. Graph.* 43, 4, Article 147 (July 2024), 12 pages. <https://doi.org/10.1145/3658158>
- Mitsuyuki Ochiai. 1990. Non-trivial projections of the trivial knot. In *Algorithmique, topologie et géométrie algébriques - Séville, 1987, Toulouse 1988, Hayat-Légrand Claude and Sergeraert Francis* (Eds.). Number 192 in Astérisque. Société mathématique de France, 7–10. [https://www.numdam.org/item/AST\\_1990\\_\\_192\\_7\\_0/](https://www.numdam.org/item/AST_1990__192_7_0/)
- G Peano. 1890. Sur une courbe qui remplit toute une aire plane. *Math. Ann.* 36 (1890).
- Hans Pedersen and Karan Singh. 2006. Organic Labyrinths and Mazes. In *Proceedings of the 4th International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France) (NPAR ’06). Association for Computing Machinery, New York, NY, USA, 79–86. <https://doi.org/10.1145/1124728.1124742>
- Hamza Pehlivan, Andrea Boscolo Camilletto, Lin Geng Foo, Marc Habermann, and Christian Theobalt. 2025. Second-order Optimization of Gaussian Splats with Importance Sampling. arXiv:2504.12905 [cs.CV] <https://arxiv.org/abs/2504.12905>
- Carlo Petronio and Adolfo Zanellati. 2016. Algorithmic simplification of knot diagrams: new moves and experiments. arXiv:1508.03226 [math.GT] <https://arxiv.org/abs/1508.03226>
- Piotr Pieranski. 1998. In search of ideal knots. *Computational Methods in Science and Technology* 4 (12 1998). <https://doi.org/10.12921/cmst.1998.04.01.09-23>
- Rei Sakura and Yasuaki Kakehi. 2025. A 3D-Printed Touch Sensor with a Single-Stroke Conductive Path. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA ’25)*. Association for Computing Machinery, New York, NY, USA, Article 38, 6 pages. <https://doi.org/10.1145/3706599.3720023>
- Robert G. Scharein. 1998. *Interactive Topological Drawing*. Ph.D. Dissertation. Department of Computer Science, The University of British Columbia.
- P. Schmidt, J. Born, D. Bommes, M. Campen, and L. Kobbelt. 2022. TinyAD: Automatic Differentiation in Geometry Processing Made Simple. *Computer Graphics Forum* 41, 5 (2022), 113–124. <https://doi.org/10.1111/cgf.14607> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14607>
- Nicholas Sharp and Keenan Crane. 2018. Variational Surface Cutting. *ACM Trans. Graph.* 37, 4, Article 156 (jul 2018), 13 pages. <https://doi.org/10.1145/3197517.3201356>
- Nicholas Sharp, Keenan Crane, et al. 2019. GeometryCentral: A modern C++ library of data structures and algorithms for geometry processing. <https://geometry-central.net/>. (2019).
- Priya Sundaresan, Jennifer Grannen, Brijen Thananjeyan, Ashwin Balakrishna, Jeffrey Ichnowski, Ellen Novoseller, Minh Hwang, Michael Laskey, Joseph E. Gonzalez, and Ken Goldberg. 2021. Untangling Dense Non-Planar Knots by Learning Manipulation Features and Recovery Policies. arXiv:2107.08942 [cs.RO] <https://arxiv.org/abs/2107.08942>
- Kenji Tojo, Ariel Shamir, Bernd Bickel, and Nobuyuki Umetani. 2024. Fabricable 3D Wire Art. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) (SIGGRAPH ’24). Association for Computing Machinery, New York, NY, USA, Article 134, 11 pages. <https://doi.org/10.1145/3641519.3657453>
- Ultimaker B.V. 2023. *Ultimaker Cura*. <https://github.com/Ultimaker/Cura>
- Shawn W. Walker. 2016. Shape optimization of self-avoiding curves. *J. Comput. Phys.* 311 (2016), 275–298. <https://doi.org/10.1016/j.jcp.2016.02.011>
- Yu Wang and Justin Solomon. 2021. Fast quasi-harmonic weights for geometric data interpolation. *ACM Trans. Graph.* 40, 4, Article 73 (July 2021), 15 pages. <https://doi.org/10.1145/3450626.3459801>
- Botao Wu, Zhendong Wang, and Huamin Wang. 2022. A GPU-based multilevel additive schwarz preconditioner for cloth and deformable body simulation. *ACM Trans. Graph.* 41, 4, Article 63 (July 2022), 14 pages. <https://doi.org/10.1145/3528223>

3530085

- Yuan Yan and Yasamin Mostofi. 2016. Efficient clustering and path planning strategies for robotic data collection using space-filling curves. *IEEE Transactions on Control of Network Systems* 4, 4 (2016), 838–849.
- Chris Yu, Caleb Brakensiek, Henrik Schumacher, and Keenan Crane. 2021a. Repulsive surfaces. *ACM Trans. Graph.* 40, 6, Article 268 (Dec. 2021), 19 pages. <https://doi.org/10.1145/3478513.3480521>
- Chris Yu, Henrik Schumacher, and Keenan Crane. 2021b. Repulsive Curves. *ACM Trans. Graph.* 40, 2, Article 10 (may 2021), 21 pages. <https://doi.org/10.1145/3439429>
- Tianyu Zhang, Guoxin Fang, Yuming Huang, Neelotpal Dutta, Sylvain Lefebvre, Zekai Murat Kilic, and Charlie C. L. Wang. 2022. S3-Slicer: A General Slicing Framework for Multi-Axis 3D Printing. *ACM Trans. Graph.* 41, 6, Article 277 (Nov. 2022), 15 pages. <https://doi.org/10.1145/3550454.3555516>