Review of "Repulsive Curves" paper

David Sahna
ENS Paris-Saclay
Gif-sur-Yvette, France
david.sahna@ens-paris-saclay.fr

Mathis Embit
ENS Paris-Saclay
Gif-sur-Yvette, France
mathis.embit@ens-paris-saclay.fr

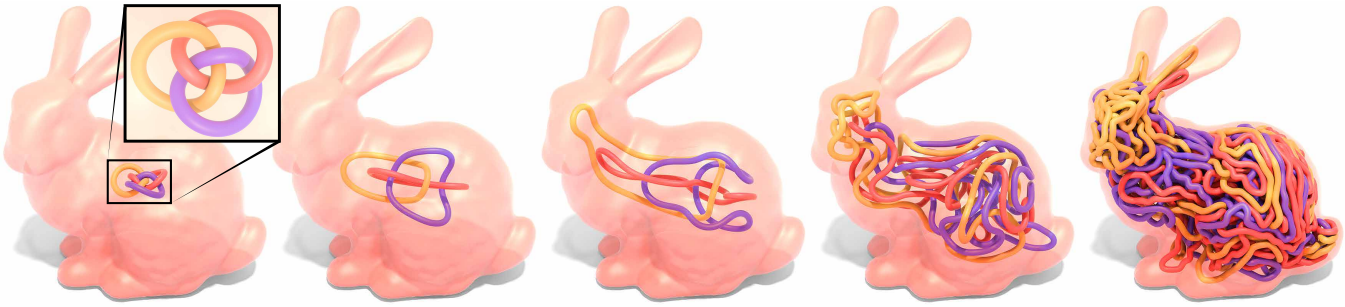


Figure 1: They developed an efficient strategy for optimizing curves while avoiding self-collisions. Here for instance, interwoven curves of increasing length are confined inside a fixed domain, resulting in an intricate "curve packing." Replacing ordinary gradient descent with a specially-tailored *fractional Sobolev gradient* lets us take very large steps toward the solution, enabling rapid design exploration.

ABSTRACT

As part of the MVA Geometric Data Analysis course, we review the paper Repulsive Curves[3]. We start by giving some context to this paper, especially in the framework of geometric data analysis. We explain theoretical and practical interest in curves. After that we describe their curve optimization solution from our perspective and highlight some limitations. Finally we discuss some applications to data analysis.

KEYWORDS

computing methodologies, shape modeling, mathematics of computing, continuous optimization, computational design, shape optimization, curves, knots

ACM Reference Format:
David Sahna and Mathis Embit. 2023. Review of "Repulsive Curves" paper. In *Proceedings of MVA 2023.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

1.1 Context

Interests in curves are multiple. From computer graphics to physical simulation or mathematical visualization, there is a need for curve design. For example, curves allow realistic interaction and path for objects in video games and virtual/augmented reality. They also allow trajectory simulation in physics by adding a time dimension. They may also find themselves useful in knot theory visualizations. However the existing algorithms do not prevent self-intersections. Self-intersection is a key issue in several domains. For example, if curves are trajectories, intersections are collisions. Another point is that having a non intersecting curve can improve network conception at various scale such as biological/medical modelling of the human body. We will come back to examples later in the review. For now the goal of the paper is to design a computationally efficient algorithm that avoid self-collisions. Apart from geometry, an important space is given to optimization and the way we discretize the problem for our computers. What is particularly interesting is that the main computational optimization comes from mathematic theory: by choosing the right inner product for gradient descent we drastically reduce the number of step needed to reach a local minima. The number of step is even independant of the curve resolution (i.e. the number of point of the curve that we are given). We won't cover their other optimization tricks and computational considerations because they are less relevant to the MVA Geometric Data Analysis course.

In mathematics, energy is often used as a concept that represents a quantity associated with a system or configuration. It is commonly employed in optimization problems and variational calculus, where the goal is to find configurations or functions that minimize or maximize the associated energy. In optimization problems, energy can represent a cost or objective function and the optimal solution corresponds to the configuration that achieves the extremum of this energy function. In functional analysis, energy functionals are used to study functions in infinite-dimensional spaces. The minimization of these functionals can lead to the identification of solutions to certain differential equations.

MVA 2023, December 2023, Paris, FR

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1.2 The paper solution

The paper proposes a method for global collision avoidance on curves. Their solution consists in discretizing a repulsive energy on a continuous curve whose gradient quickly reaches globally self-avoidance configurations. The method to follow is to minimize a repulsive energy that automatically finds self-avoidance configurations. A key part of this is to choose the right inner product for the gradient descent. They constructed the Sobolev-Slobodeckij inner product so that we perform a fractional Sobolev-Slobodeckij gradient descent rather that a L^2 gradient descent. Their contributions are:

- a principled discretization of the tangent-point energy
- a novel preconditioner based on the Sobolev-Slobodeckij inner product
- a numerical solver that easily incorporates constraints needed for design
- a Barnes-Hut strategy and hierarchical multigrid scheme for the tangent-point energy that greatly improve scalability

1.3 Our contribution

We summarize the path of ideas that lead to their solution. We also try to point out some limitations and find applications to data analysis. Additionally, here are the new notions that we dedicated effort to understand and get use to while reviewing the paper:

- differential of a functional
- gradient flow
- steepest direction of a function(al) depending on the inner product
- Sobolev spaces

2 CHOOSING THE RIGHT ENERGY

We want to define a function that goes from the space of curves to the real line that measures the "energy" of a curve. The energy function would look at each pair of points on the curve, measure a sort of repulsion force between them and add them. We need the repulsion measure between two points to be large when they are close to each other. That way minimizing the energy will lead to a curve global configuration with no self-intersections.
We will consider curves given by a parameterization $\gamma : M \to \mathbb{R}^3$, where M is made of intervals and/or loops. How can we formulate an energy that prevents self-intersection of γ? From what we said a first idea would be to define a distance between two points of M and double integrate the inverse of that distance:

$$\mathcal{E}(\gamma) = \iint_{M^2} \frac{1}{d(\gamma(x), \gamma(y))}\, dxdy$$

Depending on the distance d, the integral may not be defined. Let's precise the energy definition.
In general we will consider energies of the form

$$\mathcal{E}(\gamma) = \iint_{M^2} k(x,y)\, dx_\gamma dy_\gamma$$

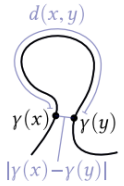
where the kernel $k : M \times M \to \mathbb{R}$ captures the interaction between two points on the curve, and dx_γ denotes the length element on γ. Now let's choose the kernel k. The paper presents two kernel before showing the final one.

2.1 Electrostatic Potential

If we imagine that there is electric charge distributed along γ that pushes it away from itself, a kernel choice would be a Coulomb-like potential

$$\boxed{k_{\text{Coulomb}}(x,y) := \frac{1}{|\gamma(x)-\gamma(y)|^\alpha}} \tag{1}$$

where the parameter α controls the strength of repulsion. According to the paper two drawbacks prevent us from using it for a continuous curve:



- for $\alpha < 2$ it is not strong enough to prevent collisions, allowing the curve to pass through itself
- for $\alpha \geq 1$ the integral does not exist, resulting in unpredictable and unreliable behavior when discretized

2.2 Möbius Energy

A solution to electrostatic potential's problem is to *regularize* the integrand in regions where x approaches y. An improvement is proposed by O'Hara [2] with the *Möbius energy*

$$\boxed{k_{\text{Möbius}}(x,y) := \frac{1}{|\gamma(x)-\gamma(y)|^2} - \frac{1}{d(x,y)^2}} \tag{2}$$

where $d(x,y)$ and $|\gamma(x)-\gamma(y)|$ denotes respectively the shortest distance between x and y along the curve and the euclidean distance. In fact, d(x,y) corresponds to the geodesic distance. We can understand the energy by thinking that

- if two points are close along the curve (hence close in space too), we remove the singular energy since the first and second term will be large in the same way
- if they are close in space but distant along the curve, they continue to repel each other since the second term will be low as long as the geodesic distance go up

The paper decides to do not keep the *Möbius energy* because it is scale invariant, which means a very tight knot has the same energy as the same less tight knot. Therefore the knot can be eliminated if we pull it tight, which do not require more energy. However, it is crucial to notice that scale invariance is not the main limitation. Another concern is the geodesic distance (super important for surfaces), which is expensive to compute and also very challenging to optimize. Hence the *Möbius energy* energy is not well-suited for applications in visual computing.

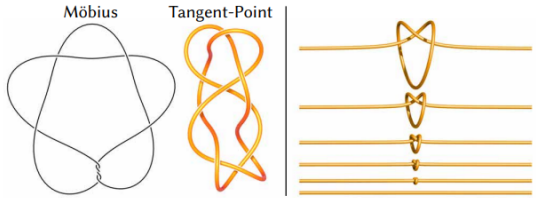


Figure 2: Left: Since the Möbius energy is scale-invariant, it allows "tight spots" where the curve nearly touches itself; such features are avoided by the tangent-point energy. Right: The Möbius energy can likewise artificially eliminate knots by pulling them tight at no energetic cost. (Leftmost image from Kusner and Sullivan [1998].)

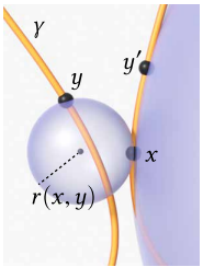
2.3 Tangent Point Energy

Finally they decided to use the tangent point energy:

$$\boxed{\mathcal{E} := \iint_{M^2} \frac{1}{r(\gamma(x),\gamma(y))^\alpha}\, dxdy.} \tag{3}$$

Where $r(x,y)$ is the radius of the smallest sphere tangent to $\gamma(x)$ and passing through $\gamma(y)$ and $\alpha \in \mathbb{R}$ is a parameter controlling the strength of repulsion.
Here is the intuition behind it:

- if $\gamma(x)$ is close to $\gamma(y)$ in space but far from it along the curve, the radius tends to 0 and the energy tends to infinity
- if $\gamma(x)$ is close to $\gamma(y)$ along the curve (hence in space too), the radius tends to infinity and the energy tends to 0



Let's justify that the integral is well-defined:

- If the curve do not contains self-intersection, radius do not reach 0
- If the curve is not a straight line, the radius do not reach infinity

Therefore, if those two conditions are satisfied, the integral is well-defined. As we will see, if we start in such a configuration and we apply gradient descent to this energy, no intersection can appear and the integral remains well-defined. We may worry that the curve turns into a straight line but that would completely minimize the energy, which is what we want. It is not a real problem, for example that would just consists in the very final step of untangling a knot. We just need to be careful with computations.
To develop the mathematics of the gradient descent on the tangent point energy we need to generalize its definition a bit. The useful definition they propose in the paper is

$$\mathcal{E}(\gamma) := \iint_{M^2} k(\gamma(x),\gamma(y),T(x))\, dx_\gamma dy_\gamma$$

where k is the *tangent-point kernel*

$$k(p,q,T) := \frac{|T \times (p-q)|^\alpha}{|p-q|^\beta}. \tag{4}$$

We can now move on to the optimization part and make a choice for the inner product used by the gradient descent. The solution is quiet technical so we will process step by step, introducing all the necessary definitions.

3 OPTIMIZING THE GRADIENT DESCENT

To perform an efficient gradient we will first dive into some mathematics. As the gradient descent will be done over curves, we properly define the derivative of a functional. Moreover, we will place ourselves in a continuous setting and will talk about gradient flow rather then gradient descent.

3.1 Differential of a functional

The notion of the differential of a functional just generalizes the differential of a function. Let $\mathcal{E}$(u) be a functional where u is a real-valued function and let us denote by $d\mathcal{E}(u)$ its differential and δu a small variation of the function u. We say that $\mathcal{E}$ is differentiable at u if the following limit exists :

$$d\mathcal{E}(u)(\delta u) := \lim_{t\to 0} \frac{\mathcal{E}(u+t\delta u) - \mathcal{E}(u)}{t} \tag{5}$$

The above quantity corresponds to the Gâteau differential of $\mathcal{E}$ at the point u in the direction δu. We will come back to that direction in the "Gradient and inner product" part.
This concept differs significantly from the differential of a function because it involves global perturbations, without any hidden local perturbations, as we are manipulating functions in this context. Then, the gradient of $\mathcal{E}$ can be define as the unique function grad $\mathcal{E}$ such that

$$\langle\!\langle \text{grad}\ \mathcal{E}, \delta u \rangle\!\rangle_V = d\mathcal{E}(\delta u) \quad \forall \delta u \tag{6}$$

where $\langle\!\langle \cdot,\cdot \rangle\!\rangle_V$ refers to the inner product in a functional space V.

3.2 Gradient and inner product

The definition of a gradient depends directly on the choice of an inner product structure. The paper *"Generalized Gradients: Priors on Minimization Flows"* [1] does a great job explaining this concept that is too often hidden under the carpet. It helped us understand the idea below.
After defining the Gâteau derivative of the energy we want to know how to move toward the direction of steepest descent of the energy. However, what we call the steepest direction is a function d such that $d\mathcal{E}(u)(d)$ is maximal. We say that the larger $d\mathcal{E}(u)(d)$ is, the steeper the direction d. We clearly see that before talking about steepness we need to provide our space additional structure. That structure is namely an inner product which allows us to talk about angles and lenghts. It is hard to get the intuition in a functional space but we can at least feel the need for an inner product when we want to compare two functions. As for vectors in $\mathbb{R}^n, n \geq 2$, there is no order between the points of a functional space. But thanks to an inner product we can compare their norms, and compute angles.
Therefore, the steepest direction depends on the inner product and our goal is now to maximize

$$d\mathcal{E}(u)(d) = \langle\!\langle \text{grad}_V\ \mathcal{E}, d \rangle\!\rangle_V$$

Each inner product defines its own gradient. Let's see how. We define our new inner product in a functionnal space V as

$$\langle\langle u, v\rangle\rangle_V = \langle\langle Au, v\rangle\rangle_{L^2}$$

where A is a symetric positive definite linear operator.
This means the new gradient, that we will call $g := \text{grad}_V \mathcal{E}$, should satisfy, for any function u,

$$\begin{aligned} d\mathcal{E}(u) &= \langle\langle g, u\rangle\rangle_V \\ &= \langle\langle Ag, u\rangle\rangle_{L^2} \end{aligned}$$

But as we also have $d\mathcal{E}(u) = \langle\langle \text{grad}_{L^2}\mathcal{E}, u\rangle\rangle_{L^2}$. Thus, g need to satisfy

$$\langle\langle Ag, u\rangle\rangle_{L^2} = d\mathcal{E}(u) = \langle\langle \text{grad}_{L^2}\mathcal{E}, u\rangle\rangle_{L^2}$$

Finally our new gradient $\text{grad}_V\mathcal{E}$ is caracterized by being the solution g to the equation

$$\boxed{Ag = \text{grad}_{L^2}\mathcal{E}} \tag{7}$$

Now that we can safely talk about gradients, let's introduce the gradient flow equation.

3.3 Gradient flow

If we want to minimize an energy $\mathcal{E}(u)$, we might follow its gradient flow, which is given by the equation

$$\frac{d}{dt}u = -\text{grad}\ \mathcal{E}(u) \tag{8}$$

Let's understand the main idea behind it. If we quickly recall the iterations of gradient descent, we have:

$$x_{k+1} = x_k - \rho.\text{grad}\ \mathcal{E}(x_k) \quad \forall k \in \mathbb{N} \tag{9}$$

Now, let's assume that $x_k = u(k\rho)$, where $u : \mathbb{R}_+ \to \mathbb{R}$. It follows that $\forall k \in \mathbb{N}$,

$$\begin{aligned} u(\rho + t) = x_{k+1} &= x_k - \rho.\text{grad}\ \mathcal{E}(x_k) \\ &= u(t) - \rho.\text{grad}\ \mathcal{E}(u(t)) \end{aligned}$$

where $t = k\rho$. That is,

$$u(\rho + t) = u(t) - \rho.\text{grad}\ \mathcal{E}(u(t))$$

Hence, after subtracting $u(t)$ and dividing by ρ on both sides of the above equality, we obtain

$$\frac{u(\rho + t) - u(t)}{\rho} = -\text{grad}\mathcal{E}(u(t)) \tag{10}$$

We identify equation (8) as ρ approaches 0 in (10) using the definition of the derivative of a real-valued function. Roughly, the gradient flow is nothing more than the limit of the gradient descent in the case where ρ goes to 0.

Everything is now clear about what the gradient, steepest direction and gradient flow equations are we can move toward the Sobolev inner product.
When the inner product linear operator A is carefully crafted from a Sobolev inner product, it eliminates spatial derivatives (only dt remains), avoiding the stringent time step restriction typically associated with numerical integration of gradient flow.
The choice of A is the most crucial part of the "Repulsive Curves" paper but also the most technical. The technicalities comes from the tangent point energy. That's why they give a warm-up example with the so called Dirichlet energy.

3.4 Dirichlet Energy to motivates the use of Sobolev gradient

Given the complexity suggested by the tangent point energy, a "toy" example we can consider is the Dirichlet energy. This will allow us to understand the main steps for optimizing the gradient descent with fractional Sobolev gradient. We begin by defining the Dirichlet energy, denoted as follows $\mathcal{E}_D : V \to \mathbb{R}$ where V is a functionnal space and,

$$\mathcal{E}_D(u) := \frac{1}{2}\int_\Omega |\nabla(u)|^2 du \tag{11}$$

The idea is, given the gradient under the penalizing integral, to penalize the variations of f. We readily observe that the definition of $\mathcal{E}_D$ involves the L^2 inner product (simply by recalling the definition of the L^2 inner product), which can be rewritten in terms of ∇u and u itself using the first Green's identity, which states that for $V \subseteq \mathbb{R}$, ϕ and ψ scalar functions and ∂V the boundary of V with outward-oriented normal n we have :

$$\int_V (\psi\Delta\phi + \nabla\psi.\nabla\phi)dV = \int_{\partial V} \psi\nabla\phi.ndS \tag{12}$$

If there is no boundary the right-hand side term in (12) is equal to 0. Thus, we have :

$$\begin{aligned} \mathcal{E}_D(u) &:= \frac{1}{2}\int_\Omega |\nabla u|^2 du \\ &= \frac{1}{2}\langle \nabla u, \nabla u\rangle_{L^2} \\ \mathcal{E}_D(u) &= -\frac{1}{2}\langle \nabla u, u\rangle_{L^2} \end{aligned}$$

Thanks to the explicit expression of $\mathcal{E}_D$, we can directly calculate its differential $d\mathcal{E}_D$ by applying the definition of the differential of a functional stated earlier, which yields:

$$\begin{aligned} d\mathcal{E}_D(u)(\delta u) &= \frac{\mathcal{E}_D(u + t\delta u) - \mathcal{E}_D(u)}{t} \\ d\mathcal{E}_D(u)(\delta u) &= -\frac{1}{2}(\langle\langle \Delta\delta u, u\rangle\rangle_{L^2} + \langle\langle \Delta u, \delta u\rangle\rangle_{L^2} + t\langle\langle \nabla u, \delta u\rangle\rangle_{L^2}) \end{aligned}$$

Using the fact that the Laplacian is a self-adjoint operator and the symmetry of the inner product, we can provide the following formula for the differential of the Dirichlet energy. As t tends to 0, the expression becomes:

$$d\mathcal{E}_D(u)(\delta u) = -\langle\langle \Delta u, \delta u\rangle\rangle_{L^2}$$

From

$$\langle\langle \text{grad}_{L^2}\ \mathcal{E}_D, \delta u\rangle\rangle_{L^2} = d\mathcal{E}_D(u)(\delta u)$$

we have, immediately by identification :

$$\text{grad}_{L^2}\ \mathcal{E}_D = -\Delta u \tag{13}$$

Then, it follows in the equation of the gradient flow that :

$$\frac{d}{dt}u = -\text{grad}_{L^2}\ \mathcal{E}(u) = \Delta u$$

that is,

$$\boxed{\frac{d}{dt}u = \Delta u} \quad (Heat\ flow) \tag{14}$$

In attempting to solve this equation using explicit finite differences with a grid spacing h, a stable solution requires a time step of size $O(h^2)$, leading to a considerable slowdown in computation as the

grid is refined. To overcome this time step limitation, an alternative inner product can be employed to define the gradient.
Now, let us take a look at the H^1 inner product.

$$\langle\!\langle \text{grad}_{H^1}\ \mathcal{E}_D, \delta u \rangle\!\rangle_{H^1} = d\mathcal{E}_D(u)(\delta u)$$

we can write,

$$\langle\!\langle \nabla \text{grad}_{H^1}\ \mathcal{E}_D, \delta u \rangle\!\rangle_{L^2} = \langle\!\langle \Delta u, \delta u \rangle\!\rangle_{L^2}$$

Thus:

$$\text{grad}_{H^1}\ \mathcal{E}_D = u \tag{15}$$

Hence, the H^1 gradient flow is given by:

$$\boxed{\frac{d}{dt}u = -u} \quad (scaling\ flow) \tag{16}$$

Unlike the L^2 inner product, the H^1 inner product is **well-suited** for the minimization problem, eliminating bumps, and allowing the Dirichlet energy to reach 0 (Figure 4). This is because, in this case, there are no spatial derivatives involved. The problem reduces to solving a simple system of ODEs, and there is no longer a need to deal with PDEs, which is fantastic from a computational standpoint. From now on, it is legitimate to ask whether using an inner product involving many derivatives can lead to better convergence. We will see that no. Here, using

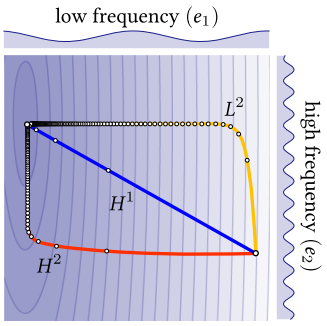


Figure 3: Gradient flows projected onto a low- and high-frequency mode e_1, e_2, resp. Notice that poor preconditioning leads to slow convergence.

$$\langle\!\langle \text{grad}_{H^2}\ \mathcal{E}_D, \delta u \rangle\!\rangle_{H^2} = d\mathcal{E}_D(u)(\delta u)$$

we can write,

$$\langle\!\langle \Delta^2 \text{grad}_{H^2}\ \mathcal{E}_D, \delta u \rangle\!\rangle_{L^2} = -\langle\!\langle \Delta u, \delta u \rangle\!\rangle_{L^2}$$

Thus:

$$\text{grad}_{H^2}\ \mathcal{E}_D = -\Delta^{-1} u \tag{17}$$

Hence, the H^2 gradient flow is given by :

$$\boxed{\frac{d}{dt}u = \Delta^{-1}u} \quad (inverse\ heat\ flow) \tag{18}$$

The flow described faces challenges in integration and struggles to smooth out high frequencies (Figure 4). In general, achieving favorable behavior requires careful selection of a Sobolev inner product, rather than blindly picking one, to match it with the energy.

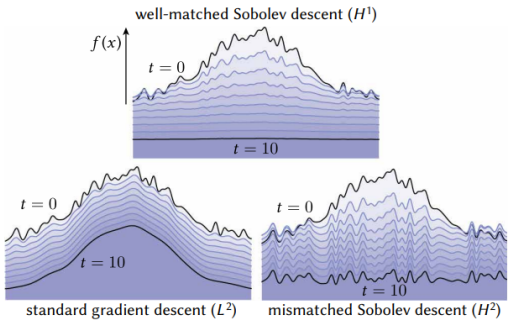


Figure 4: For Dirichlet energy, which penalizes variations in a function f (x), standard L^2 gradient descent mostly smooths out local features (bottom left), whereas an inner product that is too high-order has trouble removing high frequencies (bottom right). A Sobolev descent that is well-matched to the order of the energy yields rapid progress toward a local minimizer (top). We apply a similar strategy to quickly optimize the shape of curves.

3.5 Fractional Sobolev Gradient

Now, our goal is to compute the gradient of $\mathcal{E}^\alpha_\beta$. To do so we need an inner product. The inner constructed in the paper is the one of the space called H^s_γ where $s = \frac{\beta-1}{\alpha}$. Let's try to see how they made it up.
To exactly know the order of the differential of the tangent point energy without computing the gradient, they detailed some analysis in an extended Sobolev space called Sobolev-Slobodeckij space, and found that $d\mathcal{E}^\alpha_\beta(f)$ is a differential operator reducing the differentiability of its arguments by $2s$. All these calculations are available in the Appendix A of the "Repulsive Curves" paper.
Once we know that the order is $2s$ we can start the construction of the adapted inner product. The paper first proposes the linear operator B_σ defined by the quantity $\langle\!\langle B_\sigma u, v \rangle\!\rangle$ being equal to

$$\iint_{M^2} \frac{\mathcal{D}u(x) - \mathcal{D}u(y)}{|\gamma(x)-\gamma(y)|^\sigma} \frac{\mathcal{D}v(x)-\mathcal{D}v(y)}{|\gamma(x)-\gamma(y)|^\sigma} \frac{dx_\gamma dy_\gamma}{|\gamma(x)-\gamma(y)|} \tag{19}$$

where $\mathcal{D}u := du\ d\gamma^T/|d\gamma|^2$ the usual derivative of u along M expressed as a vector in $\mathbb{R}^3$ tangent to γ. We need to multiply by the factor $1/|d\gamma|^2$ because γ is not arc-length parameterized in general. The problem is B_σ is only semi definite and no invertible the same way $-\delta$ was for Dirichlet. It means that the inner product will be 0 for functions that are constant over each component of the domain M. In Dirichlet case we added Id to $-\Delta$, here we add B^0_σ, a low-order term defined by the quantity $\langle\!\langle B^0_\sigma u, v \rangle\!\rangle$ being equal to

$$\iint_{M^2} k_4^2(\gamma(x), \gamma(y), T(x)) \frac{(u(x)-u(y))(v(x)-v(y))}{|\gamma(x)-\gamma(y)|^{2\sigma+1}} dx_\gamma dy_\gamma \tag{20}$$

Interesting fact: this low-order term also helps to keep steady motions during gradient descent, resulting in a faster evolution of the flow.
Finally $A_\sigma = B_\sigma + B^0_\sigma$ is the final linear operator inducing the wanted $2s$ order inner product.

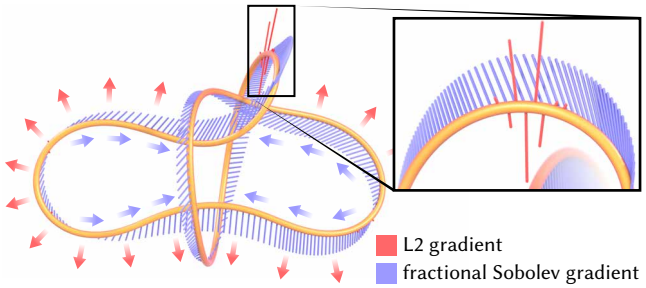


Figure 5: Since an L^2 gradient flow is always perpendicular to the curve (red), it fails to resolve even simple cases like the one shown above, where a large near-tangential motion is needed to untangle a knot. The fractional Sobolev gradient (blue) permits such motions, yielding a far more efficient flow.

To sum up, we went through the same process as for Dirichlet. For Dirichlet it was:

(1) use a Sobolev gradient $\text{grad}_{H^k}\mathcal{E}$
(2) find the right k to eliminates the spatial derivatives
(3) add low-order terms to make A invertible

For the tangent point energy it was:

(1) use a Sobolev inspired gradient $\text{grad}_{H^s_\gamma}\mathcal{E}$
(2) choose a differential operator B_σ of the same order as the differential $d\mathcal{E}^\alpha_\beta$
(3) add a lower-order operator B^0_σ that makes the overall operator $A_\sigma := B_\sigma + B^0_\sigma$ more well-behaved

The new inner product is the **Sobolev-Slobodeckij inner product** defined as

$$\boxed{\langle\!\langle u, v\rangle\!\rangle_{H^s_\gamma} := \langle\!\langle A_\sigma u, v\rangle\!\rangle_{L^2}}$$

We finally reach our goal, the **Sobolev-Slobodeckij gradient** defined by the previous fractional inner product:

$$\boxed{\text{grad}_{H^s_\gamma}\mathcal{E}^\alpha_\beta = \bar{A}^{-1}_\sigma \text{grad}_{L^2}\mathcal{E}(\gamma)} \tag{21}$$

where $\bar{A}_\sigma$ corresponds to the componentwise application of A_σ.

We choosed to spend more time on the Dirichlet example rather than on the tangent point energy differentiation because the calculations were heavy without helping to understand the key concept. The interesting thing to remember from all this is the ability geometry gave us to free oneself from the spatial derivative in the gradient flow, resulting in a faster algorithm. The theoretical point of view has a significant impact on the real world implementation of the algorithm.

4 CONCLUSIONS

4.1 How to put that in a computer?

The next step is to discretize this gradient flow to an actual algorithm to be numerically implemented. The paper present additionnal discretization optimizations, but it would need considerable further explainations and it's not the point of our Geometric Data Analysis class so we won't review it.

4.2 Limitations

- *Approximation of Energy via Quadrature*: Using quadrature to guess energy is a bit risky—it doesn't promise a collision-free ride. Maybe trying out more solid methods like Continuous Collision Detection (CCD) or getting an exact limit could make collision prevention more bulletproof.
- *Acceleration Opportunities*: The article spots some easy wins to speed up the math. Playing around with parallelization, tweaking the mesh on the fly, and trying out multigrid methods, especially for surfaces, could seriously speed up the number crunching.
- *Collision-Avoiding Interpolation/Animation*: The article sort of avoids the question of how to animate without causing crashes. It's like an open playground for future exploration, maybe making the whole collision-handling thing work even better.

So, the article says "be careful" in a few places, but it also throws out some good ideas for future work that could make things smoother and simpler.

4.3 Applications to data analysis

Writting this review for our Geometric Data Analysis course, we felt the need to provide some direct application of repulsive curves. We can think of two major applications: 2D and 3D.

Before giving some of them, it's important to note that the optimization framework discussed in the paper is not exclusively tailored for efficiently optimizing knot energies. We have the flexibility to incorporate fairly arbitrary user-defined constraints and penalties during the minimization process. For instance, we can think of a barycenter, an edge, an edge length, a point constraint, a surface constraint, a tangent constraint, a total length, a length difference, a surface and a field potential depending on the application.

4.3.1 *2D repulsive curves for data analysis.*

- *Segmentation of images*: We want to distinguish the form of each object more accurately in the image during segmentation. Thus, we create curves that repel from the limits of regions of interest. It might be crucial in medical applications such as brain MRI tissue segmentation, tumor segmentation in medical imaging...
- *Graph drawing*: The limitations of traditional 2D graph drawing algorithms that rely on nodal proximity, leading to issues like crossed edges or nodes positioned very closely together. It highlights the alternative approach of treating edges as repulsive curves, aiming to produce more compact and readable graph layouts.

4.3.2 *3D repulsive curves for data analysis.*

- *Cable packing*: We want to optimize the arrangement of cables within a confined space, aiming to maximize cable density. This approach leverages the repulsive nature of curves to efficiently organize and pack cables, providing an optimal solution for cable management in restricted environments.

REFERENCES

[1] Guillaume Charpiat, Pierre Maurel, J.-P Pons, Renaud Keriven, and Olivier Faugeras. 2007. Generalized Gradients: Priors on Minimization Flows. *International Journal of Computer Vision* 73 (07 2007), 325–344. https://doi.org/10.1007/s11263-006-9966-2

[2] J. O'Hara. 1991. Energy of a knot. *Topology* 30 (1991), 241–247.

[3] Chris Yu, Henrik Schumacher, and Keenan Crane. 2021. Repulsive Curves. *ACM Trans. Graph.* 40 (2021), 1.

Received 10 December 2023