Geometric Flows in Python

Alejandro Escoto
Mentor: Vlassis Mastrantonis

University of Maryland, College Park

Fall 2023

Overview

1. Review of Plane Curve Geometry
2. Curve Shortening Flow
3. In Python
4. Review of Surface Geometry
5. Gauss Curvature Flows

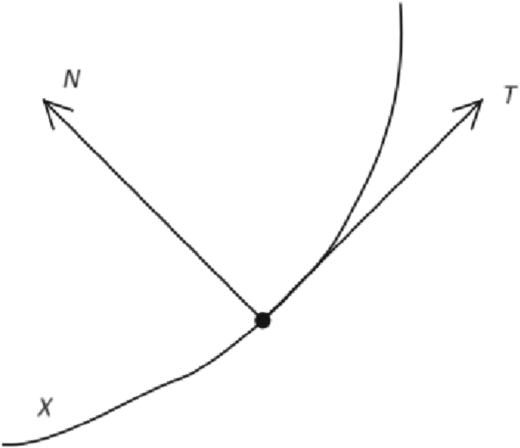
Review of Plane Curve Geometry

Curve shortening flow is a plane curve theory so we review the basics

Definitions

- A *Immersed Planar Curve* is a smooth map $X : I \to \mathbb{R}^2$ satisfying the condition $|X'(u)| \neq 0$ for $u \in I$
- The *Unit Tangent Vector* T is the derivative of an immersed planar curve with respect to an arclength parameter given by $T = \frac{dX}{ds} = \frac{X'}{|X'|}$
- Rotating the unit tangent vector counter clockwise by $\frac{\pi}{2}$ gives us the *Unit Normal Vector* $\mathbf{N}$

Review of Plane Curve Geometry



Review of Plane Curve Geometry

Differentiating N and T gives us our last invariant, *Curvature*.

Definition

The *curvature*, κ of a plane curve is defined by the *Frenet-Serret equations*:

$$\frac{d\mathbf{N}}{ds} = \kappa\mathbf{T}$$

$$\frac{d\mathbf{T}}{ds} = -\kappa\mathbf{N}$$

Intuitively, this tells us how much the unit normal changes

Review of Plane Curve Geometry

Note that since $\mathbf{T}$ is a unit vector, we can describe it via it's *turning angle*, ϕ

$$\mathbf{T} = (cos\phi, sin\phi)$$

so, differentiating yields the following

$$\frac{d\mathbf{T}}{ds} = \frac{d\phi}{ds}(-sin\phi, cos\phi)$$

via Frenet-Serret, we get the nice relation

$$\kappa = \frac{d\phi}{ds}$$

Curve Shortening Flow

Now, let $X : M^1 \times [0, T) \to \mathbb{R}^2$ be a smooth map which is an immersion for each $t \in [0, T)$

We say $X(t)$ is a solution to the *Curve-Shortening Flow* if

$$\frac{\partial \mathbf{X}}{\partial t}(u, t) = -\kappa(u, t)\mathbf{N}(u, t)$$

or, via the Frenet-Serret equations:

$$\frac{\partial \mathbf{X}}{\partial t}(u, t) = \frac{\partial^2 \mathbf{X}}{\partial s^2}(u, t)$$

Curve Shortening Flow

Now for some results

Theorem

suppose $X_i : M_i^1 \times [0, T) \to \mathbb{R}^2$ *be solutions to the curve shortening flow satisfying* $X_1(M_1^1, 0) \cap X_2(M_2^1, 0) = \emptyset$ *then* $X_1(M_1^1, t) \cap X_2(M_2^1, t) = \emptyset$ *for each* $t \in [0, T)$

Theorem

let $X_0 : M^1 \to \mathbb{R}^2$ *be a smooth embedding of* M^1 *then the solution to curve shortening flow with initial data* X_0 *exists on a maximal time interval* $[0, T)$ *and converges to a point in* $\mathbb{R}^2$

Famously, curve shortening flow was used in a modified version of Hamilton's program by Perelman to prove the Poincare Conjecture in 2003

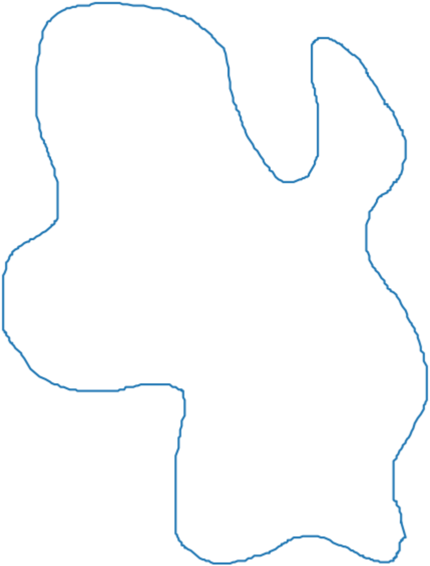
In Python

Goal: Implement curve shortening flow in Python

Steps:

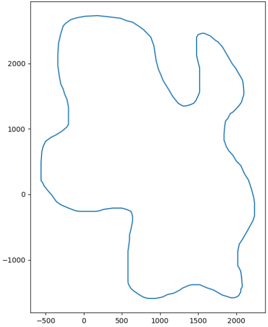
1. Approximate a curve with finite number of points
2. Compute invariants at each point
3. Use Euler's method to integrate the curve shortening flow equation

In Python

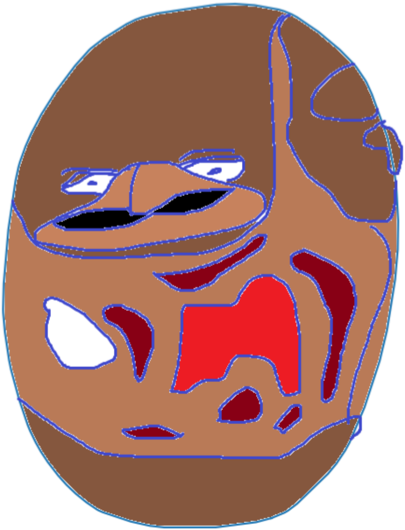


In Python

In Python



In Python



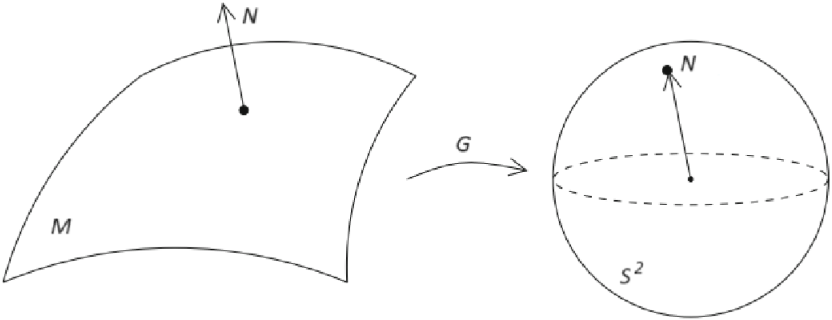
Review of Surface Geometry

We want to generalize the idea of curvature flow to surfaces, so we review some surface geometry

Definitions

- let M^n be a smooth n-dimensional manifold and let $X : M^n \to \mathbb{R}^{n+1}$ be a smooth immersion. Then we regard the image $M := X(M^n)$ to be a *hypersurface* in $\mathbb{R}^{n+1}$
- we define the *Gauss map* to be the map $G : M \to S^n$ taking points $p \in M$ to it's corresponding unit normal vector in $N \in S^n$

Review of Surface Geometry



Review of Surface Geometry

Differentiating the Gauss map at a point, gives us the *Weingarten map* given by the linear map

$$-W_{p,M} := D_pG : T_pM \to T_{G(p)}S^2$$

Intuitively, this tells us how the unit normal changes at a point and gives rise to a curvature for hyersurfaces

Definition

The *Gaussian curvature* of a hypersurface M at the point $p \in M$ is given by $K := det(W_{p,M})$

With this we introduce the Gauss curvature flow

Gauss Curvature Flows

A smooth 1-parameter family $\mathbf{X} : M^n \times I \to \mathbb{R}^{n+1}$ satisfies the α-*Gauss curvature flow* if it satisfies

$$\partial_t \mathbf{X} = -sign(\alpha) K^{\alpha} \mathbf{N}$$

Where K is the Gauss curvature of the family.

When $\alpha = 1$, we call this flow the *Gauss curvature flow*, which is a natural generalization of curve shortening flow

Gauss Curvature Flows

For uniformly convex solutions, in particular, we get a nice theorem regarding convergence

Theorem

Given $\alpha > 0$ and a smooth uniformly convex hypersurface M of $\mathbb{R}^{n+1}$, then there exists a unique smooth solution $\{M_t\}_{t\in[0,T)}$ with initial data M which remains uniformly convex on [0,T) and shrinks to a point as $t \to T$

However, this does not ensure convergence to a *round* point!

Gauss Curvature Flows

An improtant integral quantity associated with a hypersurface is the *Gaussian Entropy* given by

$$E_\alpha(M) = \left(\frac{Vol(M)}{|B^{n+1}|}\right)^{\frac{n}{n+1}} \left(\frac{1}{|S^n|}\int_M K^\alpha d\mu\right)^{\frac{1}{\alpha-1}}$$

Which characterizes regularity of M

Gauss Curvature Flows

an interesting case arises when $\alpha = \frac{1}{n+2}$

Theorem

Let $X : M \times [0, T) \to \mathbb{R}^n + 1$ be a smooth uniformly convex solution to the $\frac{1}{n+2}$-Gauss curvature flow. Given $L \in SL(n+1, \mathbb{R})$, there is a family of diffeomorphisms ϕ such that $\tilde{X}(x, t) = L(X(\phi(x, t), t))$ is also a solution to the flow

In other words, the $\frac{1}{n+2}$ flow is affine invariant

Gauss Curvature Flows

We're interested in the *self-similar shrinking solutions* to the affine normal flow.

Proposition

let M be a smooth uniformly convex hypersurface. Then M is a critical point of $E_{\frac{1}{n+2}}$ iff M is a self-similar solution to the affine normal flow

Proposition

Any smooth uniformly convex hypersurface which is a critical point of $E_{\frac{1}{n+2}}$ is an ellipsoid

Gauss Curvature Flows

Piecing together the last two propositions we get a nice theorem

Theorem

The self-similar solutions to the affine normal flow are the ellipsoids which are critical points of $E_{\frac{1}{n+2}}$

With this condition we can discern whether the affine normal flow convergence to a round point!

The End