Homework 1: Discrete and Smooth Curves

Due February 27, 2019

> This is the first homework assignment for 6.838. Check the course website for additional materials and the late policy. You may work on assignments in groups, but every student must submit their own write up; please note who you worked with on your write up. Some of the notation used in this homework may be unfamiliar or rusty for computer science students, but undergrad calculus should be sufficient to answer all the problems. Get started early, and reach out for help during office hours and/or on Piazza—we will be generous!

Problem 1 (30 points)**.** In this problem, we introduce continuous and discrete methods in variational calculus, one of the main tools of the differential geometry toolbox.

(a) Suppose you are given a regular plane curve $\gamma : [0,1] \to \mathbb{R}^2$, and take $\mathbf{v} : [0,1] \to \mathbb{R}^2$ to be a vector field along γ. Recall that the arc length of γ is given by

$$s[\gamma] = \int_0^1 \|\gamma'(t)\|_2 \, dt.$$

We can think of $\gamma_h(t) := \gamma(t) + h\mathbf{v}(t)$ to be a displacement of γ along $\mathbf{v}$. Differentiate $f(h) := s[\gamma + h\mathbf{v}]$ with respect to h at $h = 0$ to yield an expression for $\frac{d}{dh} s[\gamma + h\mathbf{v}]|_{h=0}$.

Hint: The "differentiation under the integral sign" rule shows $\frac{d}{dh} \int_a^b g(t,h)\, dt = \int_a^b \frac{\partial g}{\partial h}(t,h)\, dt$.

(b) You can think of each $\mathbf{v}$ as an infinitesimal displacement (a "variation") of the entire curve γ at once. Explain how the derivative you took in (a) can be thought of as a directional derivative of arc length in the $\mathbf{v}$ "direction." In variational calculus, this derivative is known as the Gâteaux or variational derivative of $s[\cdot]$.

(c) Suppose $\mathbf{v}(0) = \mathbf{v}(1) = \mathbf{0}$. Define a vector-valued function $\mathbf{w}(s)$ so that

$$\left.\frac{d}{dh} s[\gamma + h\mathbf{v}]\right|_{h=0} = \int_0^{s(1)} \mathbf{v}(s^{-1}(\bar{s})) \cdot \mathbf{w}(\bar{s})\, d\bar{s},$$

where $s(t) = \int_0^t \|\gamma'(\bar{t})\|_2 \, d\bar{t}$ on the right-hand side is the arc length function and $\mathbf{w}$ can be written in terms of the curvature and Frenet frame of γ.

Note: Your formula for $\mathbf{w}$ should *not* include any term involving $\mathbf{v}$.

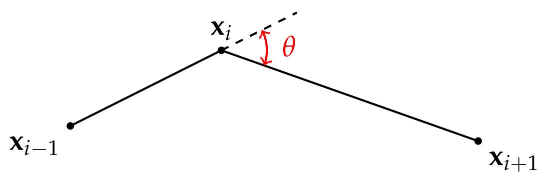
Hint: Simplify the left-hand side using your answer to (a). Use integration by parts.

> For coding assignments, you can use either Julia or MATLAB as your programming language. The starter code handles visualization/problem setup and indicates where you should fill in your solutions.

Problem 2 (35 points)**.** In this problem, you will develop a notion of discrete curvature of a plane curve.

(a) Suppose we have a discrete curve given by a series of points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^2$. You can think of the vertex positions as parameterized by a vector $\mathbf{x} \in \mathbb{R}^{2n}$. Define an arc length functional $s(\mathbf{x}) : \mathbb{R}^{2n} \to \mathbb{R}_+$; for convenience, it is acceptable to notate $s(\mathbf{x}) = s(\mathbf{x}_1, \ldots, \mathbf{x}_n)$.

(b) Suppose $1 < i < n$. Write an expression for the gradient $\nabla_{\mathbf{x}_i} s$ of $s(\cdot)$ with respect to $\mathbf{x}_i$ and show that its norm is $2 \sin \frac{\theta}{2}$, where θ is the turning angle between the two segments adjacent to $\mathbf{x}_i$ (see figure).



(c) Take a look at `discreteCurve.m` (or `discreteCurve.jl`). The curve generates an $n \times 2$ array representing n points on a discrete two-dimensional curve. Modify the code to plot the derivative you computed in part (b).

(d) Propose a measure of discrete (unsigned) per-vertex curvature of a 2D discrete curve based on your answers to 1(c) and 2(b), and draw the curve colored by this value. For this, fill in code to compute `kappa` in `discreteCurve.m` (`discreteCurve.jl`).

Note: Make sure that your notion of curvature does not go to zero as you increase the number n of samples. Multiple answers are possible.

(e) For sufficiently small $h > 0$, one simple way to decrease the length of the curve would be to replace each point $\mathbf{x}_i$ with a new point $\mathbf{x}_i' := \mathbf{x}_i - (\nabla_{\mathbf{x}_i} s)h$, where $\nabla_{\mathbf{x}_i} s$ is the derivative of s with respect to $\mathbf{x}_i$ (make sure you understand why!). Implement this forward integration scheme with the endpoints fixed, and make sure that if you iterate enough times the curve approximates a straight line. What happens if h is too large?

Problem 3 (35 points)**.** In this problem, you will implement part of the "Discrete Elastic Rods" paper discussed in class. Take a look at `bishopFrame.m` (or `bishopFrame.jl`) for starter code.

(a) Add code to compute the $(n-2) \times 3$ array `binormal`, which contains the Darboux vector $(\kappa \mathbf{b})_i$ for each vertex i except the first and last.

(b) We provide simple code for animating different initial choices of the Bishop frame $(\mathbf{u}, \mathbf{v}, \mathbf{t})$ on the first segment. Add code to fill in $\mathbf{u}$ and $\mathbf{v}$ along the rest of the curve.

(c) **Challenge problem** (5 points of the 35). In the final part of the code, we prescribe a material frame on the curve. Add code to compute the twist energy and its gradient. Perform gradient descent on the angles between the material frame and the Bishop frame and observe how the material frame "untwists." Make sure that if you run your code long enough the material frame aligns with the natural Bishop frame.