Homework 2: Discrete and Smooth Surfaces

Differential Geometry for Computer Science (Spring 2013), Stanford University

Due Monday, May 6, in the course mailbox

Problem 1 (15 points). *Let $\gamma : I \to \mathbb{R}^3$ be a smooth regular curve parametrized by arc-length and suppose $N(t)$ is a smooth choice of a unit-length vector at $\gamma(t)$ that is orthogonal to $\dot{\gamma}(t)$. Let $f : I \to \mathbb{R}$ be a smooth function. The* tubular surface *around γ with radial function f is defined by the parametrization $\phi(\theta, t) := \gamma(t) + f(t)\cos(\theta)N(t) + f(t)\sin(\theta)\dot{\gamma}(t) \times N(t)$.*

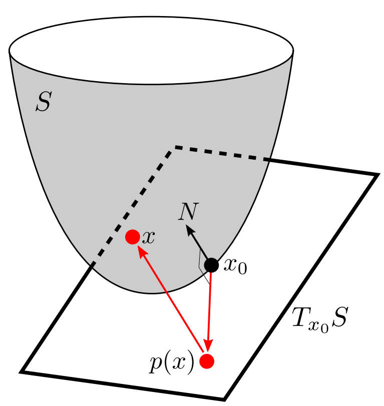
(a) Draw a picture of this set-up.

(b) A torus is the surface of revolution obtained by rotating around the z-axis a vertical circle of radius r whose center is located a distance R from the z-axis. Parametrize this torus in the form of part (a).

(c) Find a basis for the tangent space of the torus, as well as expressions for its outward-pointing unit normal vector and matrix of the second fundamental form with respect to this basis and normal vector.

Problem 2 (15 points). *Let $f : \mathbb{R}^2 \to \mathbb{R}$ be a twice-differentiable function and consider its graph $S := \{(x, y, f(x, y)) : x, y \in \mathbb{R}^2\}$. Find expressions for a basis for the tangent space of S, the unit upward-pointing normal vector of S and the matrix of the second fundamental form with respect to this basis and normal vector.*

Problem 3 (30 points). *In differential geometry we have the ability to construct special parametrizations that are well-adapted to a given geometric setting. Often, this is the key step in the proof of a theoretical result. In this problem, you'll show that* a regular surface is locally the graph over its tangent plane.



- *x_0 is a point on the surface S.*
- *$T_{x_0}S$ is its tangent plane.*
- *N is the unit normal vector at x_0.*
- *x is another point on S near x_0.*
- *$p(x)$ is the orthogonal projection of x onto the plane $T_{x_0}S$.*

What this means is that we can find $\mathcal{V} \subseteq \mathbb{R}^2$ containing the origin and a function $f : \mathcal{V} \to \mathbb{R}$ so that the graph $\{(v^1, v^2, f(v^1, v^2)) : (v^1, v^2) \in \mathcal{V}\}$ represents the surface S in the following sense: all points x near x_0 can be parametrized in the form $x(v^1, v^2) := x_0 + E_1 v^1 + E_2 v^2 + f(v^1, v^2)N$ where E_1, E_2 is *a basis for $T_{x_0}S$ and N is normal to $T_{x_0}S$ with $\|N\| = 1$. Furthermore we require that $f(0,0) = 0$ and $\frac{\partial f(0,0)}{\partial v^1} = \frac{\partial f(0,0)}{\partial v^2} = 0$, the consequence of which is that $x(0,0) = x_0$ and $\frac{\partial x(0,0)}{\partial v^1} = E_1$ and $\frac{\partial x(0,0)}{\partial v^2} = E_2$.*

The essence of the proof below is to find a form for the function f in terms of the only thing you're allowed to assume about the surface S near x_0, namely the existence of a parametrization for a neighbourhood of x_0. This is a hard problem, but is an archetype for many differential geometric constructions. If you can master this problem, then you are well on your way to being an expert!

(a) *We'll start by writing $x - x_0 = p(x) + a(x)N$ where $p(x) \perp N$. Find $a(x)$ in terms of x, x_0 and N.*

(b) *Let E_1, E_2 be any basis for $T_{x_0}S$. Find a formula for $p(x)$ in terms of x_0, E_1, E_2 that looks like $p(x) := p^1(x)E_1 + p^2(x)E_2$. So in other words, you have to find formulas for $p^1(x)$ and $p^2(x)$. There will be a matrix involved that you'll have to invert — explain why it's invertible.*

(c) *If we now let $v^1 := p^1(x)$ and $v^2 := p^2(x)$ and somehow we manage to invert these equations to find $x = q(u)$ for some function q, we now have an expression of the form $x - x_0 = v^1E_1 + v^2E_2 + a(q(v))N$. This is exactly what we want, with $f(v) := a(q(v))$. Why can't we do this?*

(d) *Here's a way around the impasse of part (c). Introduce a parametrization for a neighbourhood of x_0, i.e. let $\phi : \mathcal{U} \to \mathbb{R}^3$ parametrize S near x_0, with $\phi(0) = x_0$ without loss of generality. Let $E_1 := \frac{\partial \phi(0,0)}{\partial u^1}$ and $E_2 := \frac{\partial \phi(0,0)}{\partial u^2}$. Let $P : \mathcal{V} \to \mathbb{R}^2$ be the function $P(u^1, u^2) := (p^1(\phi(u^1, u^2)), p^2(\phi(u^1, u^2)))$. Now we have a relationship $v = P(u)$. Find the derivative DP_u at $u = 0$ and show it's invertible.*

(e) *Quote the Inverse Function Theorem correctly and argue that we have an inverse function $P^{-1} : \mathcal{V} \to \mathcal{U}'$ defined on some sets $\mathcal{V}$ and $\mathcal{U}' \subseteq \mathcal{U}$. We can now complete part (c) rigorously — what is the final form for $f : \mathcal{V} \to \mathbb{R}$?*

(f) *Show that $f(0,0) = 0$ and $\frac{\partial f(0,0)}{\partial v^1} = \frac{\partial f(0,0)}{\partial v^2} = 0$. You'll need the formula for DP^{-1} in terms of DP provided by the Inverse Function Theorem at some point in this calculation.*

Problem 4 (20 points)**.** *Many geometric operators assume that a mesh is* oriented*, meaning that the underlying surface is orientable and that the ordering of the vertices (clockwise vs. counterclockwise) in each triangle is consistent. Complete `assignCoherentOrientation.m` to take a list of triangles with arbitrary orientation and generate a new list where the triangles are consistently oriented. Note that globally there are two acceptable orientations (inward normal or outward normal); leave the orientation of the first triangle the same. The script `problem4.m` provides some code for testing your method.*

Problem 5 (20 points)**.** *In class, we discussed "boundary operators" ∂ for an oriented triangle mesh with vertices V, edges E, and triangles T. While finding the boundary of a simplex takes you down one dimension (from triangles to edges to vertices), we can define an operator d that does the opposite. In particular, d will take one value per k-simplex and return one value per $(k+1)$-simplex (a triangle is a 2-simplex, an edge is a 1-simplex, and a vertex is a 0-simplex). We will represent d using two matrices:*

- $d_{0\to1} \in \mathbb{R}^{|E|\times|V|}$ *will take one value per vertex and return one value per edge. You must find a list of edges E and assign each an orientation (unlike the halfedge structure, we won't double edges; just assign each a single arbitrary orientation). The row of this matrix corresponding to edge $e = v_1 \to v_2$ takes the difference of the values $f(v_2) - f(v_1)$.*

- $d_{1\to 2} \in \mathbb{R}^{|T|\times|E|}$ *combines edge values to triangles with* $+1$ *when the edge's orientation coincides with that of the triangle and* -1 *otherwise.*

We'll proceed in two parts:

(a) *Complete* `boundaryOperators.m` *to find the matrices of these two operators. To receive credit, these operators must be* sparse *matrices, meaning that most of their elements are zero. Check out Matlab's* `sparse` *method for how to construct a sparse matrix. The script* `problem5.m` *provides some testing material. As a challenge, try to write this method without using any loops, which are slow in Matlab.*

(b) *Show that* $d_{1\to 2}d_{0\to 1} \equiv 0$.