Repulsive Shape Optimization for Modeling Complex Geometries

ME 6104 Final Project
4/21/2022

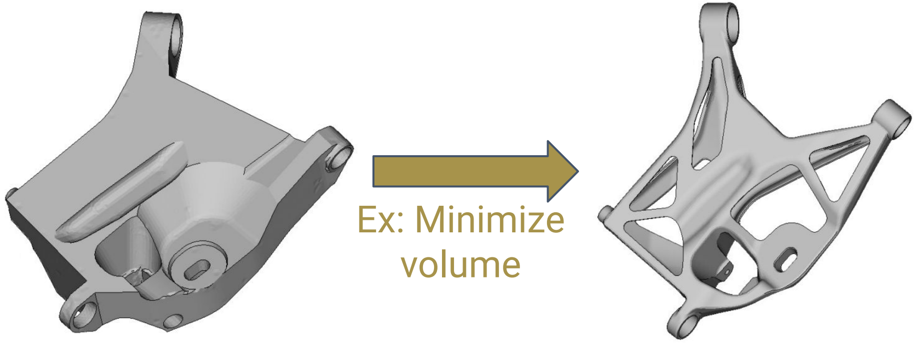
Professor: Dr. Yan Wang
Team Members: Ryan Grajewski & Paul Babou

Georgia Tech

Introduction to Repulsive Shape Optimization

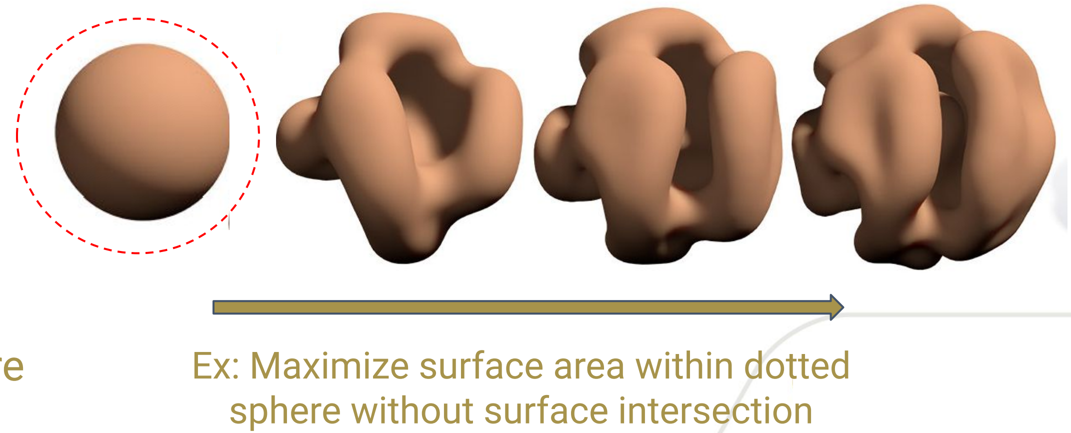
First: The Traditional Optimal Control Problem

- Max/Minimize a desired parameter (*cost functional*) while satisfying constraints



Now: Repulsive Optimization

- Same concept, but…
- Additionally prohibits curve and surface self-intersections



Benefits:

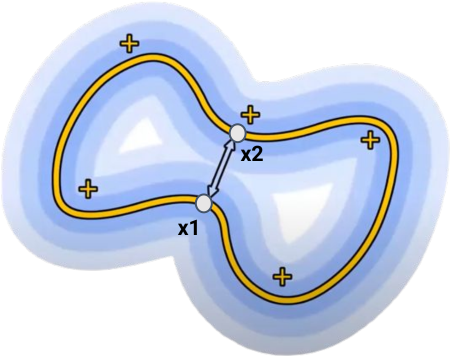
- Accurately modeling geometries/shapes found in nature

Various Repulsive Energies Available - tradeoffs

Coulomb/Electrostatic Potential

- Analogous to an electric charge being distributed along a curve

$$\epsilon_{coulomb} = \iint \frac{1}{|\gamma(x_1) - \gamma(x_2)|^{\alpha}}$$



Möbius Energy

- Ignores high energies between adjacent points along a closed curve
- Möbius Transformation Invariant
  - cannot distinguish “tightly wound” points

$$\epsilon_{Möbius} = \iint \frac{1}{|\gamma(x_1) - \gamma(x_2)|^2} - \frac{1}{[D(x_1, x_2)]^2}$$

Shortest distance between x1 and x2 *along* the curve

Tangent-Point Energy

- Ignores high energies between adjacent points
- Not Möbius invariant (for $\alpha > 2$)
- More robust

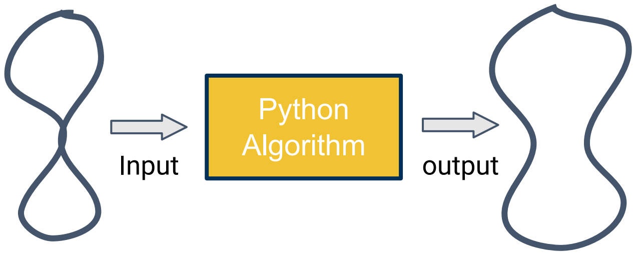
$$\epsilon_{Tangent-Point} = \iint \frac{1}{r(x_1, x_2)^{\alpha}}$$

Georgia Tech

Project Formulation & Goal

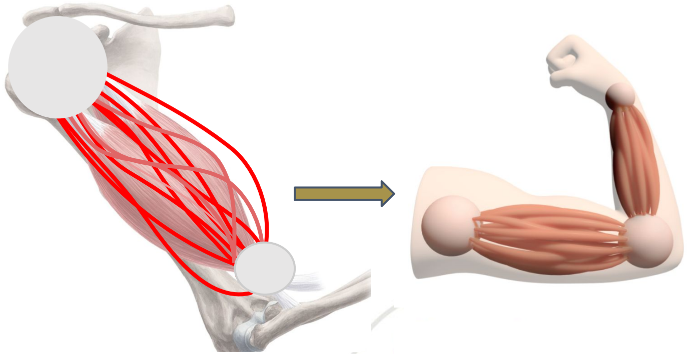
Objective: Develop a python algorithm that:

- Implements the tangent-point energy function
- Calculates the energy value at each point along a given closed-curve
- Optimizes the given input curve by translating certain points so that all energies are minimized



Ultimate Motivation:

- Produce a simplified 3D model of bicep muscle fibers to demonstrate ability to depict natural geometries



Tangent-Point Energy

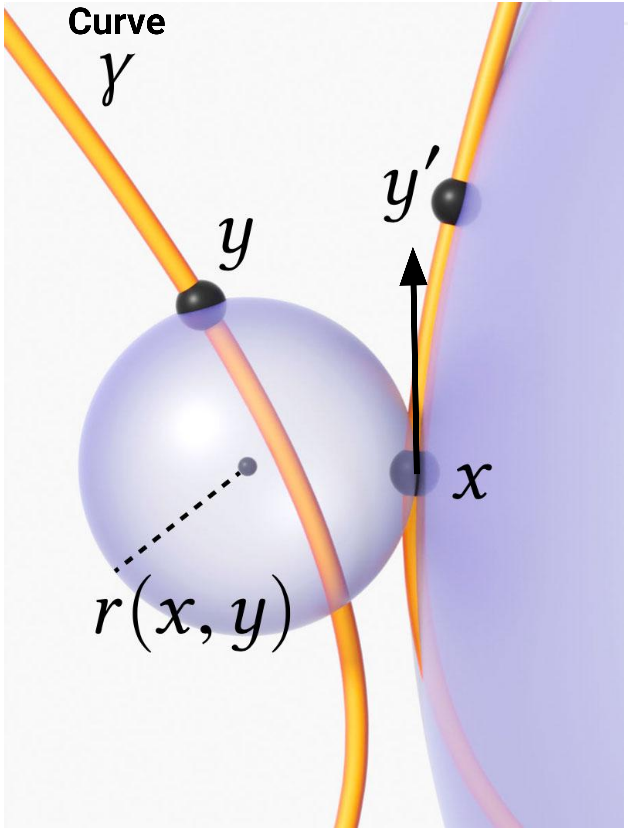
*The inverse of the radius of the smallest sphere passing through point **y** that is tangent to point **x***

$$\epsilon_{Tangent-Point} = \iint \frac{1}{R_t(x,y)^{\alpha}} \qquad R_t(x,y) := \frac{|\gamma(x)-\gamma(y)|^2}{2\,|\gamma'(x)\wedge(\gamma(x)-\gamma(y))|}$$

Step 1: Define a general function for the radius of the sphere

Step 2: Iteratively calculate the sphere radius for each discrete point along the closed curve with respect to every other point

Step 3: Identify the points with the highest energies



The exponent α controls how exaggerated the repulsion is → to be scalar variant, α > 2

Georgia Tech

Gradient Descent Method

Fundamental optimization method used to find local max/minimum

⇒ In this setting, gradient descent employed to minimize tangent-point energy by translating a point with high energy in the direction of decreasing energy

$$p_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$$

Requirements of ∈:

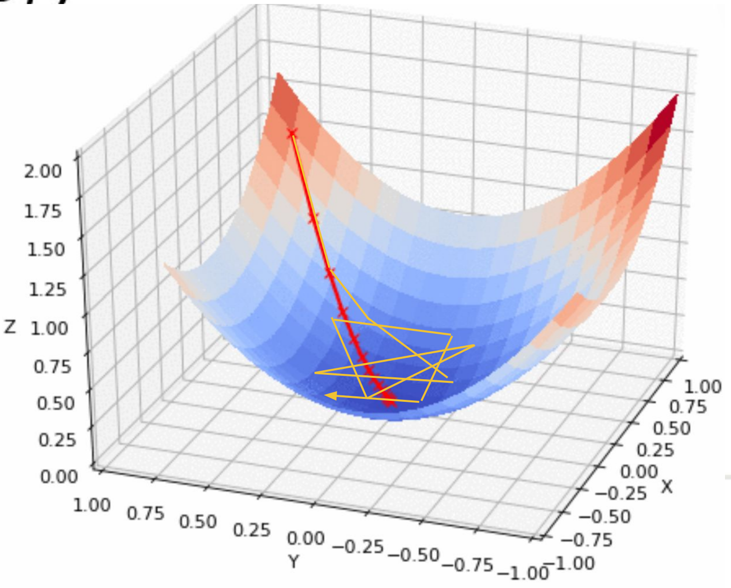
- Differentiable
- concave

$$p_{i+1} = p_i - \eta \cdot \nabla \epsilon(p_i)$$

Step 1: Calculate the gradient at the current point

Step 2: move a step (scaled by η) in the opposite direction of the gradient

Step 3: Repeat until minimum energy criteria is met

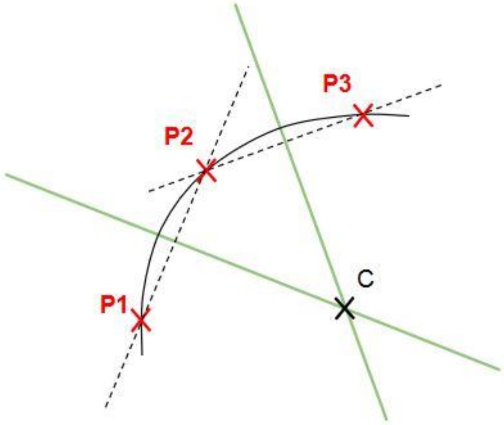


η is the learning rate - it controls the size of the step

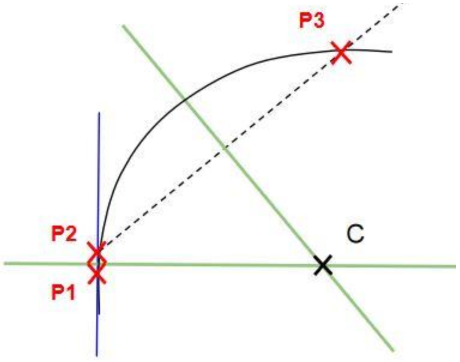
- (red line) = good learning rate
- (yellow line) = learning rate too large

Implementation: Tangent-Point Energy (1/3)

1. How to determine the center C of the smallest sphere connecting 3 points ?



2. Adding the tangency condition



3. Equations of both planes are calculated using their normals : segment P1P2 and P2P3

$$\text{Plane equation : } Ax + By + Cz + D = 0$$
$$\Leftrightarrow n_x \cdot x + n_y \cdot y + n_z \cdot z - (n_x \cdot x_m + n_y \cdot y_m + n_z \cdot z_m) = 0$$

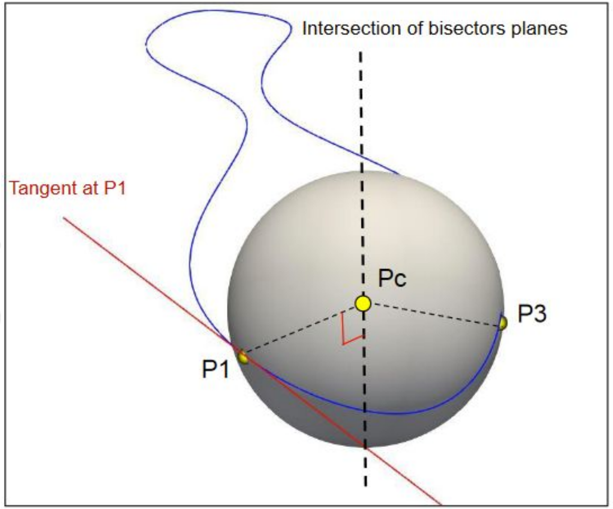
4. Function that determine the intersection of both planes : gives a line equation

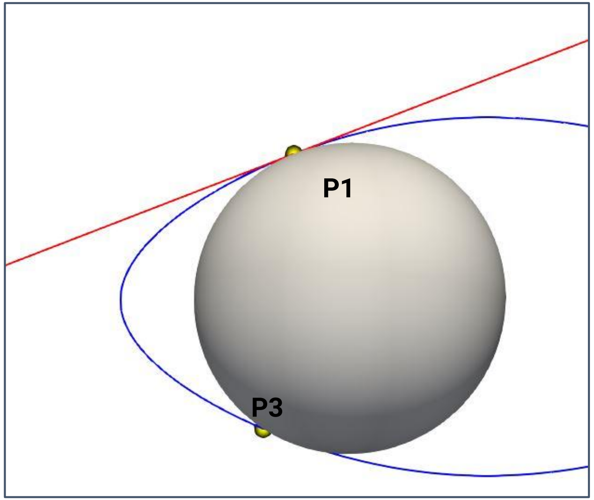
Georgia Tech

Implementation: Tangent-Point Energy (2/3)

5. Finally, the closest point to P1 that is on the intersection line is calculated which corresponds to the center of the sphere

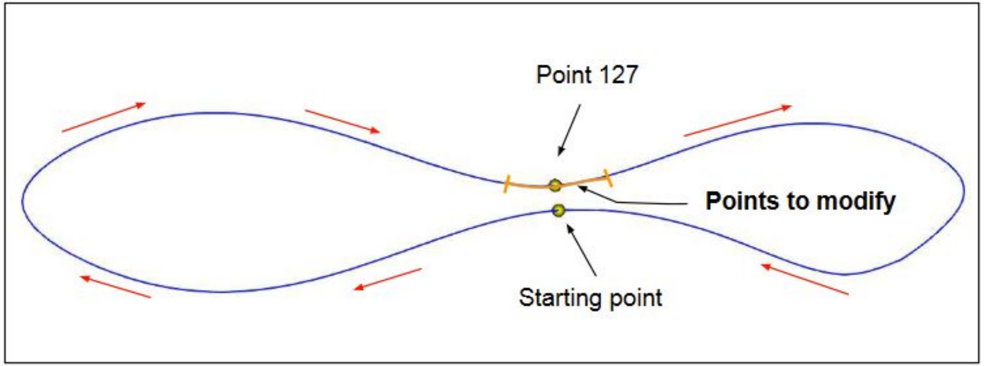
Example (200 points):



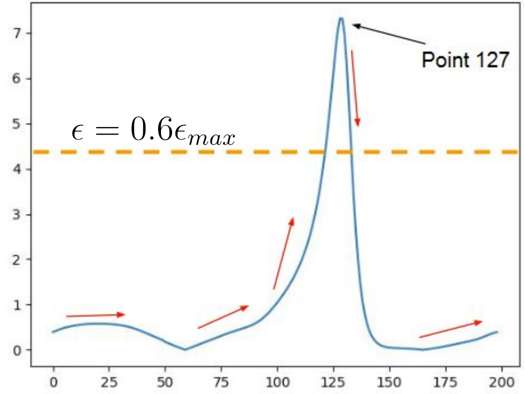


Implementation: Tangent-Point Energy (3/3)

Tangent point energy : $\epsilon = \frac{1}{r^{\alpha}}$



Top view of the spline (200 points)



Tangent-point energy with respect to the starting point

- As expected, really high energy around point 127
- Indicates points that need to be modified

Implementation: Gradient Descent (1/2)

Tangent-point energy : $\epsilon = \dfrac{1}{(\sqrt{(x-x_c)^2+(y-y_c)^2+(z-z_c)^2})^\alpha}$

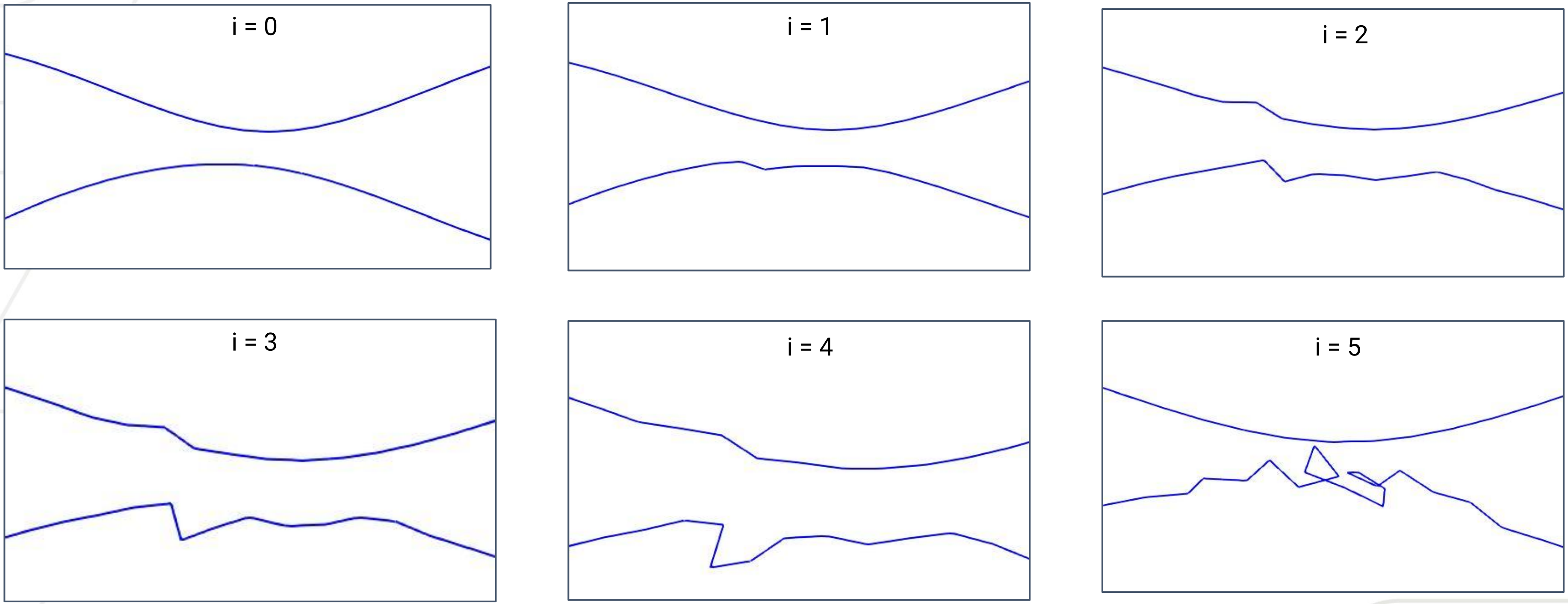
Gradient :

$$\nabla\epsilon = (\frac{-\alpha(x-x_c)}{((x-x_c)^2+(y-y_c)^2+(z-z_c)^2)^{\frac{\alpha}{2}+1}}, \frac{-\alpha(y-y_c)}{((x-x_c)^2+(y-y_c)^2+(z-z_c)^2)^{\frac{\alpha}{2}+1}}, \frac{-\alpha(z-z_c)}{((x-x_c)^2+(y-y_c)^2+(z-z_c)^2)^{\frac{\alpha}{2}+1}},)$$

Modified point : $(x, y, z) = (x, y, z) - \eta \times \nabla\epsilon$

$\eta : learning\ \ rate\ \ (0 < \eta < 1)$

Implementation: Gradient Descent (2/2)



Optimisation of the spline after 5 iterations

To Conclude: Successes, Challenges, & Future Work

Successes

- Use of course material to find tangent-point energy sphere
- Points of highest energy accurately identified
- Curve visualization

Challenges

- Current implementation of gradient descent not converging

  ⇒ Experiment with learning rate η
- Tangent-point energy can exhibit fractional derivatives

Future Work

- Generate 3D model of bicep muscle strand
- Experiment with faster optimization methods
- Expand to surface optimization

Georgia Tech

Questions?

References

https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21

https://en.wikipedia.org/wiki/Gradient_descent

https://arxiv.org/pdf/2104.10238.pdf

https://www.cs.cmu.edu/~kmcrane/Projects/RepulsiveSurfaces/index.html

http://www.cs.cmu.edu/

~kmcrane/Projects/RepulsiveCurves/RepulsiveCurves.pdf

https://iopscience.iop.org/article/10.1088/1367-2630/4/1/320/pdf

http://liberzon.csl.illinois.edu/teaching/cvoc/node4.html

https://www.chalmers.se/SiteCollectionDocuments/Produkt-%20och%20produktionsutveckling/Nationell%20kompetensarena%20kring%20produktoptimering/Methodology_for_Topology_and_Shape_Optimization_report.pdf

Georgia Tech