Summary

Note: In my research, we actually did not use my method and stuck with the original Augmented Lagrangian method, because it generalized to higher dimensions (which was necessary for the research project).

I'm just talking about my solution, because I believe it still demonstrates algorithmic thinking… Moreover, if a simpler solution like this exists for the 3D case, I think it would be an even simpler alternative to the existing 3D Augmented Lagrangian solution.

- Consider a cycle graph of N points in the plane (Graph Laplacian with 2 on the diagonal, –1 on adjacent entries, and –1 on upper right and lower left entries).
- The problem is to **minimize the global perimeter** of this cyclic polygon while **keeping its total signed area CONSTANT** (2D Area Shoelace formula).
-
- Instead of using a general constrained optimization method, I made a **locally optimum geometric algorithm.**
- Each vertex P_i participates in a triangle with its neighbors (P_{i-1},P_i,P_{i+1}). Moving P_i along the line parallel to the base preserves the triangle's area while enabling us to minimize the sum of its two incident edge lengths.
- **Because each triangle is an independent "chunk" of the shape, this also reduces global perimeter while preserving global area.**
-
- Concavity is detected using the sign of the cross product of consecutive edge vectors; **concave triangles are flipped across their base (which preserves global perimeter but increases global area), and a global shrinking similarity transform restores the correct area and further reduces perimeter.**
-
- **Alternating vertices can be updated in parallel,** since the other vertices serve as fixed anchors.
-
- This algorithm is parameter-free, robust to random and even self-intersecting initial configurations, and guarantees monotonic improvement.
- The Augmented Lagrangian method is more complicated, requires parameter tuning, and is sensitive to initialization.
-
- **The method combines local geometric properties, global constraints, and parallelization to solve a constrained optimization problem without relying on heavy optimization machinery.**

Problem Setup

- N points on the 2D X-Y plane are **randomly initialized**.
- Connectivity represented as an **NxN Graph Laplacian matrix**:
  - 2 on diagonals, -1 on neighbors of diagonals, -1 in top-right and bottom-left, 0 elsewhere.

$$L = \begin{bmatrix} 2 & -1 & & & & -1 \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & -1 & \ddots & \ddots & \\ & & & \ddots & 2 & -1 \\ -1 & & & & -1 & 2 \end{bmatrix}$$

  - In other words, every point has two distinct neighbors
    - (Point 1 is connected only to Point N and Point 2)
    - (Point 2 is connected only to Point 1 and Point 3) …
    - (Point N is connected only to Point (N-1) and Point 1)

- Objective: **minimize total perimeter** while **preserving the total area**
  - Total area is calculated with 2D Area Shoelace formula.

$$A = \frac{1}{2} \sum_{i=1}^{N} \left(p_i \times p_{i+1}\right) \qquad L = \sum_{i=1}^{N} \|p_{i+1} - p_i\|$$

One Potential Solution (Augmented Lagrangian)

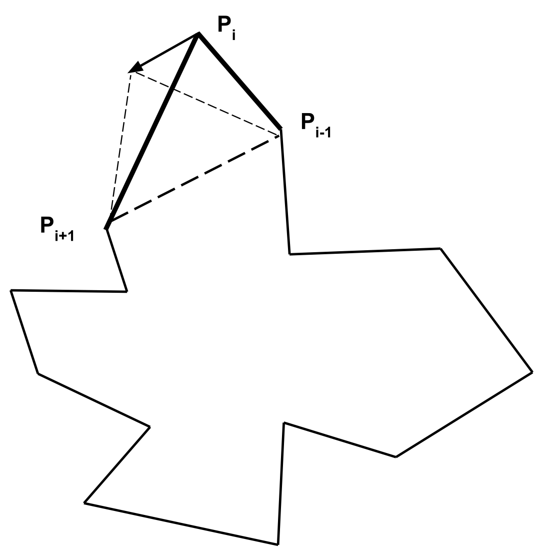
- The problem can be formulated as a continuous constrained optimization problem: minimize perimeter subject to an equality constraint on the area.
- The Augmented Lagrangian method treats the area constraint indirectly using Lagrange multipliers and penalty terms, without relying on any geometric structure.
- It updates all point positions simultaneously

Advantages:
- Fully general; does not require the structure of triangles or polygons.
- Extends to higher dimensions and arbitrary constraints.

Disadvantages:
- Sensitive to initial conditions, especially when the polygon is self-intersecting or poorly conditioned.
- Requires tuning of penalty parameters and step sizes.
- Uses relatively heavy machinery for what is, in this special case, a simple geometric problem.

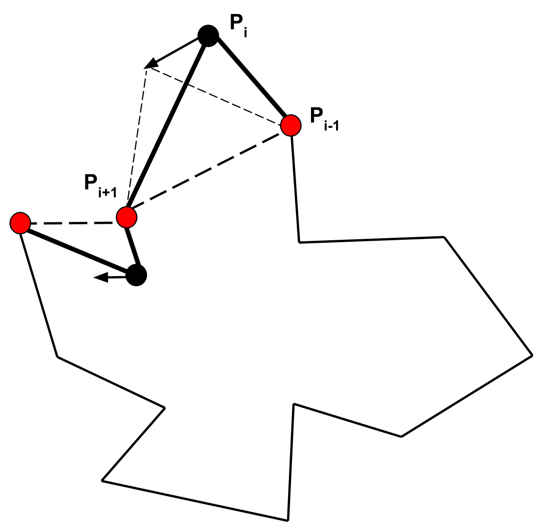
My Simpler Algorithm Details

1. **Local Triangle Decomposition:**
    - For each vertex P_i, define a **local triangle** with neighbors: (P_{i-1}, P_i, P_{i+1}).
    - This local triangle is a **chunk of the overall shape**: moving P_i within this local triangle does not affect the Total Area or Total Perimeter of the remaining portion of the shape.
    - Base of Local Triangle: “Imaginary” line segment connecting P_{i-1} to P_{i+1}
        - This is kept constant (it’s the reference line segment)
    - Sides of Local Triangle: “Actual” line segment connecting P_{i-1} to P_i and P_i to P_{i+1}.
        - These are changed via moving P_i (they are the sides of the local triangle that are also part of the overall graph, and therefore **do affect total perimeter**)

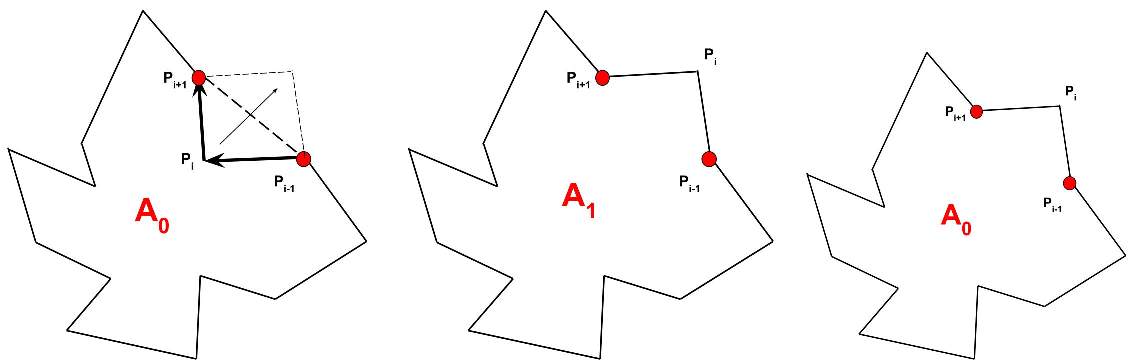


2. **Isosceles Minimization:**
    - Move P_i **parallel to the base of the local triangle**, along the line that preserves the area of the local triangle, because we’re not modifying the “base” or the “height” of the local triangle.
    - Choose the position that minimizes the sum of the two side lengths, which makes the triangle go from scalene to isosceles.
    - Reducing the sum of local side lengths **reduces the total perimeter of the overall shape.**
    - Maintaining the area of the local triangle ensures that the **total area of the overall shape is also the same.**

3. **Parallelization:**
   - **Alternate sets of vertices can be adjusted/moved simultaneously** (**the black MOVING vertices**), which changes the total side lengths of all the local triangles, all at once.
   - This is possible, because the complementary alternate set of vertices (the **red ANCHOR vertices** that form the bases of the local triangles) remain fixed, ensuring that all the local triangles have consistent references that don't change, no matter how we adjust the MOVING vertices.
   - For example, if all the odd vertices are chosen as MOVING vertices, then their movements are completely independent of one another, because their corresponding ANCHOR vertices (the adjacent even vertices they're moving in relation to, don't change)



4. **Flip Concave Triangles to Convex, and do Similarity Transform (if necessary):**
   - **Going counterclockwise**, we compute the cross product between the vectors representing the two consecutive line segments (P_{i-1} to P_i) and (P_i to P_{i+1}).
   - Negative cross product ⇒ concave triangle. Positive ⇒ convex.
   - If the local triangle is concave, **flip P_i across the base of the local triangle** to make it convex;
     - Flipping preserves side lengths and area of local triangle
     - Which means the **total area increases and the total perimeter stays constant**.
   - If there are at least 1 flips, a **similarity rescaling** must be applied.
     - Calculate the new area A_1
     - Shrink the entire shape by a factor of sqrt(A_0/A_1)
     - So now, the **total area is constant at A_0 again and the total perimeter is reduced**



Algorithmic Properties

- **Provable global improvement from adjusting local properties:**
  - Moving P_i along area-preserving line strictly minimizes local edge lengths (and thus total perimeter).
  - Flipping P_i across base of local triangle to go from concave to convex, then applying global shrinking similarity transformation, also strictly minimizes total perimeter while preserving total area
- **Geometric robustness:** Handles random, self-intersecting initial configurations.
- **Parallelizable** due to the fact that alternating points don't change, which means the movements of the other alternating points are independent of one another (since the moving points are only dependent on their adjacent points, which are fixed).

Why It's Better than Augmented Lagrangian for This Problem

- It uses the structure of the problem: each vertex forms a well-defined local triangle whose contribution to the global area and global perimeter can be manipulated independently
- Area and perimeter changes are **exact and local**, whereas Augmented Lagrangian enforces constraints only approximately.
- The algorithm is **robust to arbitrary random initializations**, including self-intersecting polygons. AL, by contrast, can become unstable under such conditions.
- Moving P_i along the area-preserving line that minimizes its two incident edge lengths produces an **exact, closed-form local improvement:** no gradients, step-size tuning, or penalty updates.
- Concave-to-convex flips **guarantee monotonic improvement**: flipping does not change local edge lengths but increases area, and the subsequent uniform similarity transform reduces the global perimeter exactly.

- Parallelization is straightforward because alternating vertices act as **anchors**, making all local triangle operations independent.
- Overall, the method achieves **the same goal without no parameter/step-size tuning and no complex continuous optimization machinery**, making it substantially simpler and more predictable for this specific problem.