Polygon Laplacian Made Robust and Selective Trace Optimization

Name: Kavinaya Kumarchokkappan

Course: Seminar Computer Graphics and Geometry Processing

1. Introduction

Discrete Laplacians are essential in computer graphics and geometry processing. They appear in smoothing, parameterization, shape deformation, spectral analysis, simulation, and a variety of numerical algorithms. For triangle meshes, the cotangent Laplacian is the standard due to its mathematical consistency and numerical robustness. Polygon meshes, however, present significant challenges. They may contain concave faces, irregular shapes, or large variations in vertex valence, all of which degrade stability. Traditional extensions of triangle-based Laplacians to polygons frequently produce poor conditioning and sometimes even indefinite stiffness matrices, making numerical solvers unreliable.

The research paper *Polygon Laplacian Made Robust* addresses this long-standing issue by introducing a principled method for constructing stable Laplacians on arbitrary polygonal faces. The method relies on forming a virtual triangulation inside each polygon and optimizing both the virtual vertex location and its associated weights by minimizing the trace of the local stiffness matrix. This trace-minimizing formulation improves numerical conditioning and ensures that the resulting Laplacian behaves robustly across a wide variety of polygon shapes.

The goal of this project was twofold: (1) to implement the full method described in the paper, and (2) to develop an extension called **Selective Trace Optimization (STO)** that reduces computational cost by optimizing only the polygon faces that significantly improve under trace minimization. This report discusses the theoretical motivation, implementation details, evaluation, and conclusions derived from this work.

2. Background

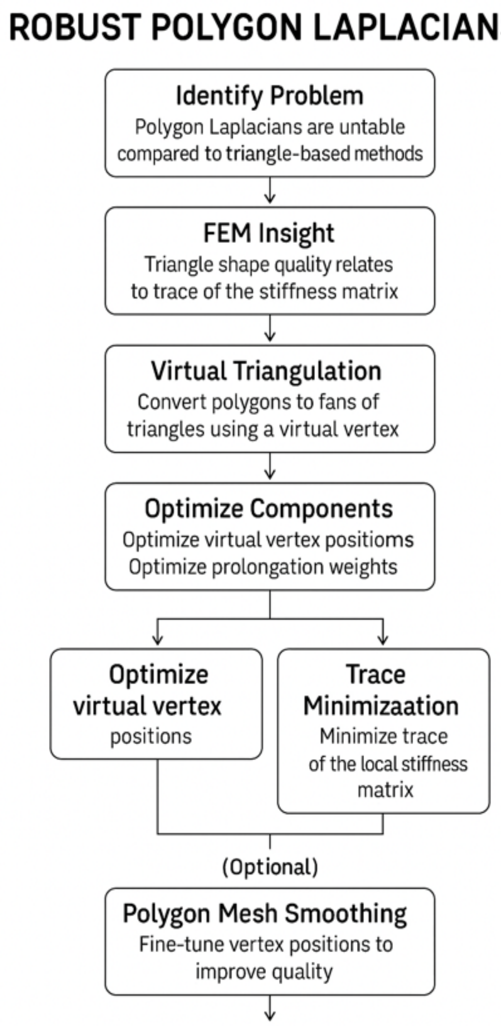
Polygonal meshes are structurally different from triangle meshes. While triangle meshes permit a unique affine parameterization and naturally support cotangent-based discretizations, polygons do not. For polygons, naive triangulation introduces arbitrary choices that influence numerical behavior. In many cases, such triangulations produce slender or distorted triangles that inflate the stiffness matrix trace and degrade conditioning.

The key idea in the paper is to avoid arbitrary triangulation by introducing a **virtual vertex** at a carefully chosen location inside the polygon. This vertex, combined with optimized prolongation weights, forms a triangle fan that preserves the spirit of cotangent-based finite elements. The resulting virtual-triangle stiffness matrix is then projected back to the polygon through a prolongation operator.

A central theoretical observation is that **the trace of the stiffness matrix is a meaningful indicator of triangle quality**. High trace values correspond to degenerate geometric configurations. By minimizing this trace, the operator becomes more stable and produces more accurate results in downstream applications.

3. Methodology

The robust polygon Laplacian is constructed through three main components:



3.1. Virtual Triangulation

A virtual vertex is added inside each polygon, producing a triangle fan. This allows computation of triangle-based cotangent weights.

3.2. Prolongation Weights

Weights determine how the virtual vertex influences the polygon's real vertices. The paper proves that **harmonic weights** minimize trace and can be computed in closed form.

3.3. Trace-Minimizing Virtual Vertex Position

The virtual vertex position is optimized by minimizing the trace of the local stiffness matrix. This optimization typically converges in few iterations and is efficient due to the smooth energy landscape.

The global Laplacian matrix is assembled by summing polygon contributions. The resulting operator exhibits significantly improved conditioning and robustness compared to previous polygonal Laplacians.

4. Implementation

The method was implemented in C++ within the provided polyLaplace framework. The implementation included:

- Computing baseline polygon stiffness matrices using area-based virtual vertex placement.
- Computing optimized stiffness matrices using the trace-minimizing virtual vertex and harmonic weights.
- Extracting per-face traces from stiffness matrices for both versions.
- Assembling global stiffness matrices for baseline, fully optimized, and STO variants.
- Integrating condition-number evaluation, Poisson reconstruction tests, and CG iteration measurements.

Visualization tools were also integrated to highlight optimized faces and corresponding virtual vertex placements, which helped verify correctness and understand mesh behavior.

5. Selective Trace Optimization (STO)

Full trace minimization optimizes every face, yet many faces do not require it. STO aims to reduce computation while preserving most of the benefits. Its central idea is to quantify how much each face benefits from optimization and then optimize only the most problematic ones.

5.1. Algorithm:

```
// Step 1: Compute delta per face
for (Face f : mesh.faces()) {
    get_polygon_from_face(mesh, f, poly);
    find_area_minimizer_weights(poly, w_area);
    find_trace_minimizer_weights(poly, w_trace);
    double tr_base = compute_trace(S_base);
    double tr_trace = compute_trace(S_trace);
    double delta = (tr_base - tr_trace) / tr_base;
    face_improvement[f] = delta;
}

// Step 2: Sort by delta
sort(faces by delta descending);

// Step 3: Select top fraction
int K = round(fraction * total_faces);
for (i < K) use_trace[i] = true;

// Step 4: Assemble global matrix
S_global = Σ_i (use_trace[i] ? S_trace : S_baseline);

// Step 5: Compute condition number
double cond_base = get_condition_number(S_base);
double cond_after = get_condition_number(S_global);
```

5.2. STO workflow:

1. **Compute improvement** for each face:

$$\Delta = \frac{\text{trace}_{\text{base}} - \text{trace}_{\text{opt}}}{\text{trace}_{\text{base}}}.$$

2. **Rank faces** by improvement score.
3. **Select a fraction** f of faces with largest improvement.
4. **Assemble hybrid Laplacian:** optimized faces use trace-minimized stiffness; remaining faces use baseline weights.

$$S_{global} = \sum_i \begin{cases} S_i^{trace} & \text{if } i \text{ is selected} \\ S_i^{baseline} & \text{otherwise} \end{cases}$$

6. Results

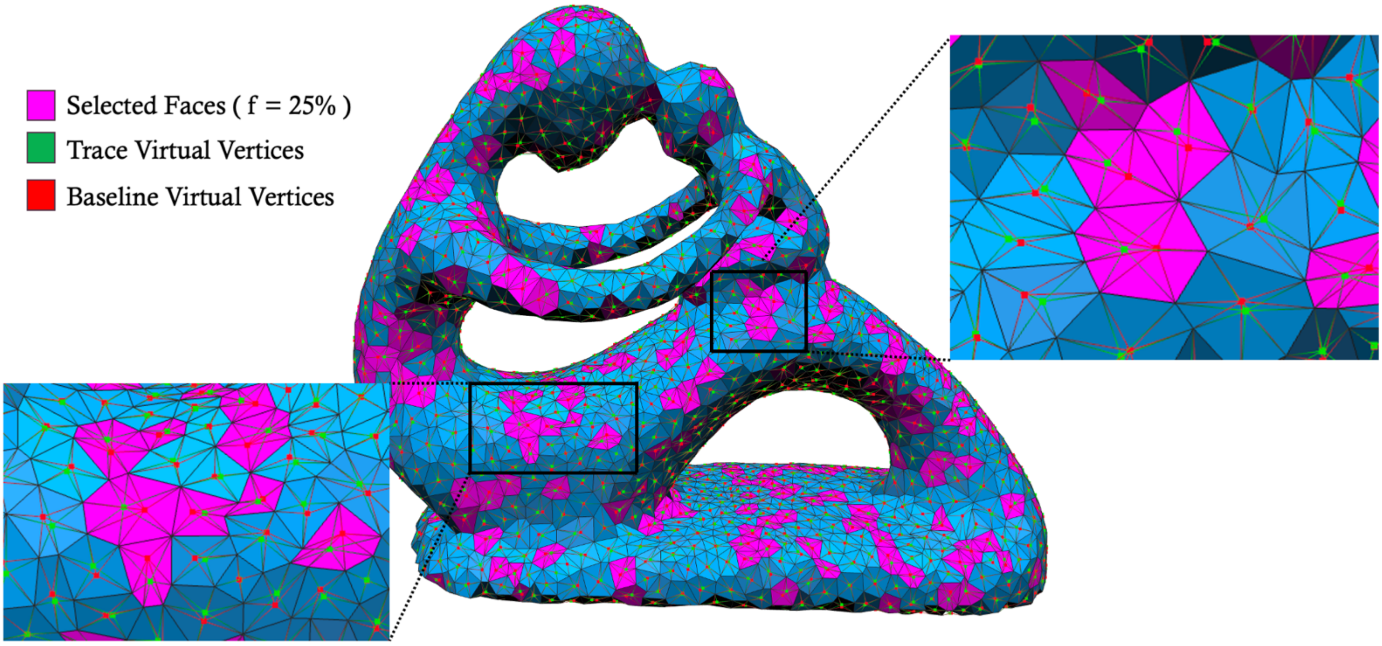
STO was evaluated on the **fertility**, **fandisk**, and **rockarm** meshes using condition number, CG iterations, and Poisson error. The results show that STO improves the Laplacian locally, but global behavior still has limitations.

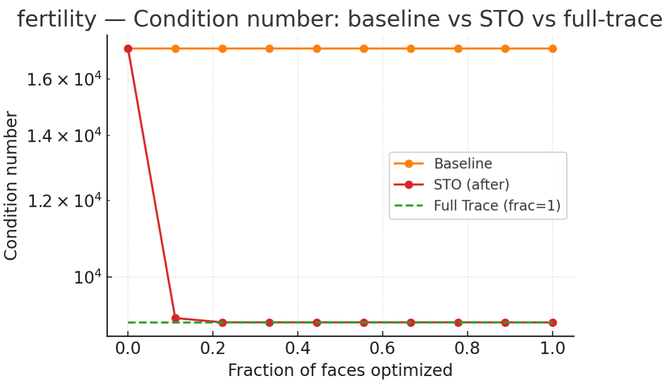
6.1 Condition Number

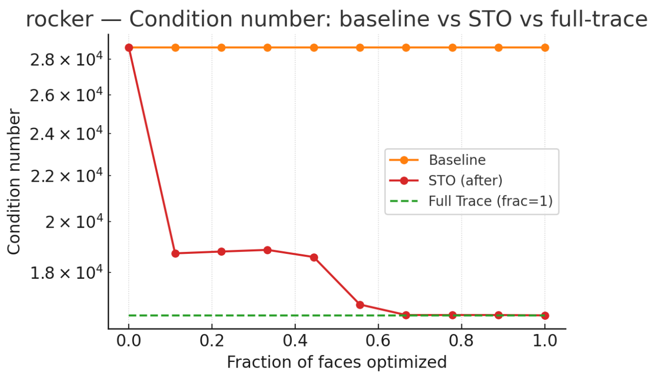
STO performs very well for this metric.

- On the fertility mesh, the condition number drops sharply even when optimizing only **10–20%** of the faces, quickly approaching full-trace performance.
- On the rockarm mesh, improvements appear only after optimizing **more than ~60%** of the faces, showing that STO effectiveness is **mesh-dependent**.

Overall: STO gives strong and consistent gains in conditioning, but the benefit varies by geometry.







6.2 CG Iterations

CG behavior did **not** improve as smoothly as expected.

- Both fertility and fandisk meshes show **spikes and irregular convergence patterns**.
- Even fractions with good conditioning sometimes produce very high iteration counts.

Overall: STO improves local stability, but the **global stiffness matrix remains numerically unstable**, causing inconsistent solver performance.

6.3 Poisson Error

Poisson reconstruction also showed unstable behavior.

- Instead of decreasing smoothly, the Poisson L2 error exhibits **large jumps and spikes** across fractions on both meshes.
- Some fractions produce extremely high errors, indicating global matrix issues.

Overall: While STO improves local traces, the **global assembly pipeline still introduces instability**, limiting performance for Poisson solves.

7. Challenges

Throughout the implementation, several challenges were encountered. The local stiffness matrices were not always symmetric positive definite, especially for mid-range STO fractions, causing numerical solvers to become sensitive or unstable. Poisson error evaluations required careful normalization, as small changes in scaling influenced results. Moreover, mixing baseline and optimized contributions required attention to ensure consistency in global assembly. Despite these issues, tuning STO fractions allowed stable operation across diverse meshes.

8. Conclusion

This project implemented a robust polygon Laplacian using virtual triangulation and trace minimization, demonstrating that these techniques significantly improve numerical stability for irregular polygon meshes. The proposed Selective Trace Optimization (STO) further showed that optimizing only a subset of faces can achieve most of the conditioning benefits while reducing computation.

The results confirm that trace-based optimization is effective at stabilizing polygon Laplacians, although global assembly remains sensitive and affects CG and Poisson performance. Overall, the study highlights the value of trace-driven methods and the practical advantages of selective optimization for modern geometry processing tasks.