Embedding Convex Polyhedral Metrics Using the Adiabatic Isometric Mapping (AIM) Algorithm

Shannon Ray
Advisor: Dr. Warner Miller
Florida Atlantic university
Physics Department

Abstract

The Alexandrov embedding theorem states that any simplicial complex homeomorphic to a sphere with strictly non-negative Gaussian curvature at each vertex can be isometrically embedded in $\mathbb{R}^3$ as a convex polyhedron. Due to the nonconstructive nature of his proof, there have yet to be any algorithms that realize the Alexandrov embedding in polynomial time. Following his proof, we produced the adiabatic isometric mapping (AIM) algorithm. The AIM algorithm is approximately quadratic with time and reproduces edge lengths up to arbitrary accuracy.

Introduction

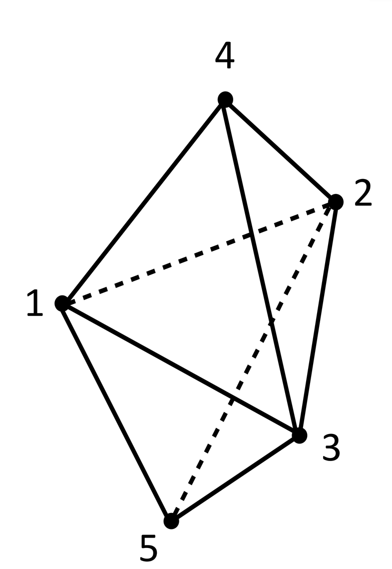
The problem of embedding surfaces homeomorphic to a sphere with a metric of positive Gaussian curvature into $\mathbb{R}^3$ was posed by Herman Weyl in 1916. The first attempt to prove the existence of an embedding was given by Weyl himself. Though he was unable to complete his proof, he did make progress in outlining a solution. Following Weyl's approach, a proof was given by H. Lewy in 1938 though his solution required the components of the metric to be infinitely differentiable. Requirements on differentiability of the metric were reduced to continuous fourth derivatives by L. Nirenberg in 1953. They were further reduced to continuous third derivatives by E. Heinz in 1962. All of these solutions relied on the components of the metric in the continuum. In 1941, Alexandrov gave an approach in the discrete that relied on proving the existence of a convex polyhedron given any convex polyhedral metric. He then showed that in the limit as the number of vertices in the polyhedral metric goes to infinity, one recovers the metric in the continuum thus solving the problem without restrictions on differentiability of the metric.

This poster presents the adiabatic isometric mapping (AIM) algorithm which is a numerical realization of Alexandrov's proof for embedding convex polyhedral metrics. Our algorithm produces approximately convex polyhedrons which are adequate solutions to Alexandrov's embedding problem. A solution is considered approximately convex if it does not have inverted vertices. Vertices are inverted if all the edges emanating from it have negative extrinsic curvature. The integrated mean curvature of these approximately convex solutions agree below a tenth of a percent. This means each solution has the same global extrinsic structure.

Formulation of Problem

Polyhedral metrics are simplicial complexes homeomorphic to a sphere with N_0 vertices, $N_1=3N_0-6$ edges and $N_2=2N_0-4$ triangles. Their data structure is given as lists of triangles, edges and edge lengths. For example, an arbitrary metric with $N_0=5$ vertices is given as,

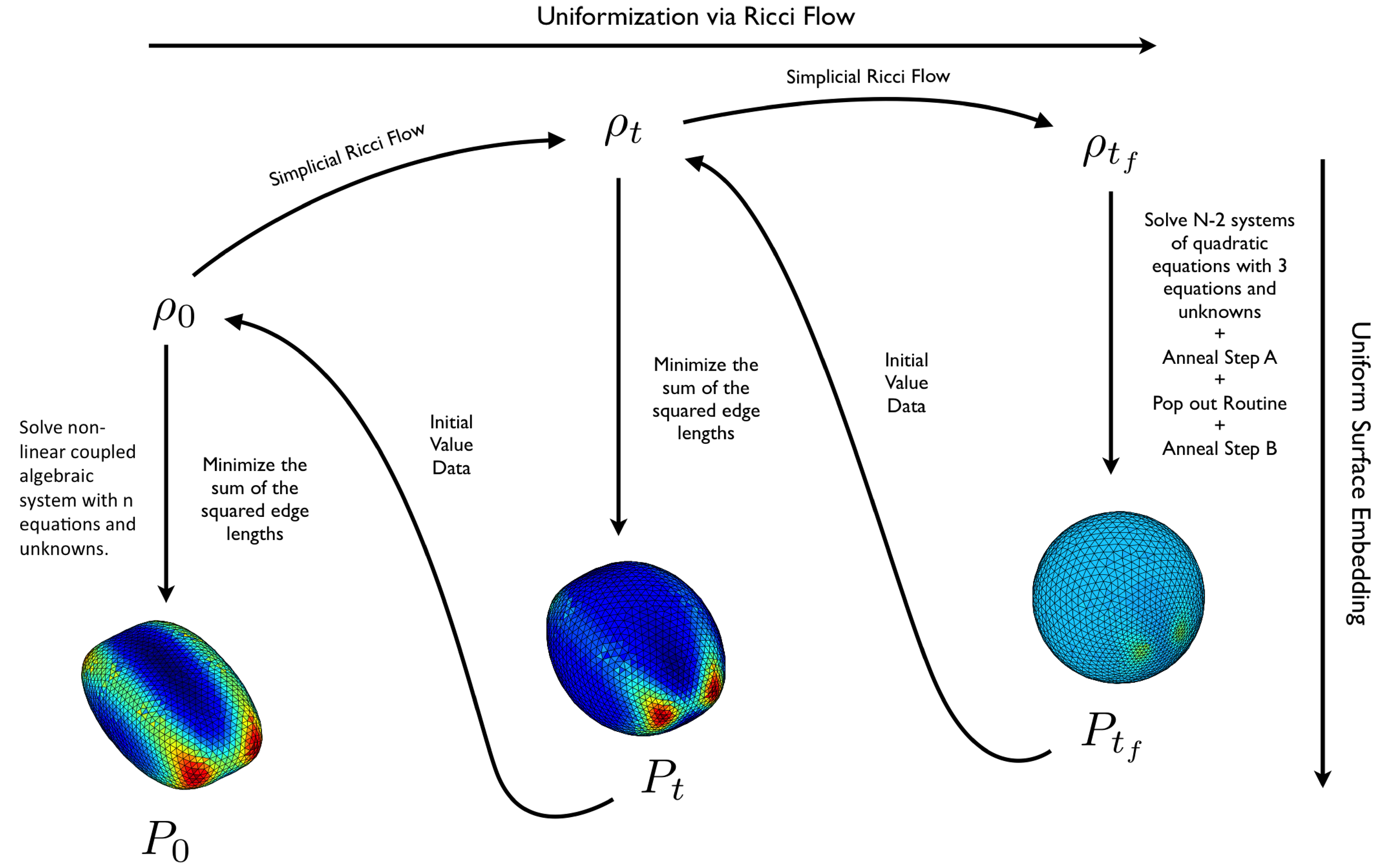
Polyhedral Metric ρ_0



Triangles	Edges	Edge Lenghs
1 2 4	1 2	L_{12}
2 1 5	1 3	L_{13}
5 1 3	1 4	L_{14}
3 1 4	1 5	L_{15}
4 3 2	2 3	L_{23}
3 2 5	2 4	L_{24}
	2 5	L_{25}
	3 4	L_{34}
	3 5	L_{35}

The goal is to produce coordinates for each vertex such that the Euclidean distance $L^2_{ab} = (x_a - x_b)^2 + (y_a - y_b)^2 + (z_a-z_b)^2$ between vertices is equal to the edge lengths given by the original metric ρ_0. Six of the $3N_0$ coordinates can be freely specified in order to mod out translational and rotational degrees of freedom. This is typically done by taking initial triangle Δ_{abc} and placing vertex a at the origin, b on the x-axis and c on the x-y plane. Once the initial triangle is fixed, we have a quadratic system of equations with $3N_0 - 6$ equations and unknowns. Due to the non-linear nature of the equations, there is a "sea of solutions" that satisfy the conditions on edge lengths. The AIM algorithm's purpose is to simplify this system of equations to obtain suitable solutions that are approximately convex.

Adiabatic Isometric Mapping Algorithm



1. Uniformization via Ricci Flow

- At $t=0$ conformal factors are given at each vertex of ρ_0 as $\{u_a(t=0)=0\}$ where a indexes the vertices.
- The Ricci flow equation for each conformal factor is given as $\frac{du_a}{dt} = -\left(k_a - \bar{k}\right)$ where k_a is the Gaussian curvature at each vertex and $\bar{k}$ is the average Gaussian curvature of the surface.
- The conformal relationship between edge lengths is given as $L_{ab}(t) = L_{ab}(0)e^{\frac{u_a(t)+u_b(t)}{2}}$.
- For a sufficiently long time, the Gaussian curvature at each vertex satisfies $||k_a(t_f) - \bar{k}||_2 < \epsilon$ where ϵ is some tolerance.

2. Uniform Surface Embedding

- We assume that P_{t_f} lies on a sphere centered at the origin with radius R. This gives us the constraint $R^2 = x_a^2 + y_a^2 + z_a^2$ for all vertices in ρ_{t_f}.
- We fix an arbitrary triangle Δ_{abc} on the sphere by placing vertex a on the x-axis at a distance R from the origin. Vertex b is placed on the equator in the x-y plane such that the Euclidean distance is equal to L_{ab} from ρ_{t_f}. The coordinates for vertex c are found by solving a system of equations given by the two constraints on Euclidean distance and one radial constraint.
- This System of equations is solved for each subsequent vertex in ρ_{t_f}.
- The uniform surface is found by minimizing $\sum_{ab}(L^2_{ab}(t_f) - [x_a(t_f) - x_b(t_f)]^2 - [y_a(t_f) - y_b(t_f)]^2 - [z_a(t_f) - z_b(t_f)]^2)^2$ using Newton's method with the coordinates on the sphere used as the initial guess. This is called the annealingA procedure. The "pop out" and annealingB routines are used to make P_{t_f} convex.

3. Backward Adiabatic March (BAM)

- We march back from P_{t_f} to P_0 by finding coordinates of for P_j for all steps $t \epsilon [t_0, t_f]$. Coordinates are found by minimizing $\sum_{ab}(L^2_{ab}(t) - [x_a(t) - x_b(t)]^2 - [y_a(t) - y_b(t)]^2 - [z_a(t) - z_b(t)]^2)^2$ using Newton's method.
- If j=2,3 we use the coordinates at time t_{j-1} as the initial guess for Newton's method. If j>3 we extrapolate the coordinates at t_j using the previous three coordinate sets. These extrapolated coordinates are now used as the initial guess for Newton's method.
- Time steps from t_j to t_{j+1} are chosen so that one remains near the global minimum of a approximately convex solution. This restriction on step size determines the adiabaticiy of BAM.

Results

Using the distmesh software, we produced known embedding in $\mathbb{R}^3$. We then remove the coordinates and use the triangulation and edge length as initial data for AIM. The following results compare extrinsic and intrinsic quantities between distmesh and AIM.

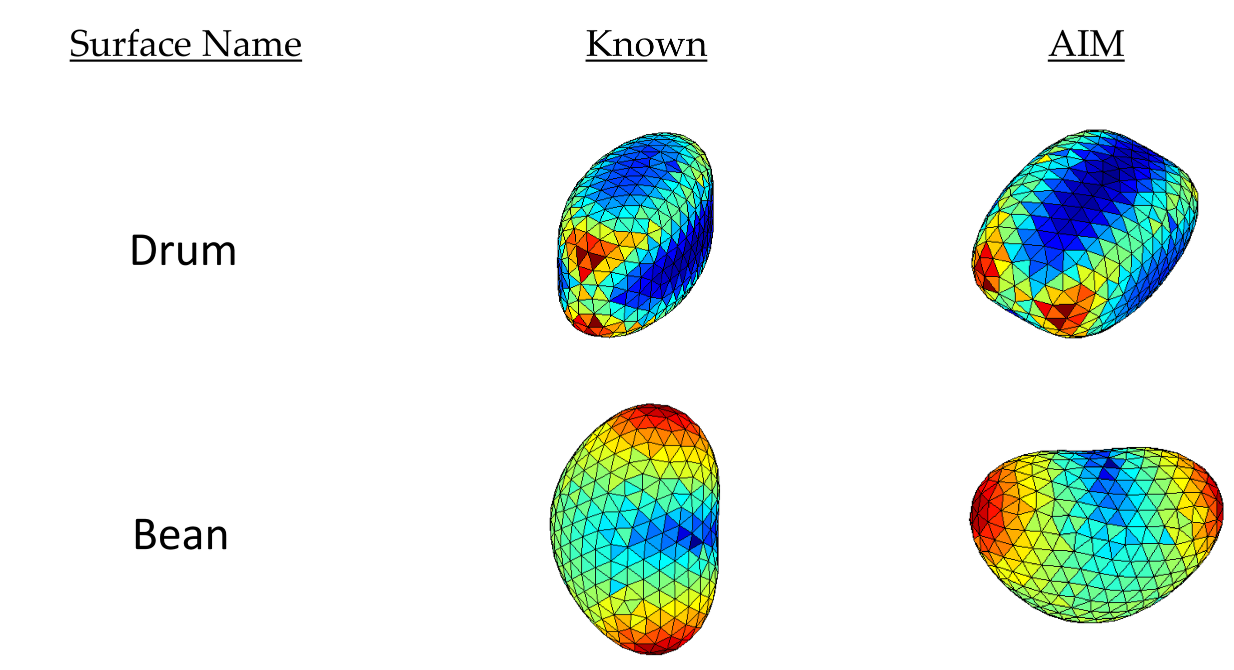


Table 1: Difference between distmesh and AIM

Surface	Max. length	Avg. length	Min. length	$\int_{\Omega}\bar{k}$
Bean (390)	2.5×10^{-6}	1.2×10^{-7}	3.2×10^{-10}	1.7×10^{-6}
Drum (414)	1.2×10^{-6}	8.5×10^{-8}	3.1×10^{-10}	1.7×10^{-6}

We made log-log plots of time vs. number of vertices for each routine and the AIM algorithm as a whole. The slopes from these graphs tell us how run time scales with resolution.

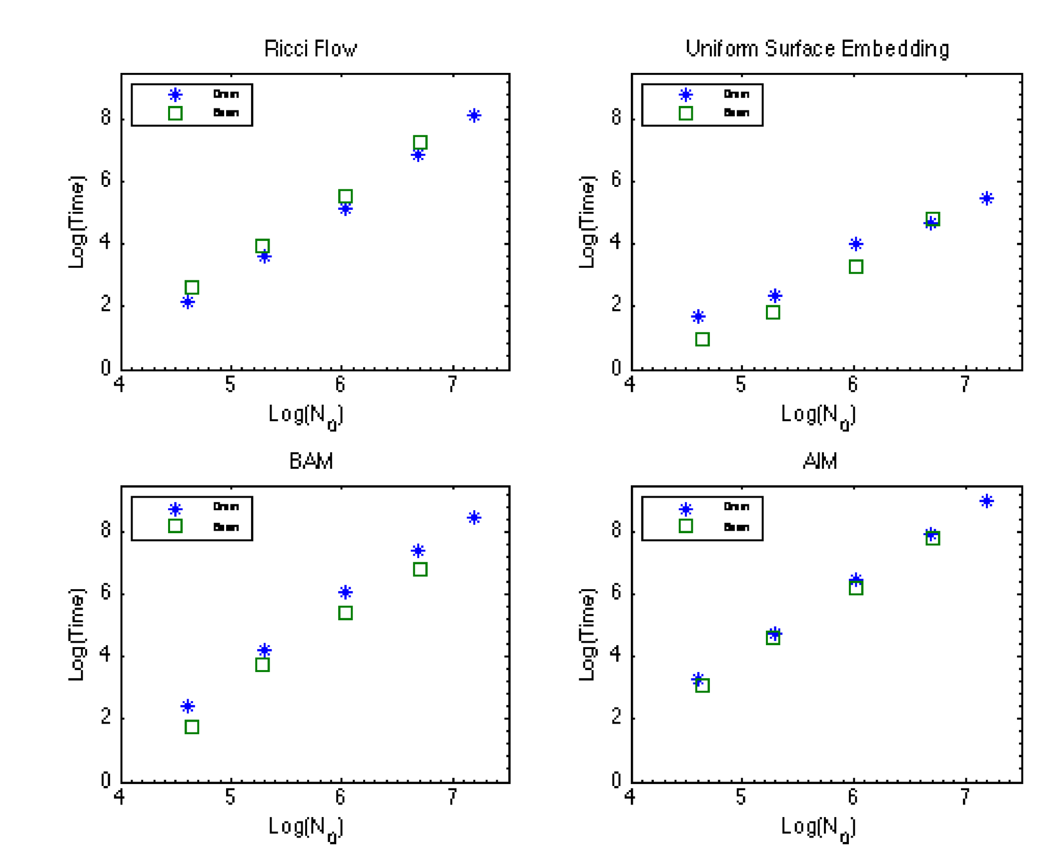


Table 2: Slopes of Log-Log plots

Surface	Ricci flow	Uniform Surface Embedding	BAM	AIM
Bean	2.24	1.89	2.42	2.27
Drum	2.29	1.52	2.33	2.24

Conclusion

An embedding is considered successful if the Euclidean distance between vertices agrees with the edge lengths of the original metric ρ_0 and the surface is approximately convex. Table 1 shows that the maximum discrepancy between edge lengths is on the order of 10^{-6} for both surfaces. We also see in table 1 column four that the integrated mean curvature also agrees up to 10^{-6} which only occurs if the results is approximately convex. Tests have shown that these results are consistent despite resolution. In table 2 wee see that the AIM algorithm scales as about 2.25 for each object. These results are preliminary and must be extended for greater resolutions.