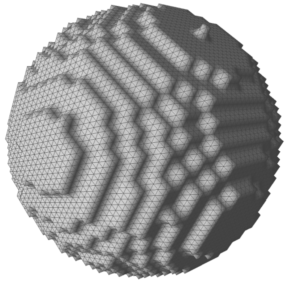
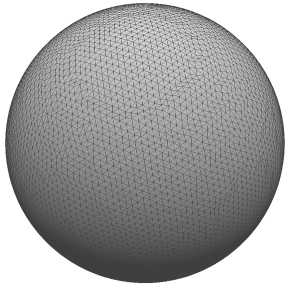
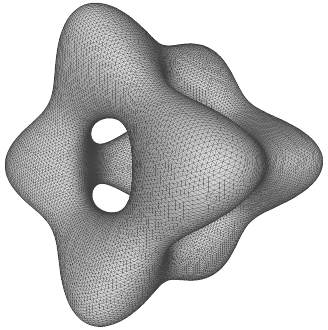
Approximating Isosurfaces by guaranteed-quality triangular meshes

Joel Hass, University of California, Davis
Maria Trnkova, University of California, Davis

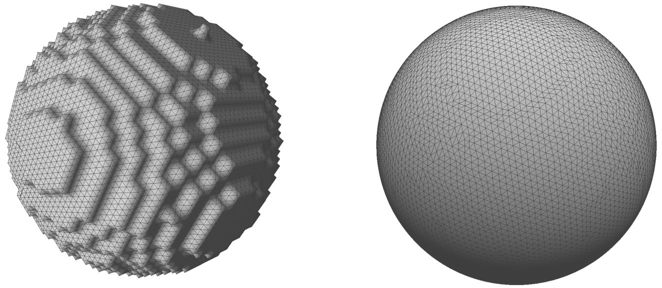


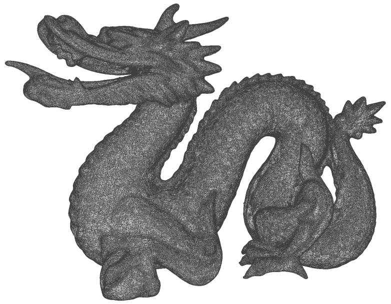




Finding Optimal Meshes

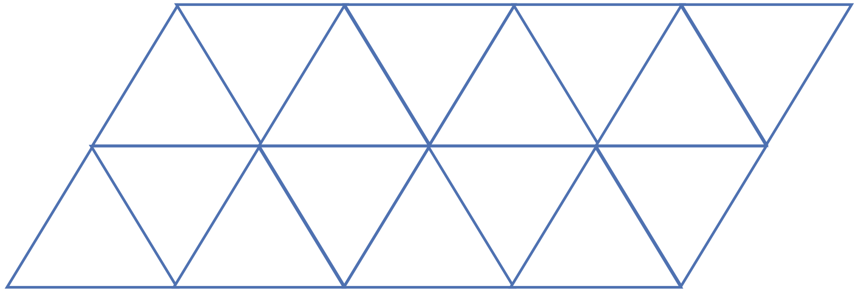
A *mesh* is a 2-dimensional surface made from flat triangles.



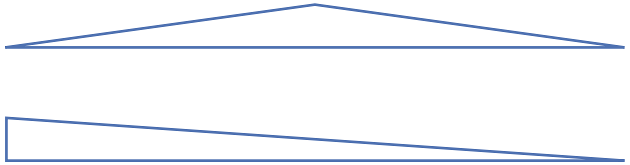


Goal: Approximate a surface in $\mathbf{R}^3$ by a mesh whose triangles are as close to equilateral as possible.

Good



Bad



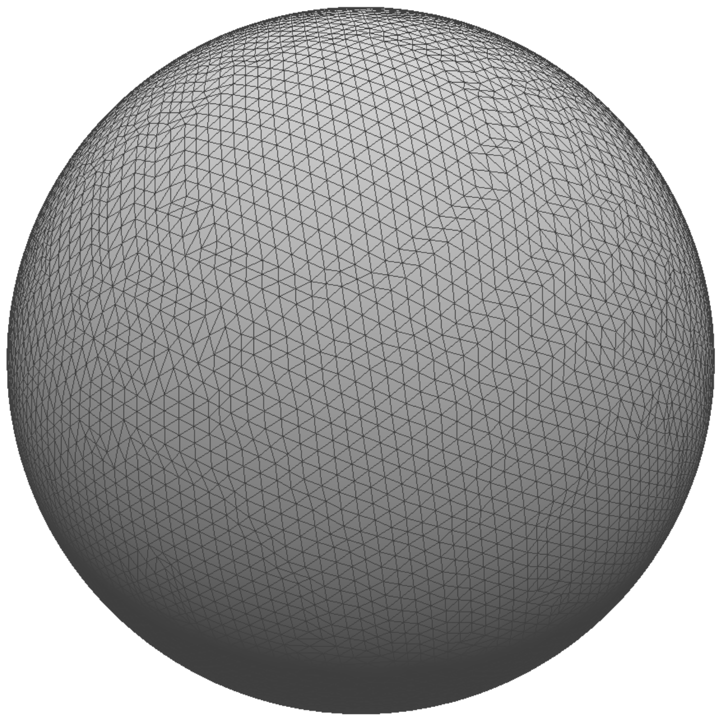
A *sliver* is a triangle with one or more small angles. Slivers cause problems in applications and are to be avoided.

The GradNormal algorithm for smooth function inputs

Start with a surface defined implicitly by an equation $f(x,y,z) = 0$, where f is a differentiable function with regular value 0.

$$F = f^{-1}(0)$$

Decide on how fine a mesh is desired.
Output a mesh approximating F.



Example
Input: A function $f(x,y,z) = x^2 + y^2 + z^2 - 4$.

Output: A high-quality mesh approximating the sphere of radius 2.

Theorem. *When F is smooth, then at a sufficiently fine resolution the GradNormal Mesh has all angles in the interval* **[35.2º, 101.5º].**

The GradNormal Algorithm in 3D

We establish the following bounds.

Theorem. *When F is smooth, then at a sufficiently fine resolution the GradNormal Mesh has all angles in the interval* **[35.2^o, 101.5^o].**

Compare to:

P. Chew (1993): Delaunay with point insertion gives **[30^o, 120^o].**

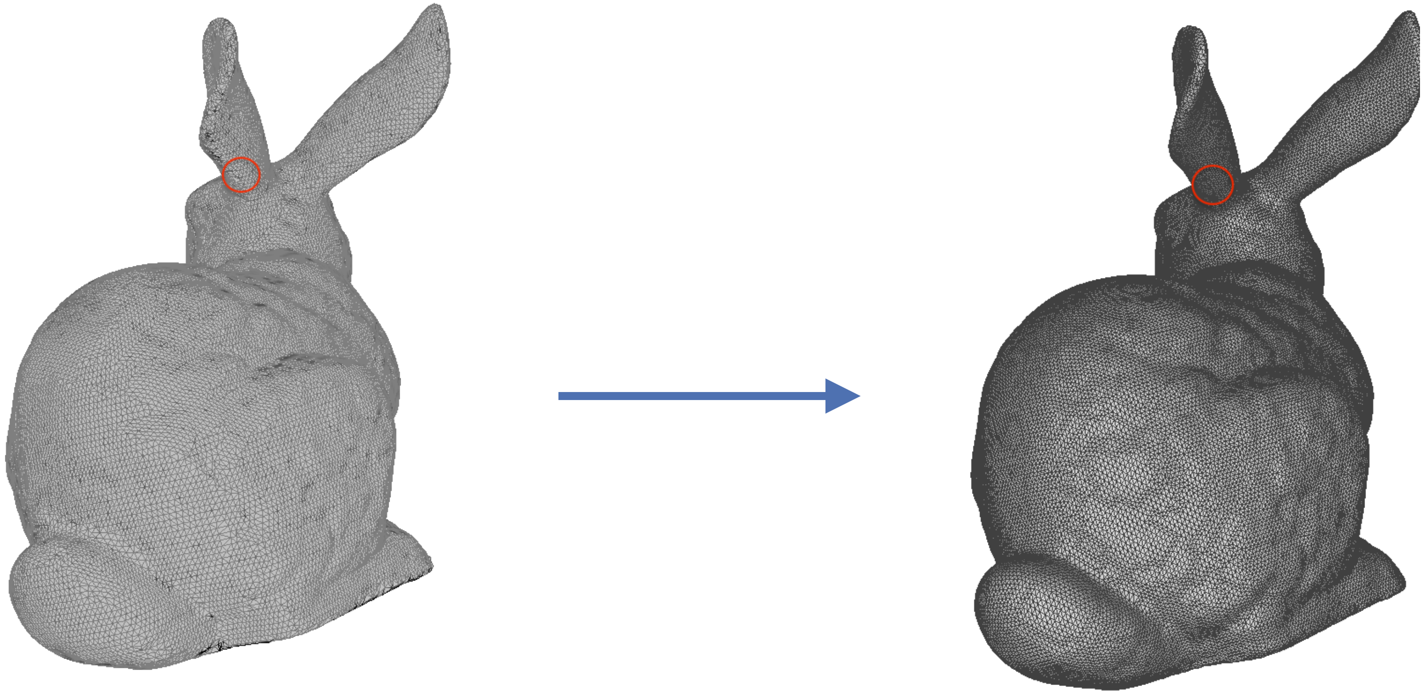
Labelle & Shewchuck (2007): The Isosurface Stuffing algorithm triangulates the surface interior with high quality tetrahedra and also establishes boundary-mesh bounds of **[16^o, 145^o].**

de Verdiere & Marin (1990): There exist meshes realizing ≈ **[51.6^o, 72^o]** on any surface (best possible by Gauss-Bonnet).

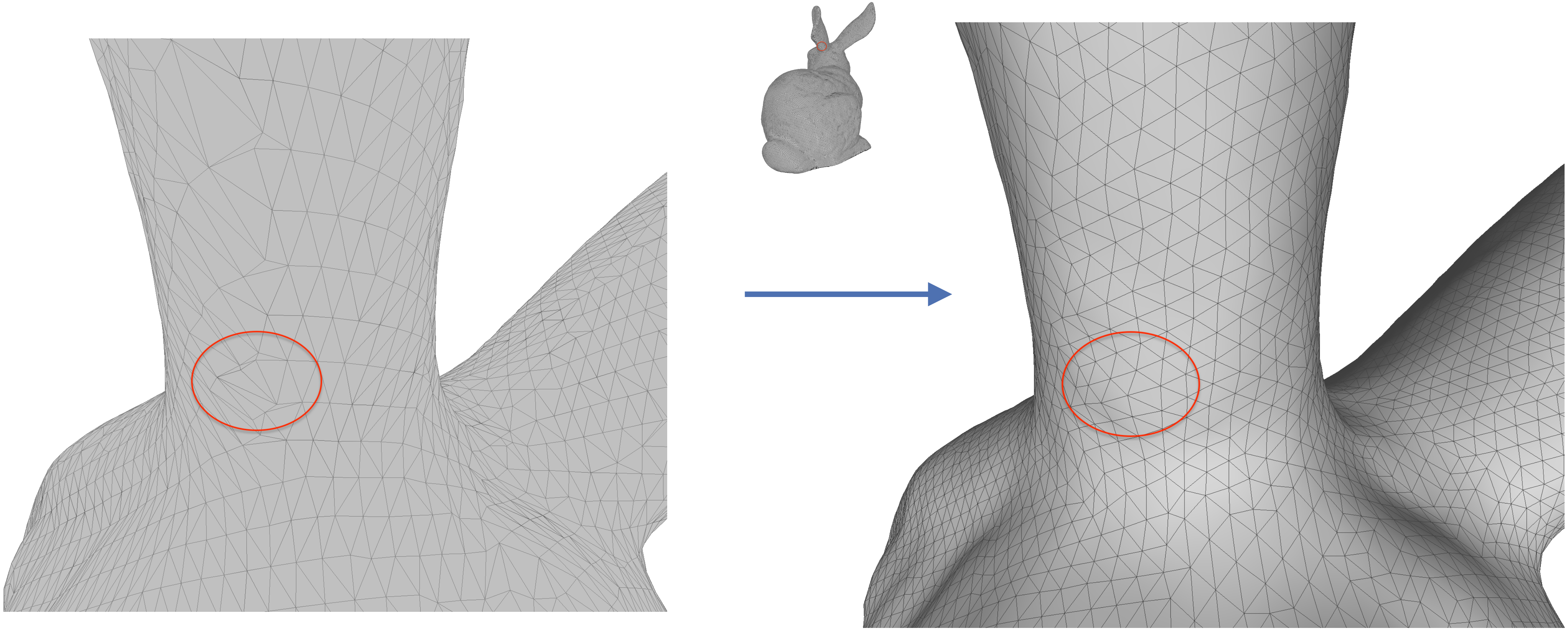
The GradNormal Algorithm for Mesh inputs

Alternately, start with input given by a surface defined by any mesh approximating *F*.
Produce a mesh approximating *F* with good angle quality.

Example: Start with the Stanford Bunny mesh (58,029 vertices). This has angles as low as 0.5^o. Output a new GradNormal mesh (34,834 vertices) with all angles larger than 10.4^o.

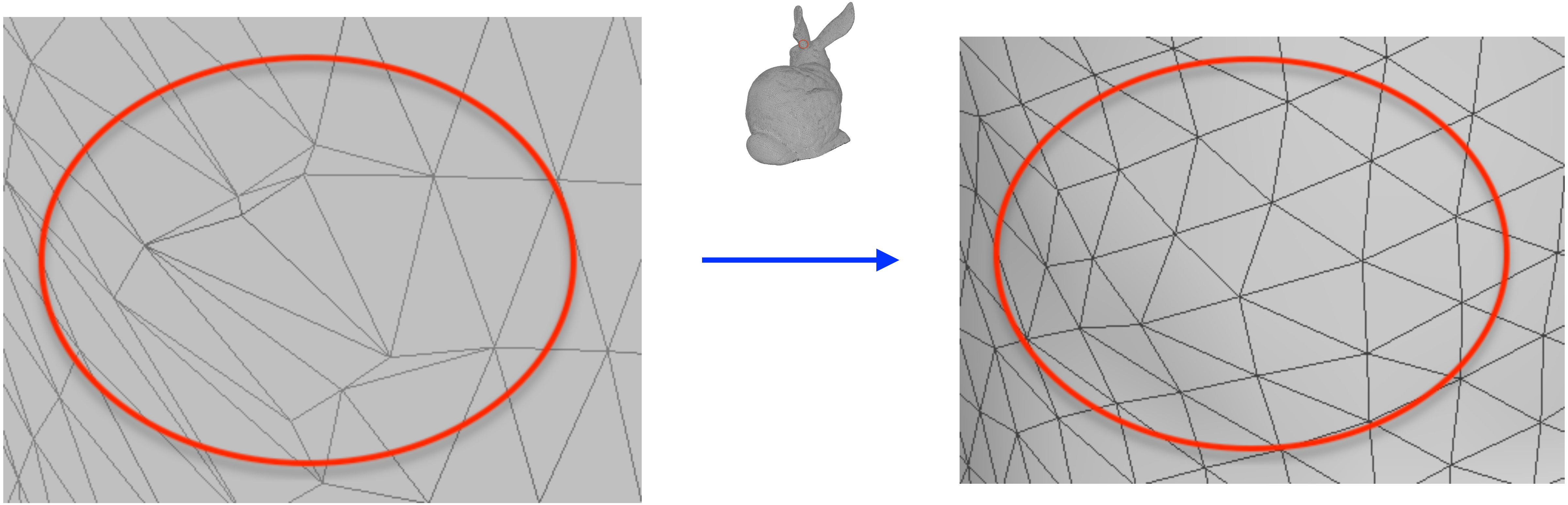


The GradNormal Algorithm for Mesh inputs



Original Bunny Mesh GradNormal Mesh

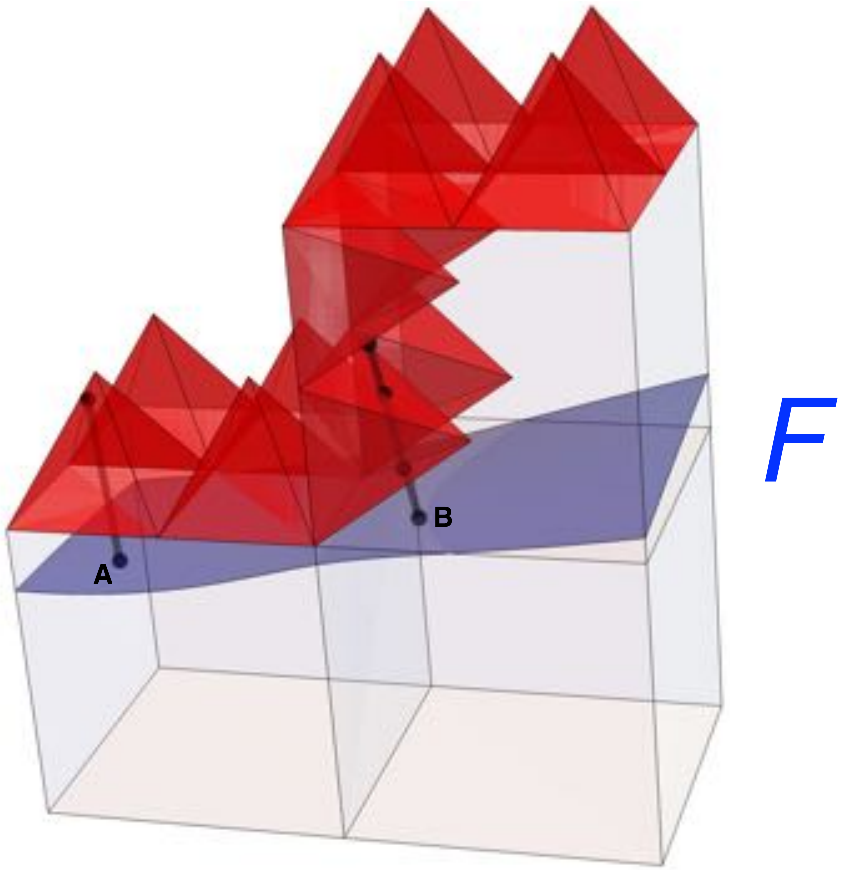
The GradNormal Algorithm for Mesh inputs



Original Mesh GradNormal Mesh

We achieve **Smooth** Approximation

A mesh using pyramids stacked on cubes can be used to approximate a surface *F* with a mesh consisting entirely of equilateral triangles. This optimizes angles but does not approximate normal directions.

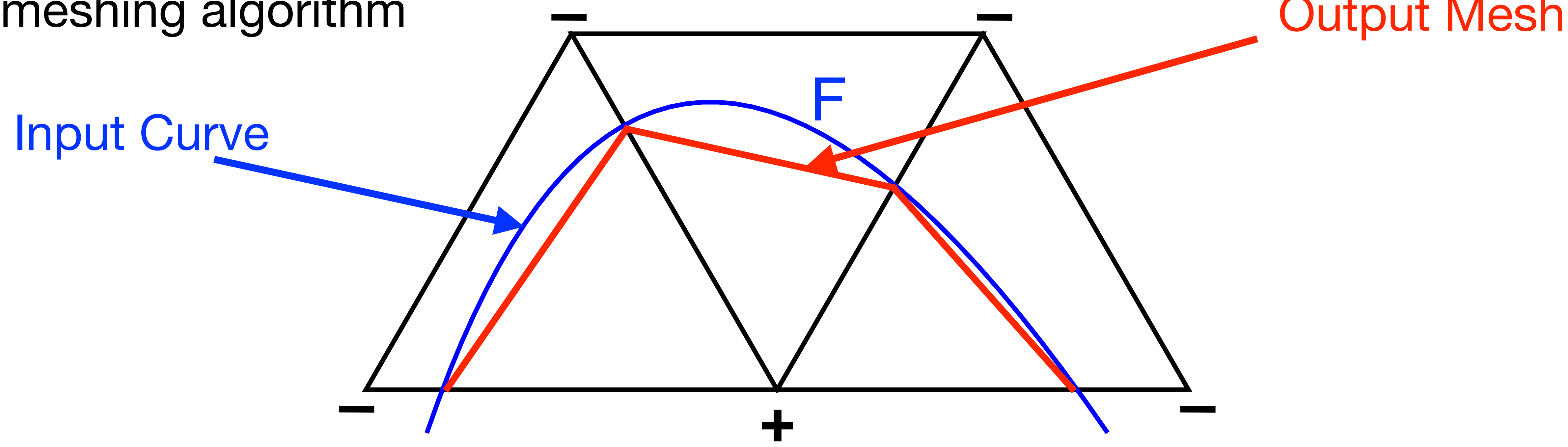


Our goal is to produce a mesh with
a) good angles
and
b) good tangent space approximation. (A piecewise-C^1 approximation)

Illustrate idea - dimension two

The algorithm has some resemblance to **Marching Tetrahedra.**

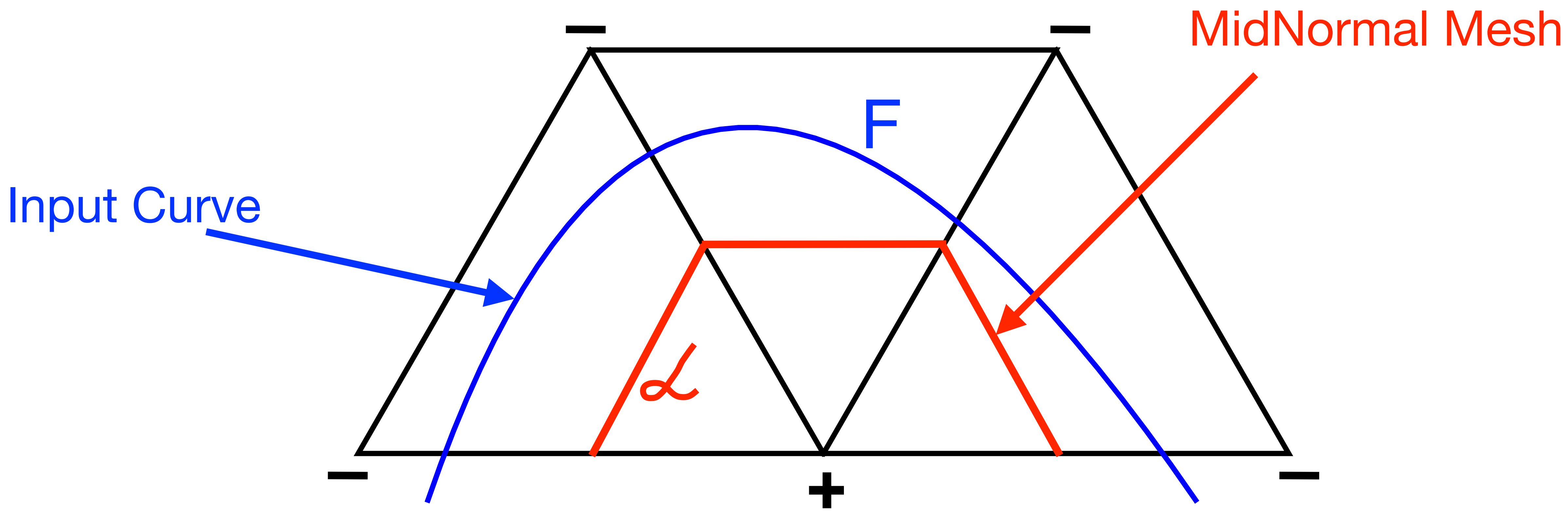
Marching Tetrahedra is a commonly used meshing algorithm



This method can result in very small angles in dimension three

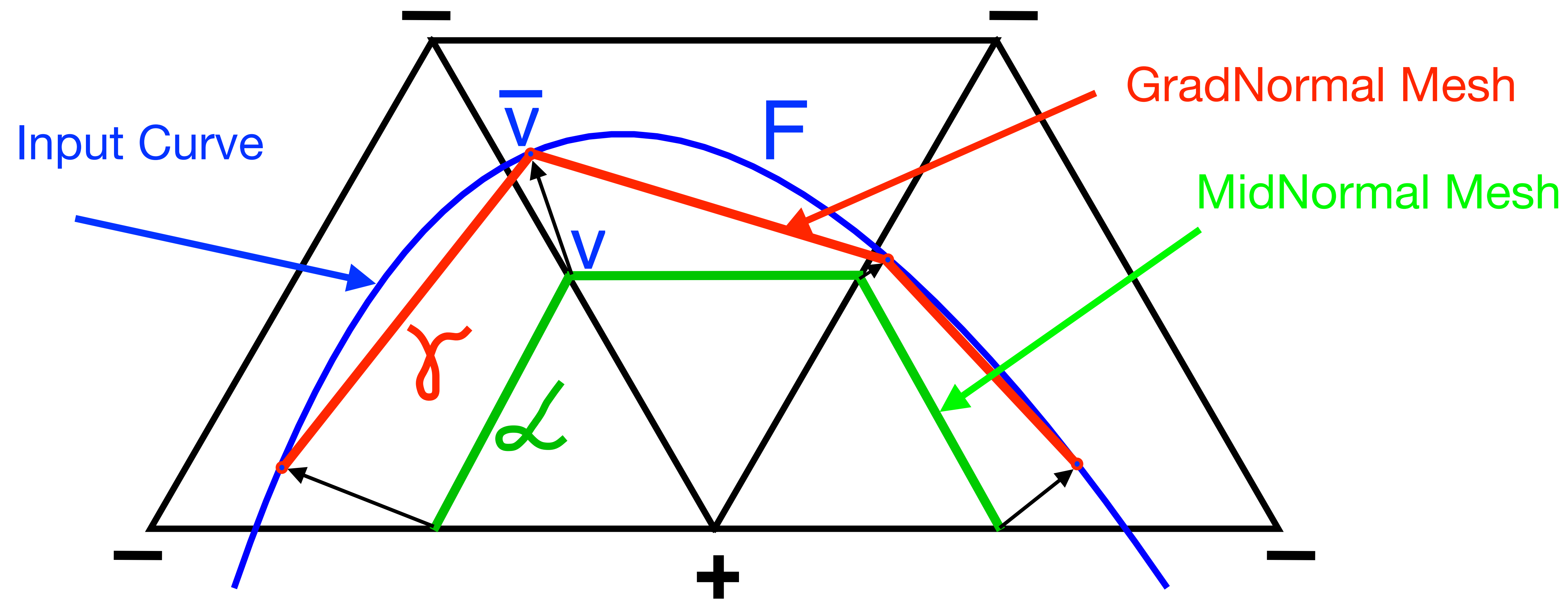
The GradNormal Algorithm in dimension two

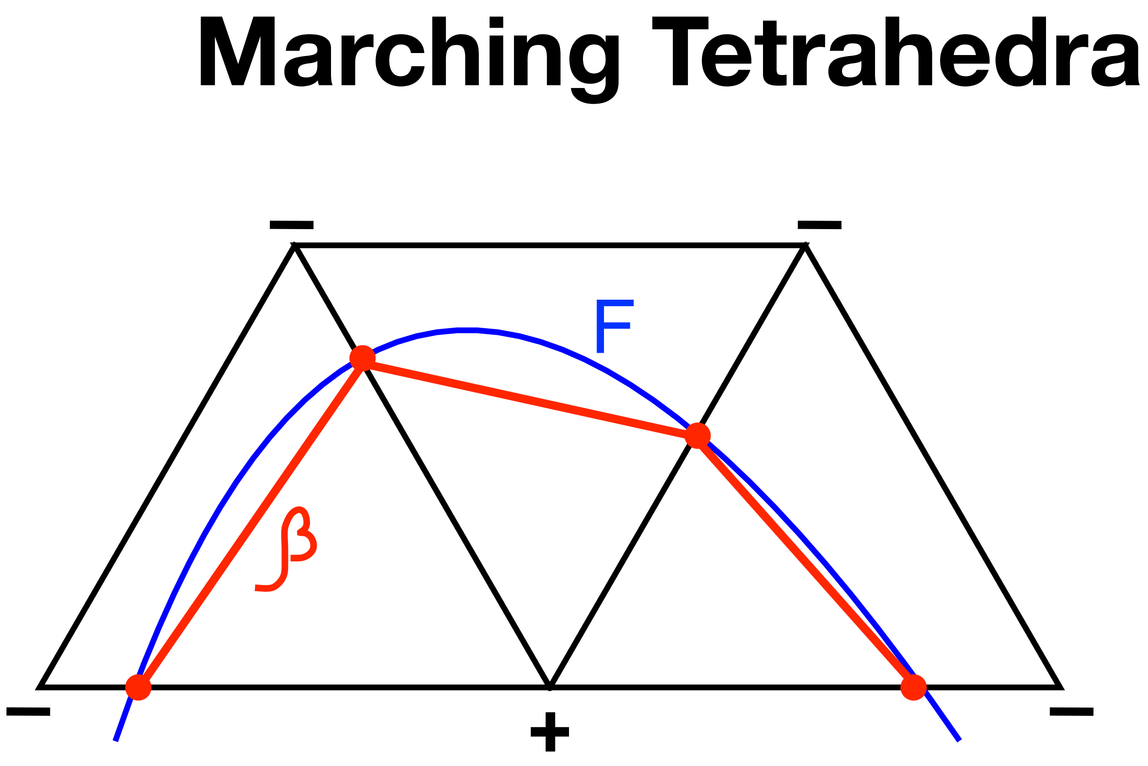
Step 1: Construct a MidNormal Mesh

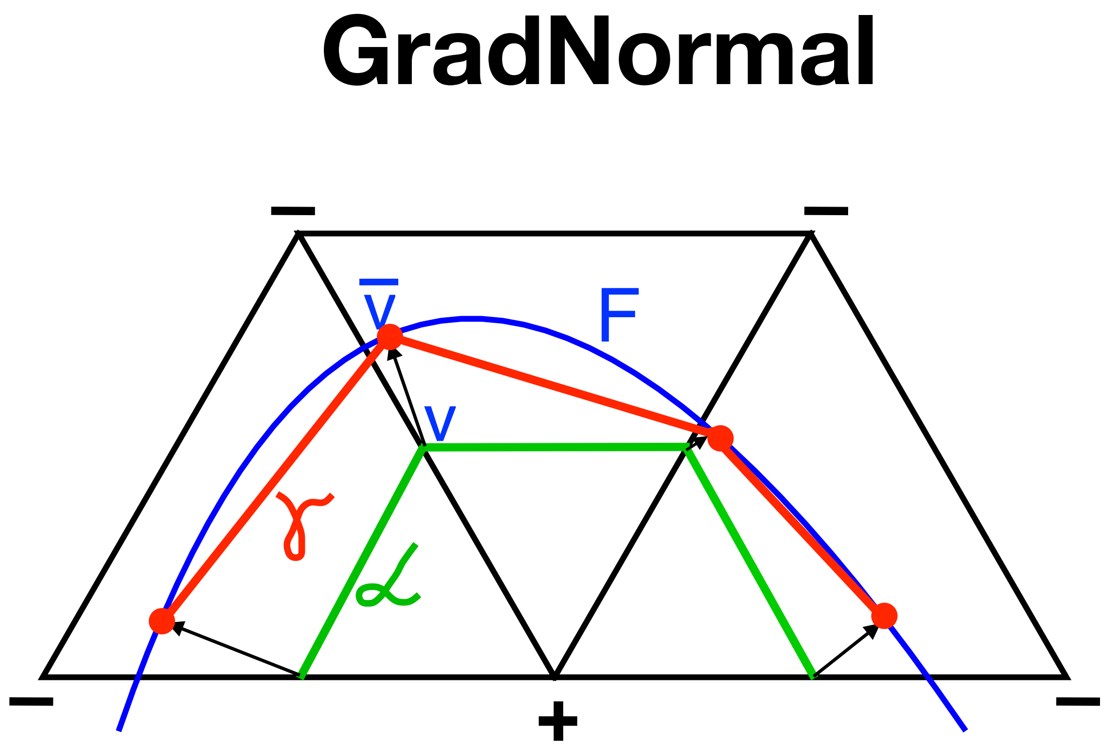


The GradNormal Algorithm in dimension two

Step 2: Project each MidNormal Mesh vertex to the closest point on F.




Marching Tetrahedra
F
β
+

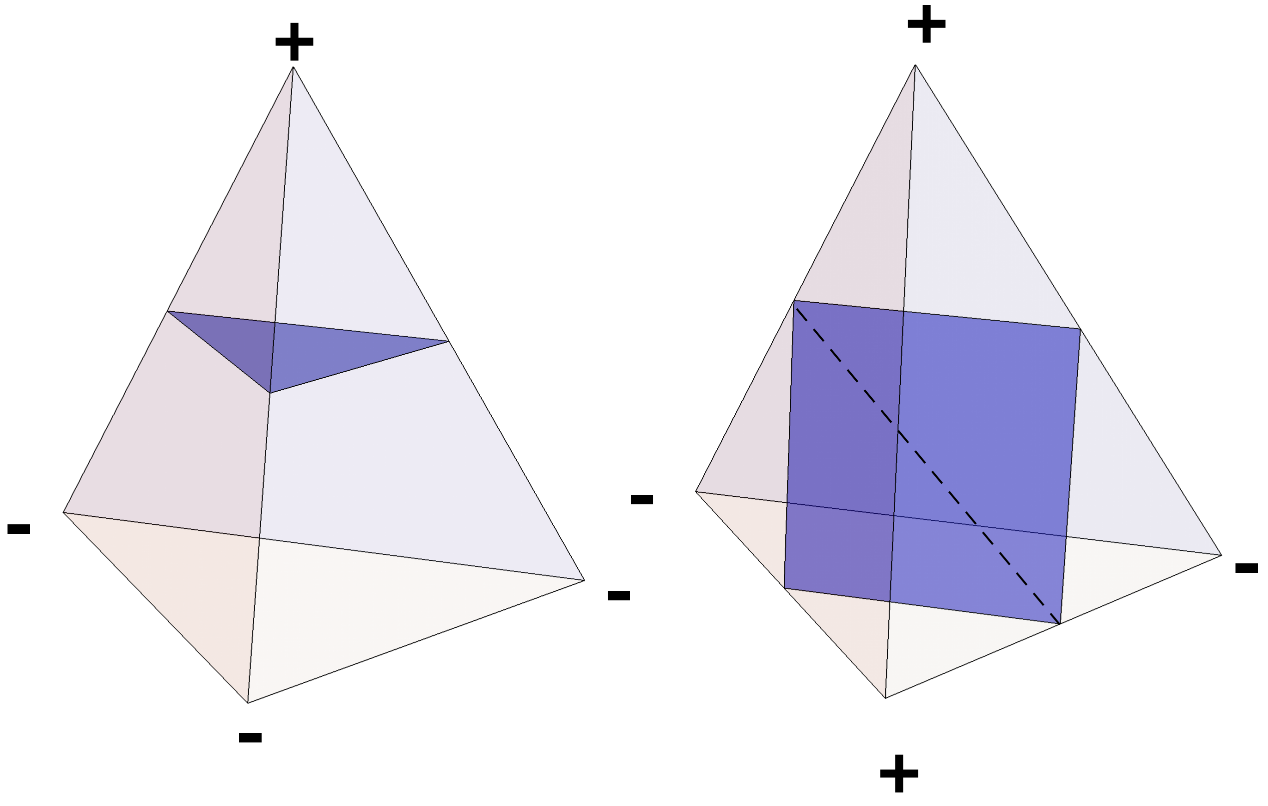

GradNormal
F
v̄
v
γ
α
+

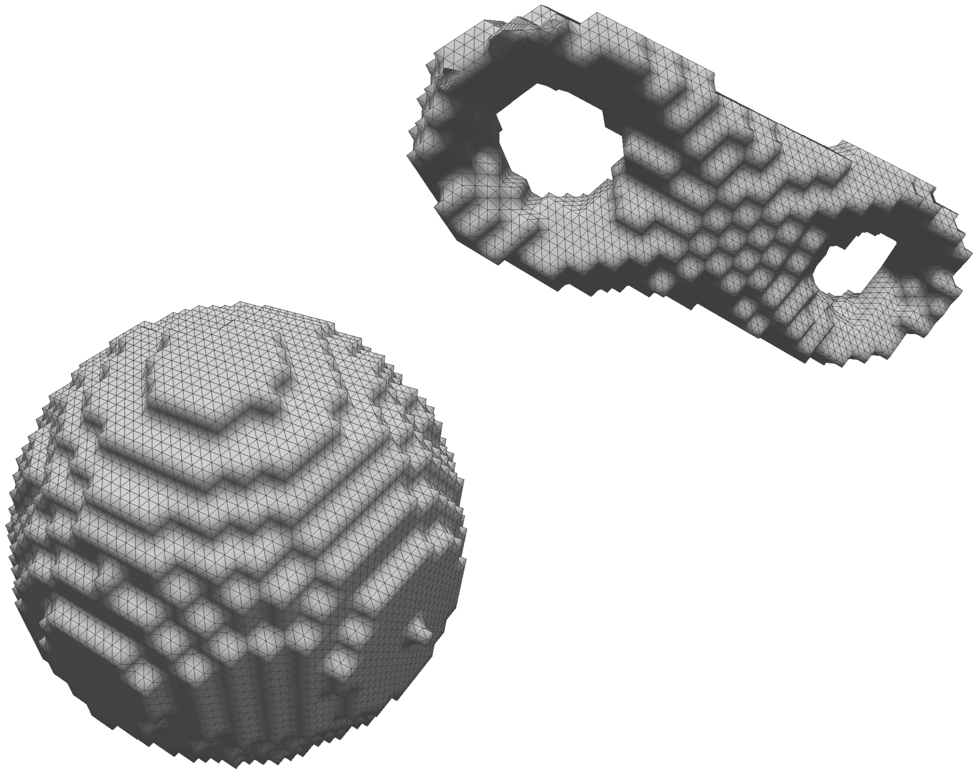
In dimension three there is an extra step

Step 3:

The GradNormal Algorithm in 3D

Step 1. Tile space with tetrahedra and construct a MidNormal Mesh.

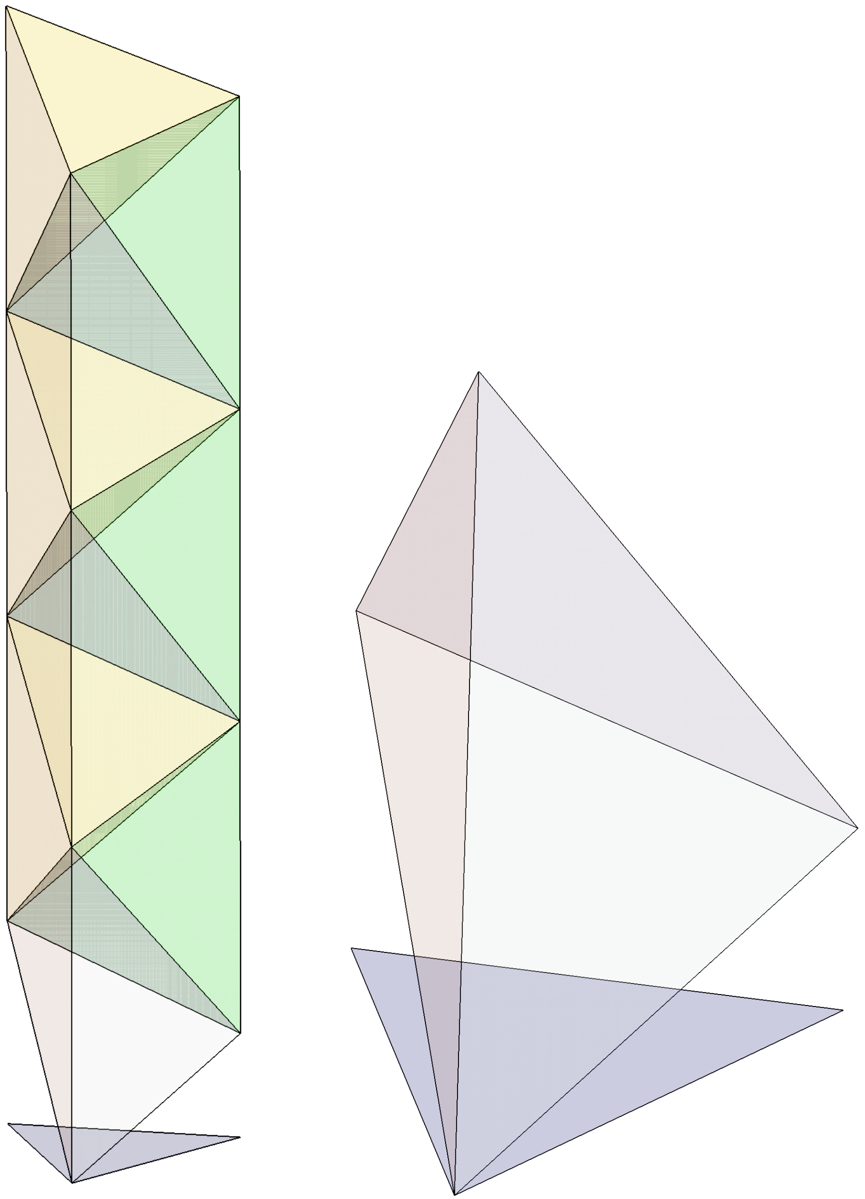




This requires a tetrahedral tiling. Which tiling should we use?

The GradNormal Algorithm in 3D

Step 1. Tile space with tetrahedra

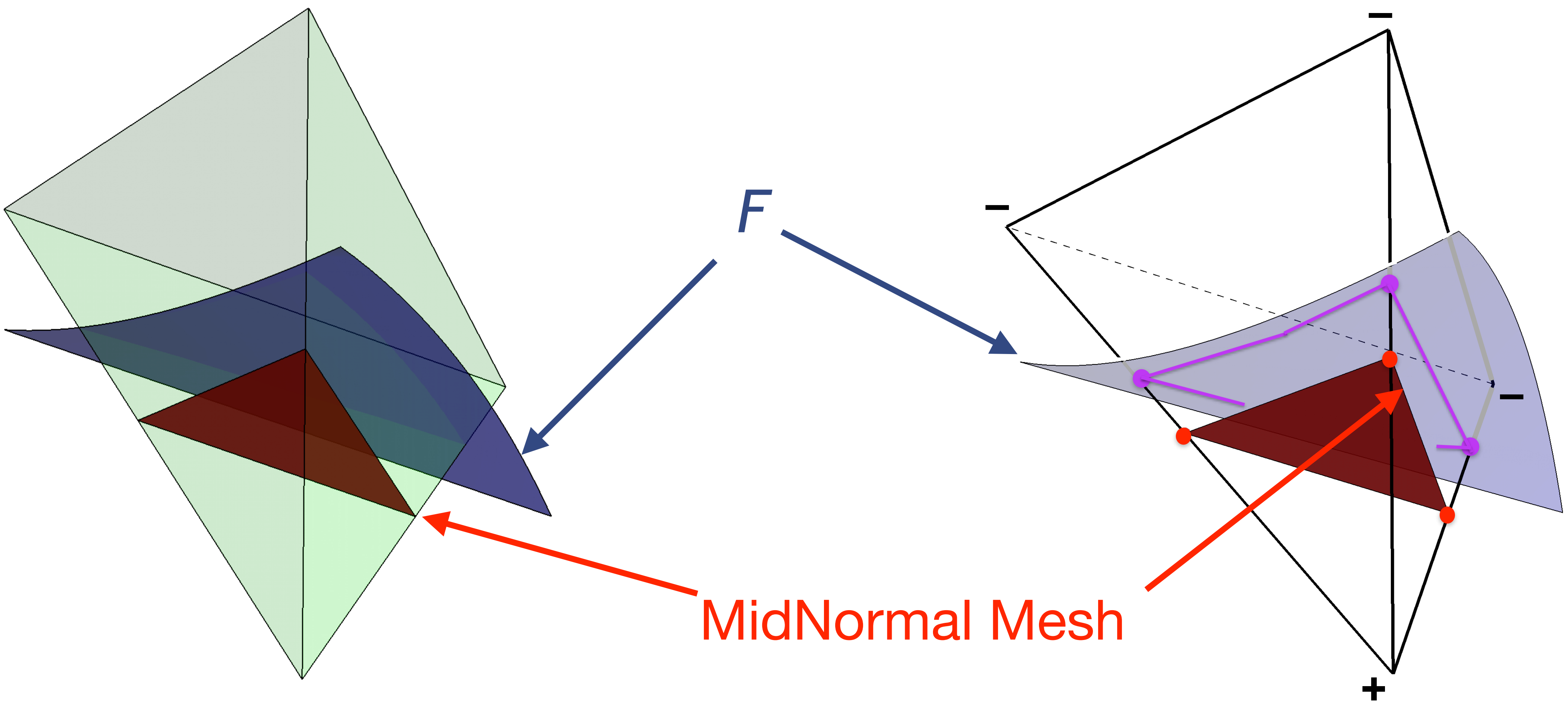


The tetrahedral tiling we use is one of a family of tilings we investigated, discovered by Goldberg in 1974. The optimal Goldberg tiling turns out to coincide with a tetrahedral tiling discovered by **Sommerville** in 1923, with vertices located at the body centered cubic lattice.

We have shown that no tetrahedral shape can significantly improve the angles achieved by our method.

The GradNormal Algorithm in 3D

Step 1. Construct a MidNormal Mesh

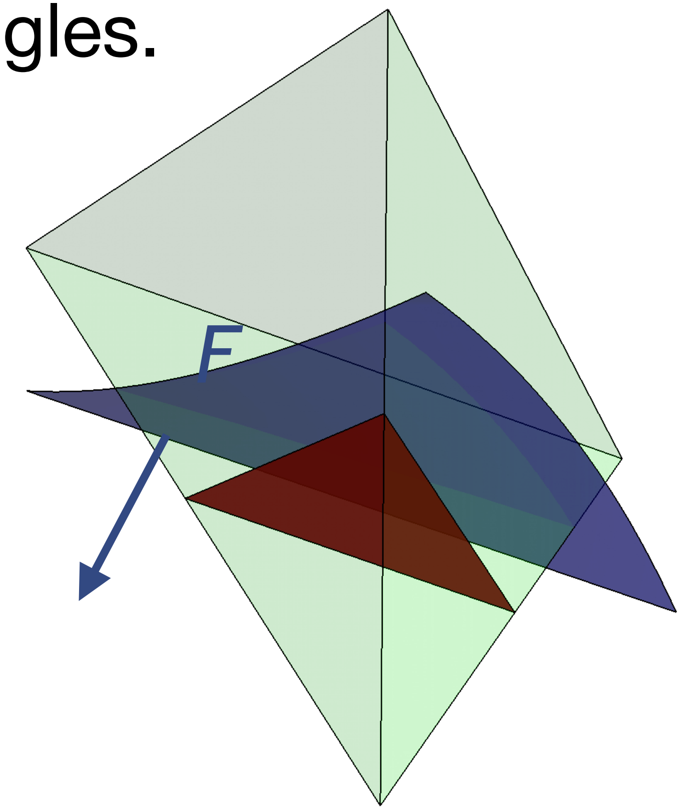


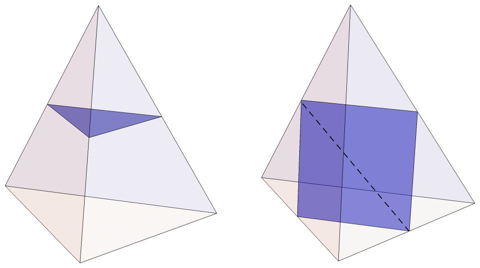
The GradNormal Algorithm in 3D

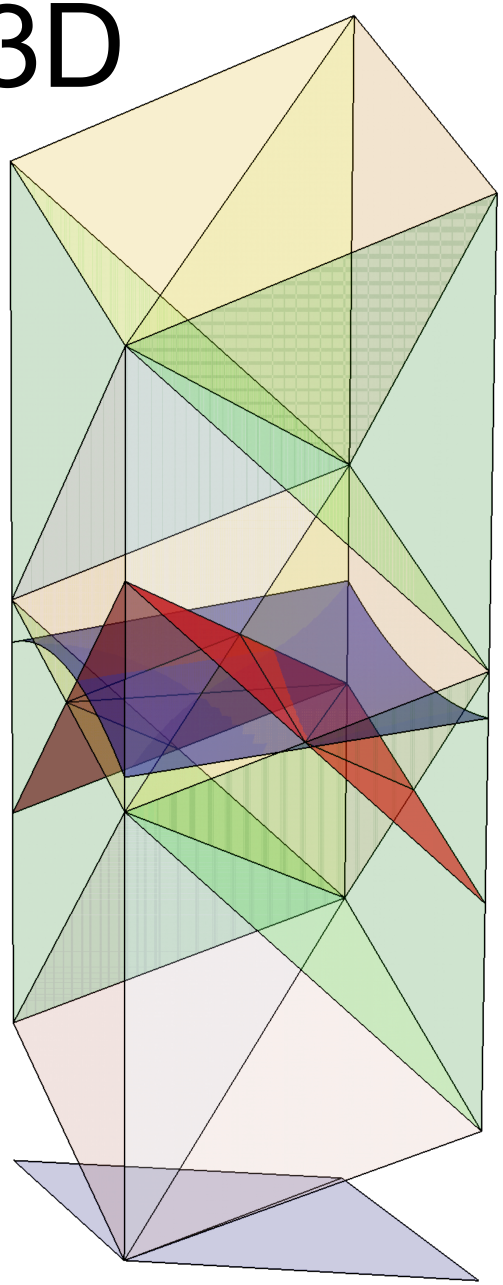
Step 1.

Construct a MidNormal Mesh.

Tile space with tetrahedra. Determine the MidNormal triangles.

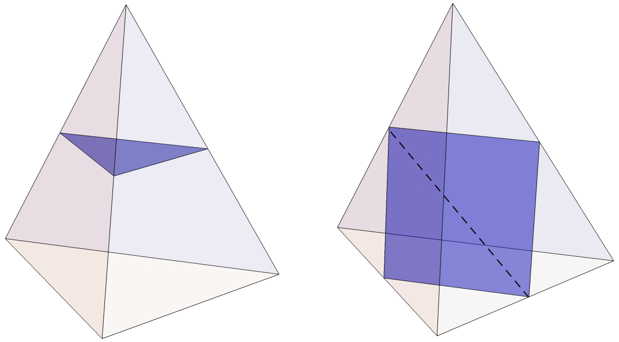


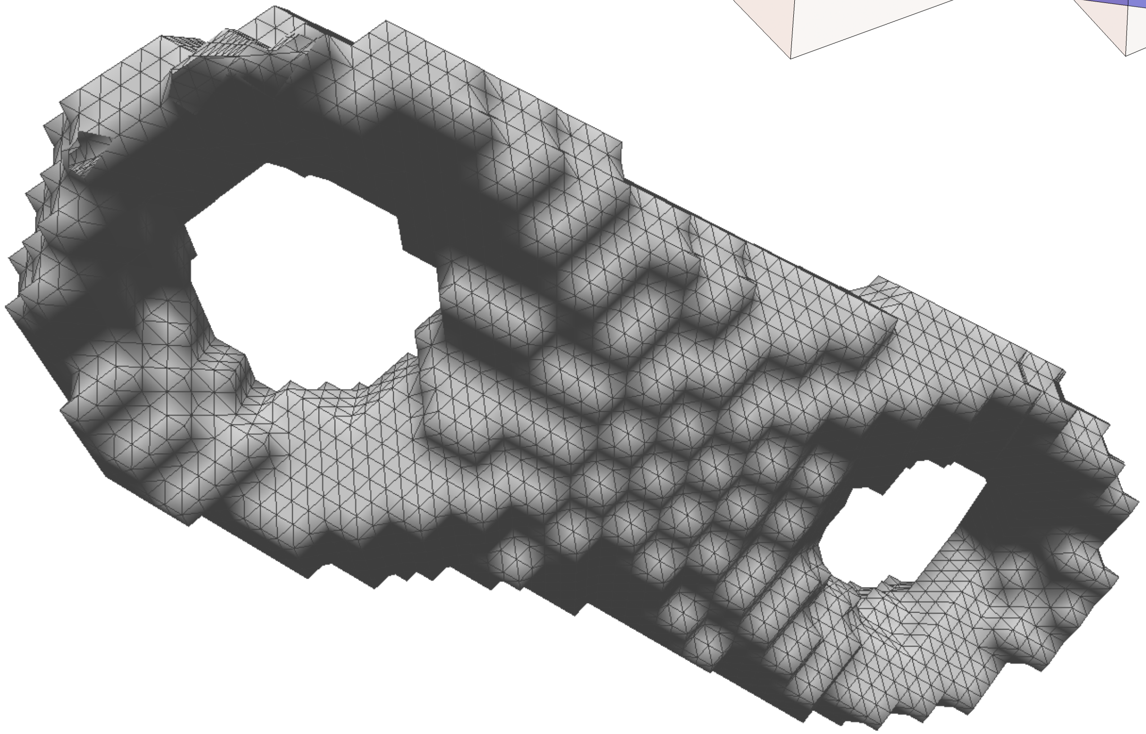


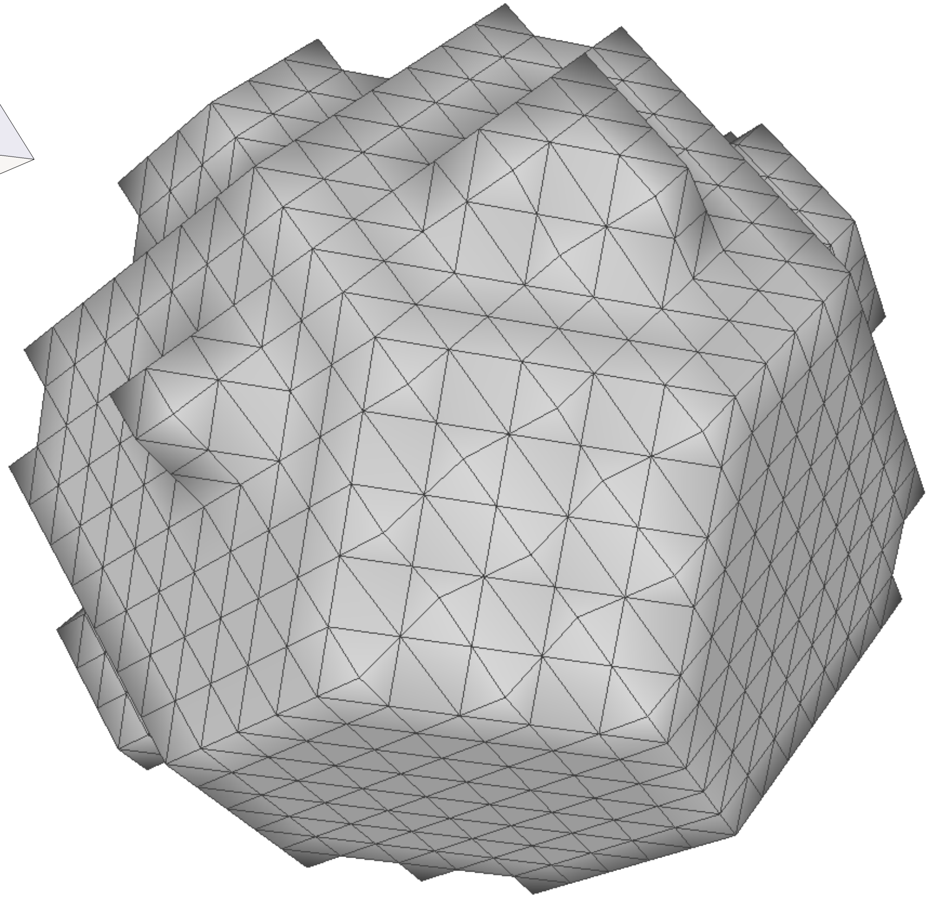


The GradNormal Algorithm in 3D

Step 1. A MidNormal Mesh

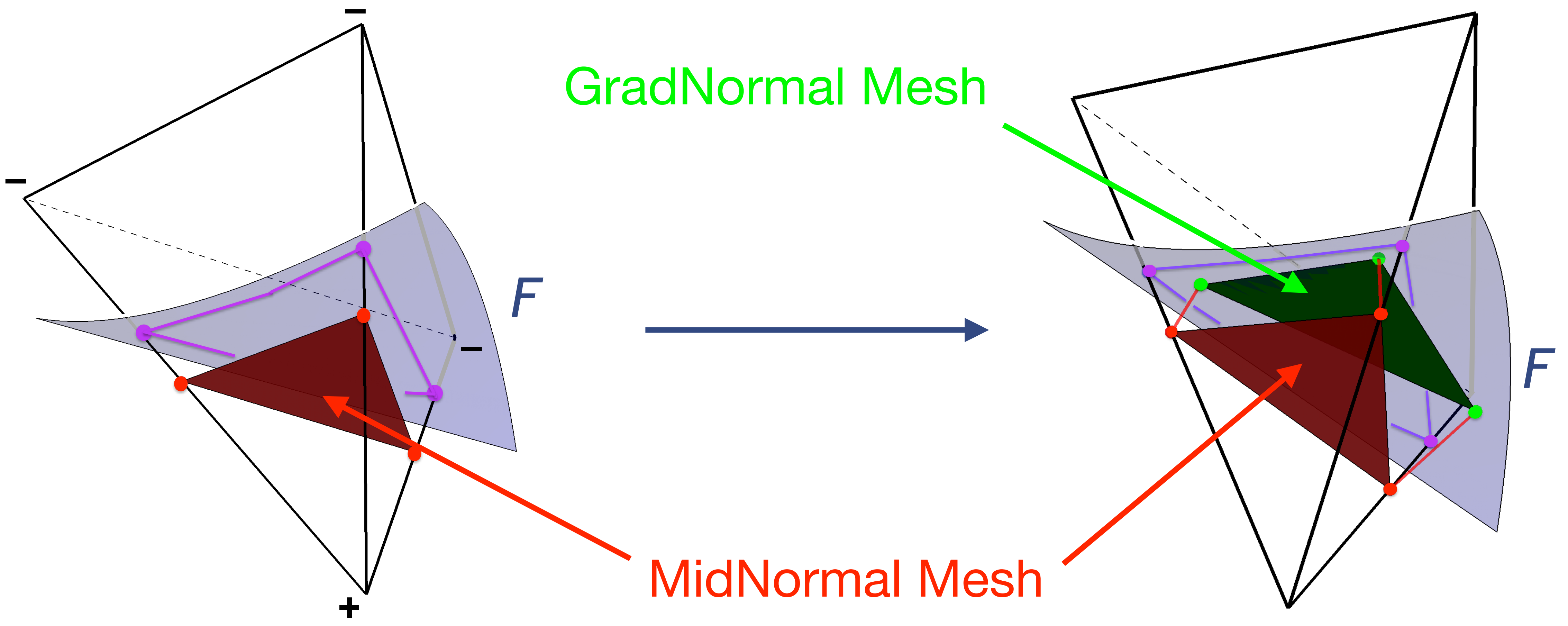




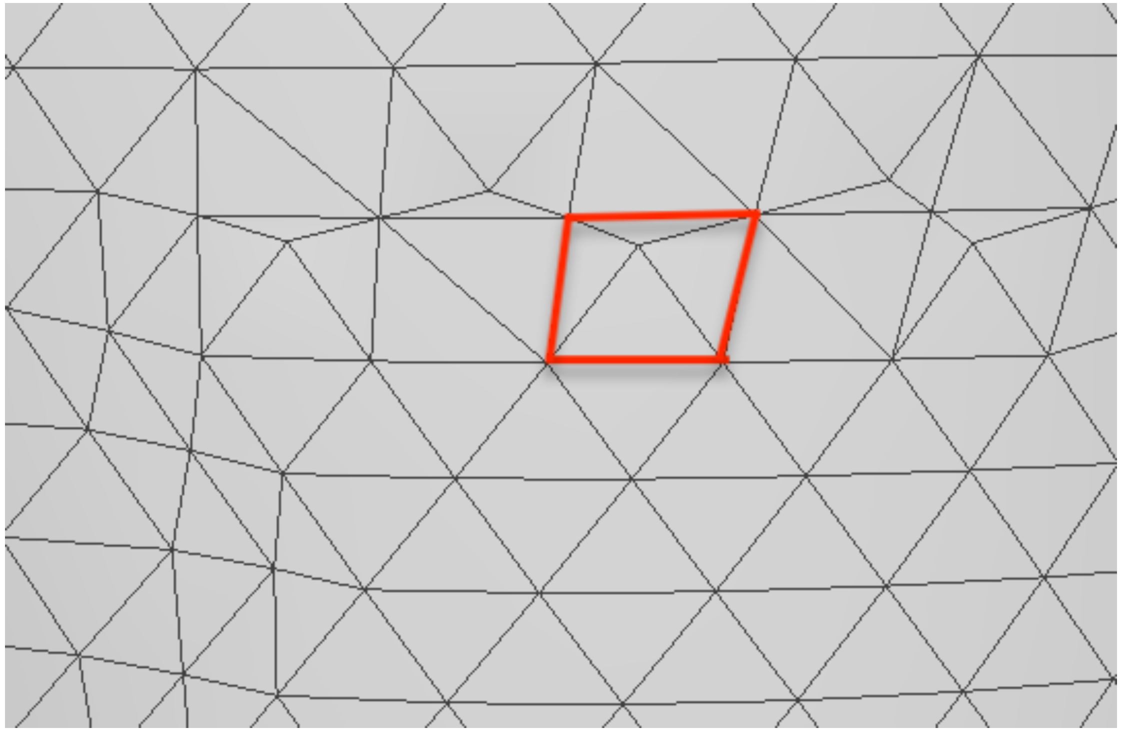


The GradNormal Algorithm in 3D

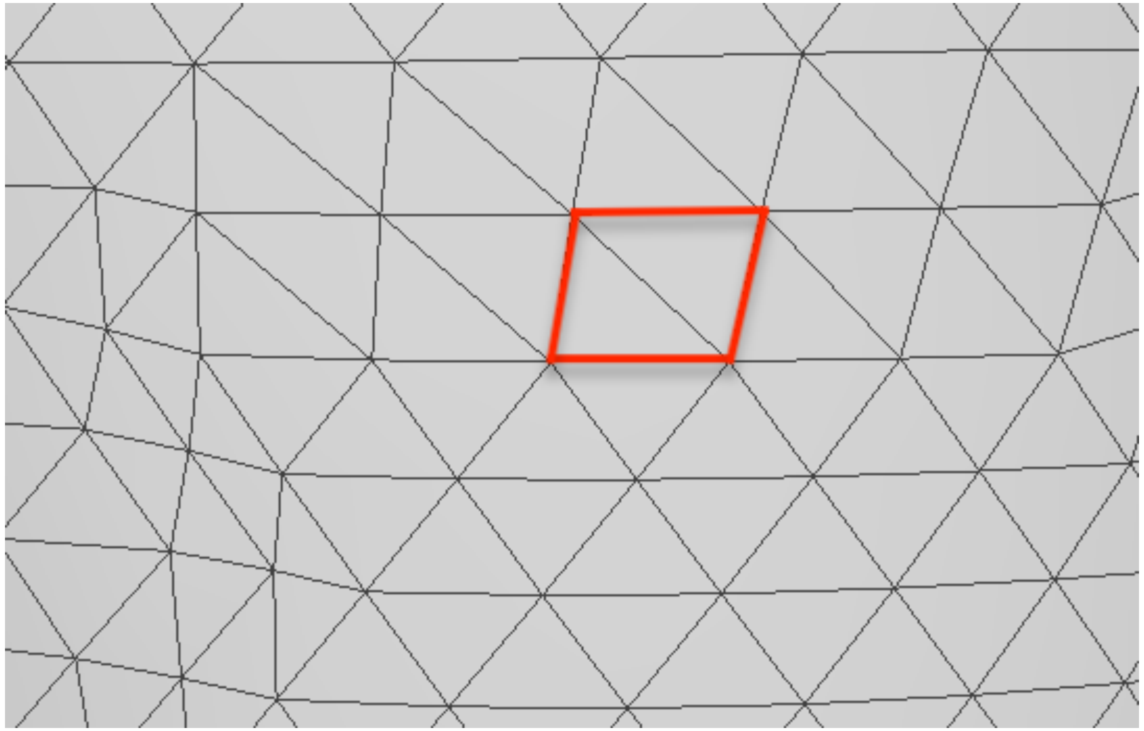
Step 2: Project each MidNormal Mesh vertex to the closest point on *F*.



Step 3: Remove all valence-four vertices

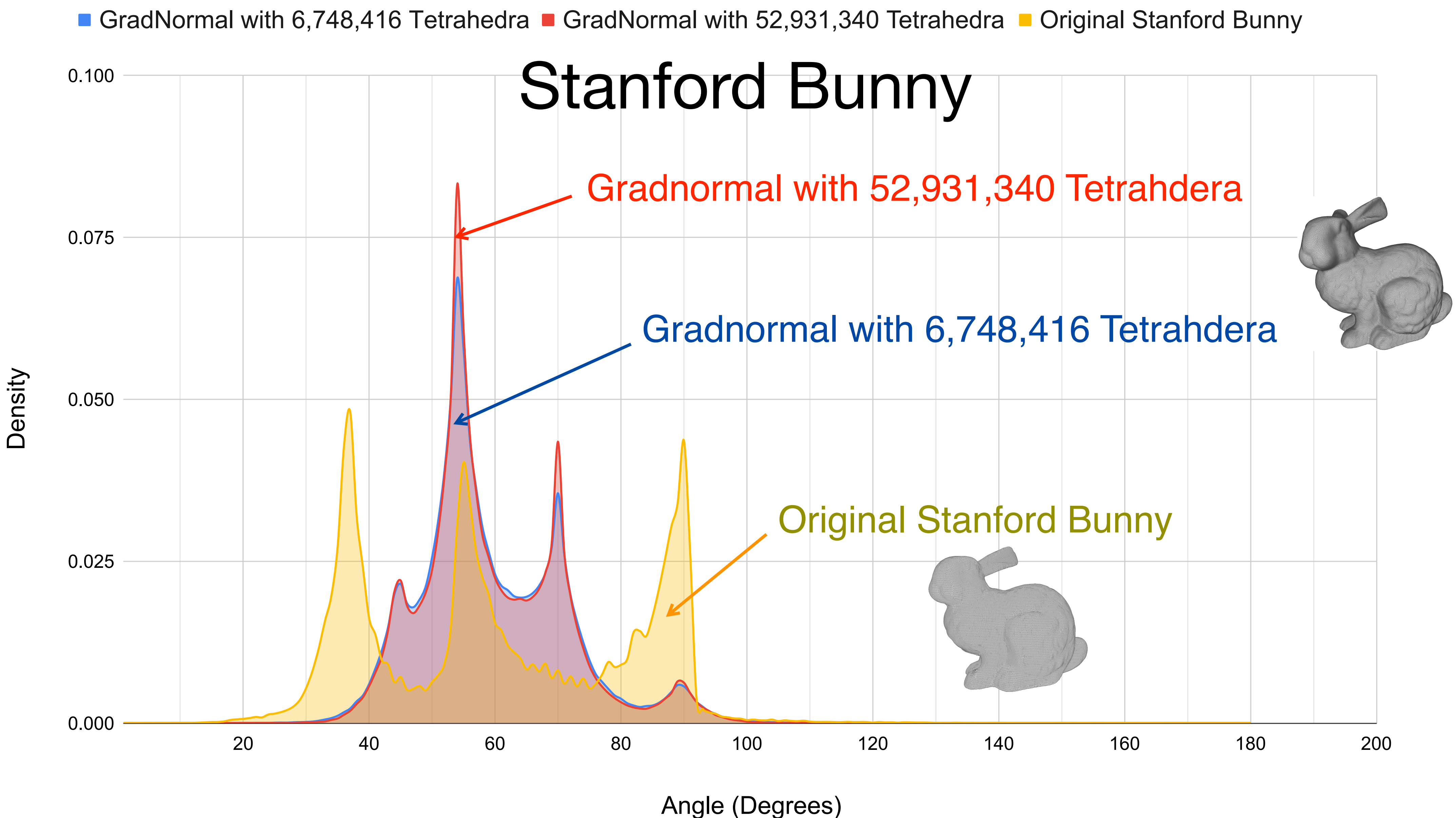


Before removal

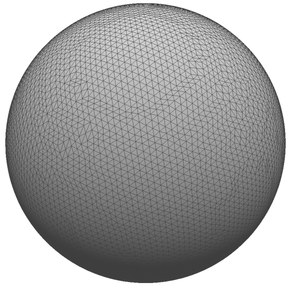


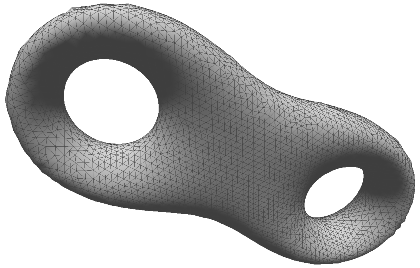
After

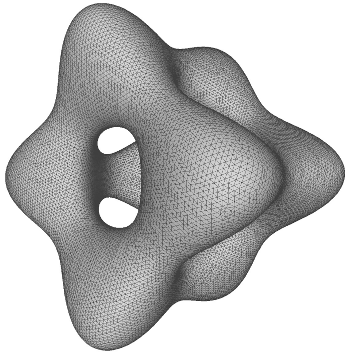
GradNormal Angle Distributions



GradNormal Angle Bounds (smooth F)







Genus	k_M	θ_m	θ_M	vertices	faces
0	0.23	33.0°	102.8°	1,082	1,988
	0.09	34.2°	101.3°	6,782	12,564
	0.05	35.4°	102.7°	27,104	50,300
	0.03	35.2°	101.1°	433,208	866,412
2	0.57	10.9°	153.8°	1,336	2,540
	0.29	22.2°	129.0°	5,438	10,306
	0.15	27.8°	118.8°	21,880	41,600
	0.08	31.7°	108.9°	87,802	166,898
5	0.8	26.6°	122.0°	7,318	14,652
	0.4	29.2°	113.4°	29,348	55,122
	0.2	30.8°	109.4°	117,878	235,772
	0.1	33.4°	104.4°	471,696	943,408

Hopefully Useful!