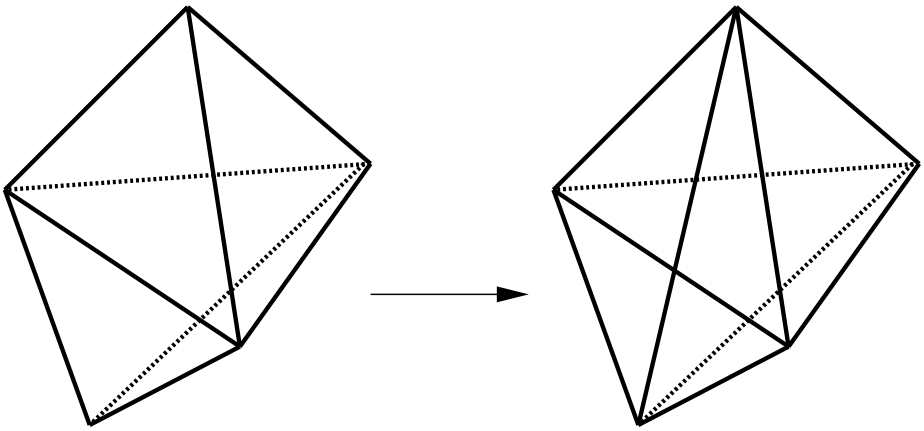
POST-GENERATION IMPROVEMENT

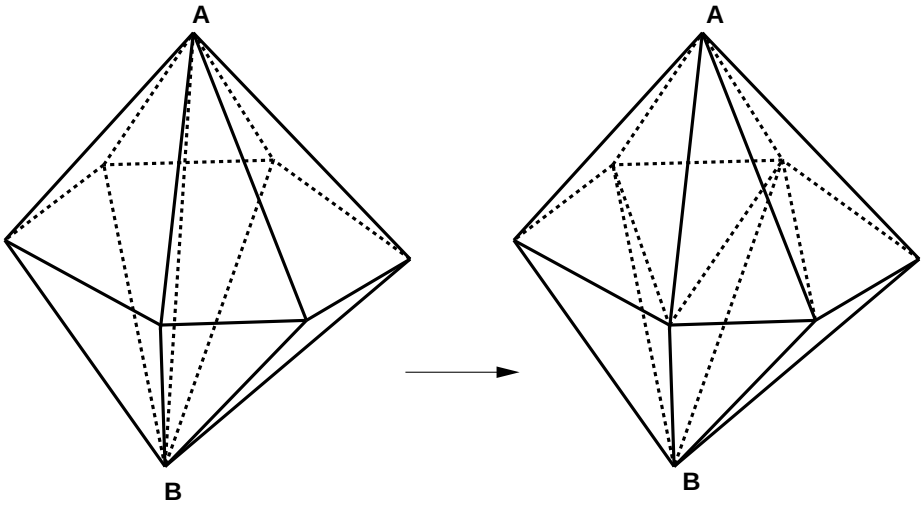
Techniques Used:

- Diagonal Swapping
- Removal of Bad Elements
- Laplacian Smoothing

DIAGONAL SWAPPING (1)



Diagonal Swap Case 2:3



Diagonal Swap Case 6:8

DIAGONAL SWAPPING (2)

Optimality Criterion:

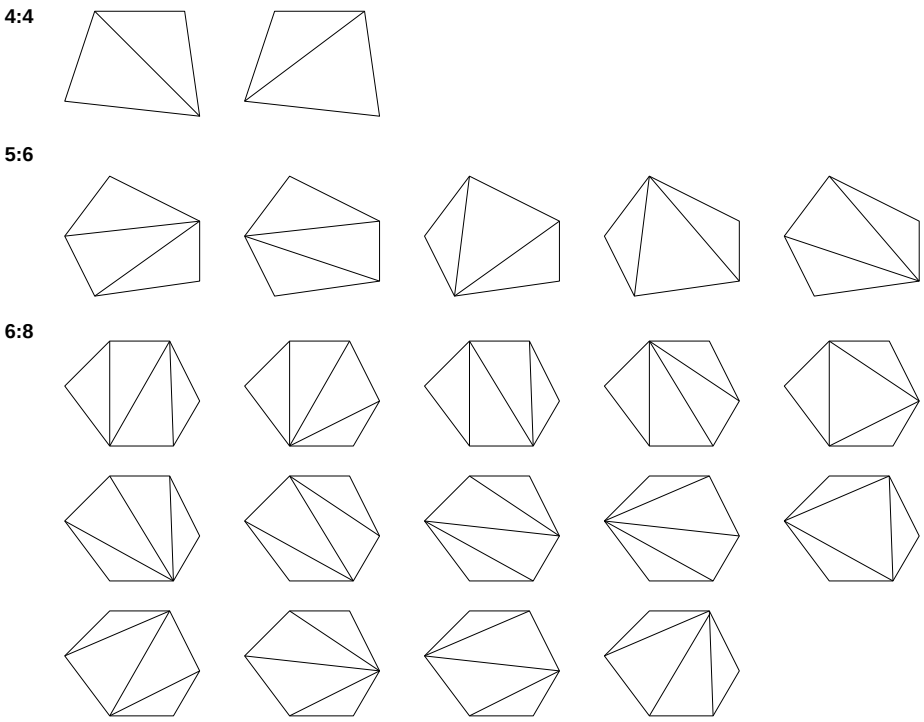
$$Q = \frac{h_{max} S}{V}$$

Many Cases $\Rightarrow$ Very CPU Intensive

CPU Reduction Techniques:

- Process Elements in Ordered Way, Starting From Worst
- Test Only Bad Elements ($Q > Q_{tol}$)
- Earliest Possible Rejection of Bad Combinations
- Mark Tested and Unswapped Elements

DIAGONAL SWAPPING (3)



Swapping Cases

DIAGONAL SWAPPING (4)

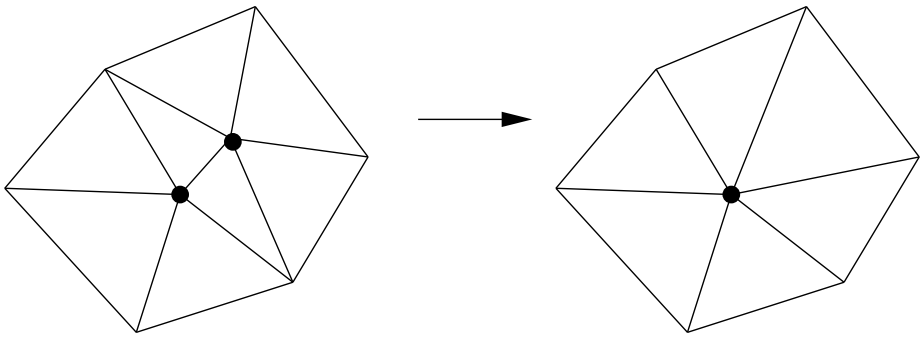
CPU Intensive Part: Testing

⇒ **Parallelize Testing**

- `WHILE`: Bad, Untested Elements in Heap:
  - Check Worst Elements (Parallel);
  - Reconnect If Improved and Neighbouring Elements Untouched (Scalar, Integer);
  - Remember Swapped Elements;
- `ENDWHILE`

REMOVAL OF BAD ELEMENTS

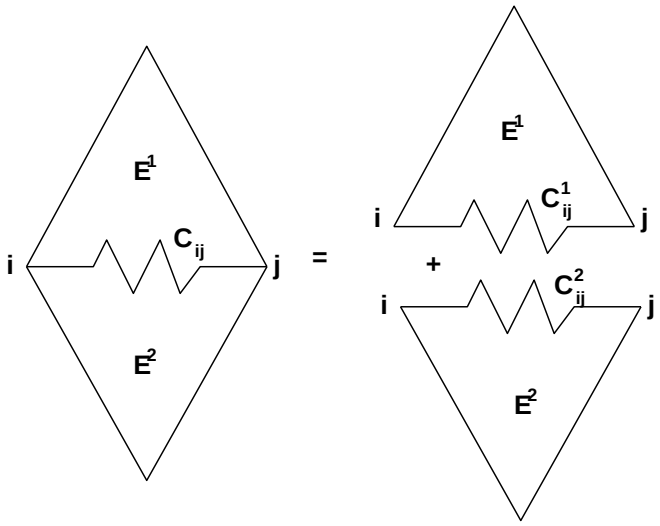
How: Collapse Edge



Removal of Element

- Fast
- Pre-Smoothing/Pre-Optimization

LAPLACIAN SMOOTHING



Treat Edges as Springs $\Rightarrow$

$$\mathbf{f}_i = c \sum_{j=1}^{ns_i} (\mathbf{x}_j - \mathbf{x}_i)$$

$$\Delta \mathbf{x}_i = \Delta t \frac{1}{ns_i} \mathbf{f}_i \quad .$$

- On Boundary: $\Delta \mathbf{x} = 0$
- 3-4 passes yield acceptable mesh
- Mesh quality and uniformity improved
- Appearance of negative elements in 3-D ($< 0.1\%$)

$\Rightarrow$ Use Spring Analogy Smoothing in combination with Sweep and Retry