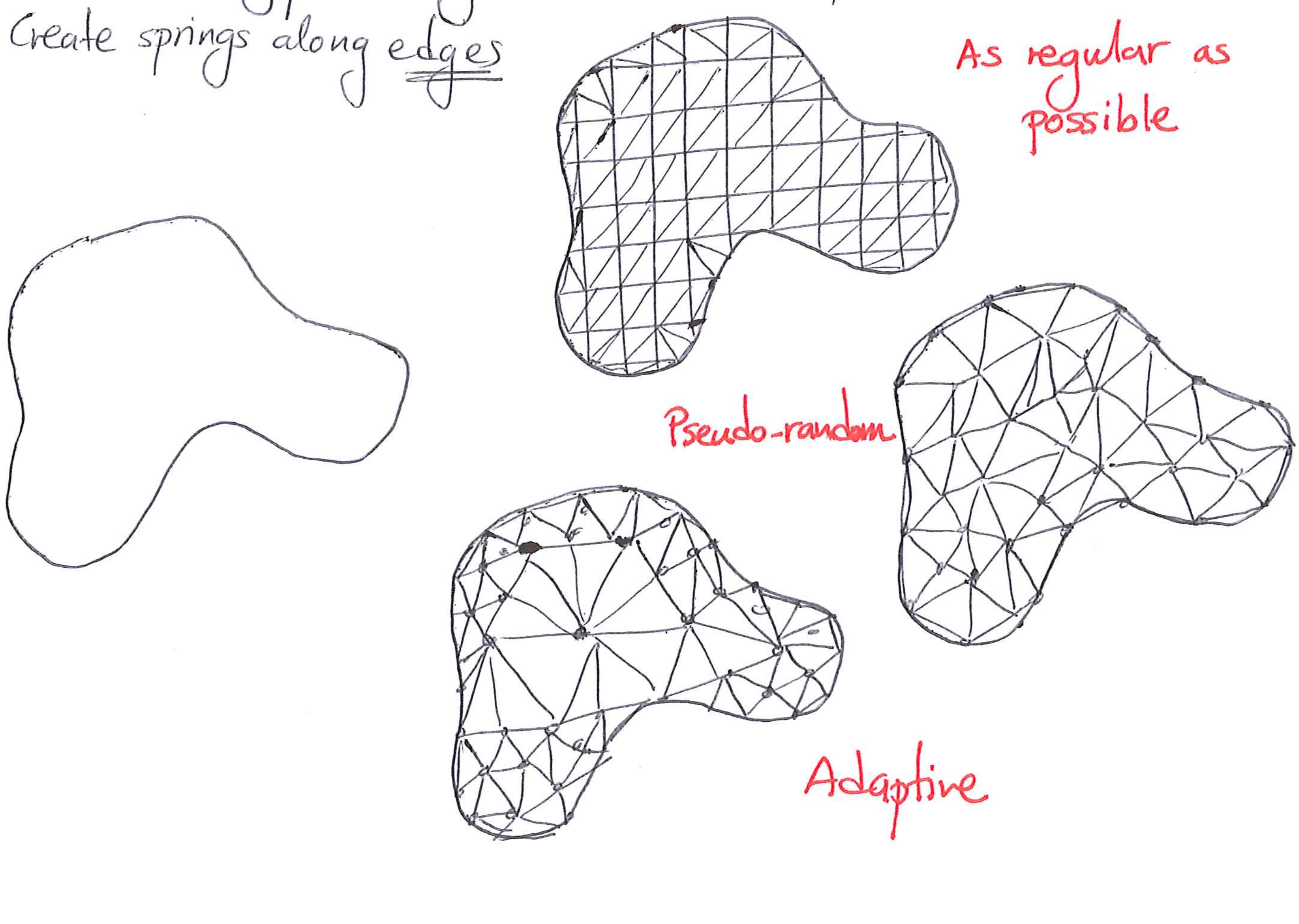
# Cloth simulation as a mass spring system

- Most of concepts & techniques apply.
- Network of springs need to cover/span the simulated surface
- Special springs and/or forces
- Material parameter selection
- Introduction to collision processing

## Design of spring network for cloth simulation

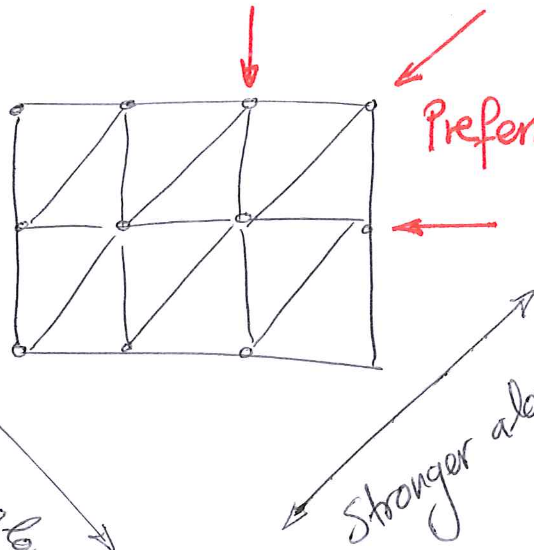
→ Basic methodology: Triangulate the surface/area

→ Create springs along edges



## Things to be aware of:

- Spurious anisotropy

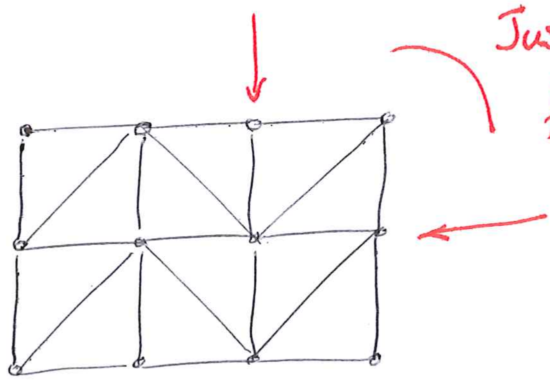


Preferred fold directions

Stronger along this direction

Weaker along this direction

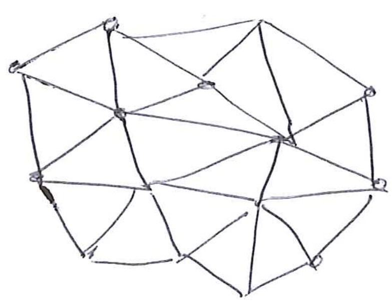
Slightly better:



Just 2 preferred fold directions

No direction is particularly stronger or weaker (or, at least, not by much).

Even better??



No fold lines? (they're just curved)

No direction is particularly stronger or weaker.

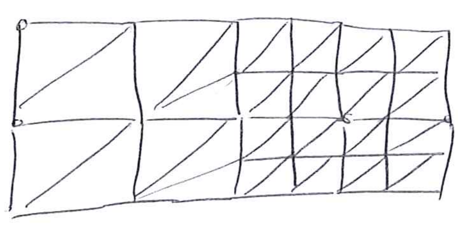
## Hazards:

- Nonuniform density


If Young's modulus is the same for all springs, and we create springs on every edge, the model on the left will be weaker, and the one on the right stronger / more stiff.

But, by how much?

It can be really hard to adjust the Young's modulus on a spring-by-spring basis, to compensate for this difference in apparent strength ⇒ Problem when mesh transitions from large trianges / small density to small triangles / large density.



When possible it is better to use a uniform density and avoid the challenge of ad-hoc adjustments to material parameters for compensation.

# Hazards (cont'd)

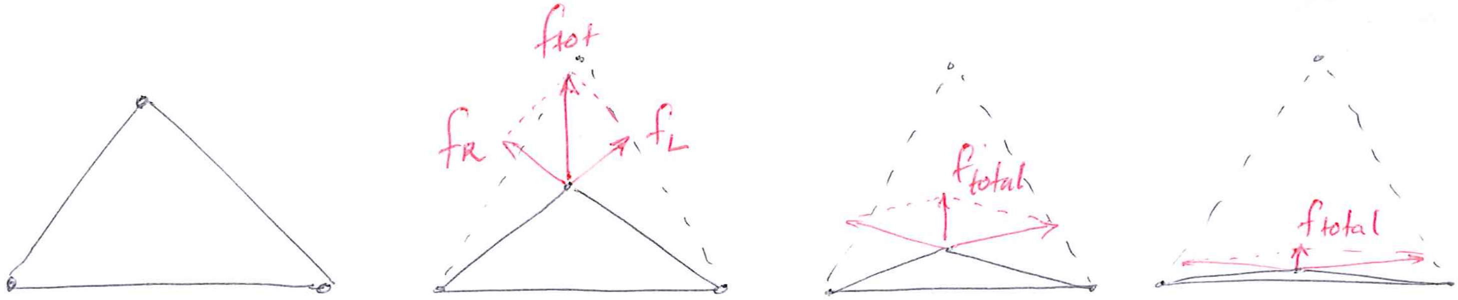
- Anisotropy

↳ Anisotropic tessellations like this artificially increase the stiffness (stretch resistance) of the mesh

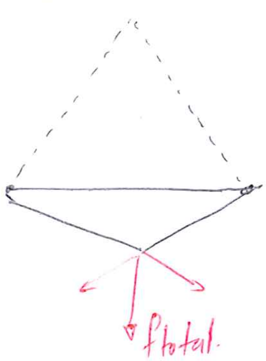
⇒ Solution (optionally): FEM methods (later lectures)

- Compression resistance

As a triangle, rigged with springs along the edges, gets compressed, its resistance to further compression is in fact diminished!




Even worse, when the triangle has "inverted", the same spring forces cause it to invert even more:

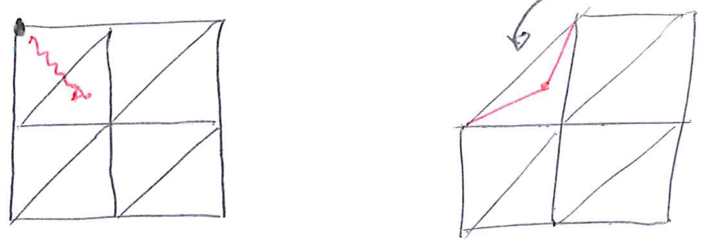



Why do we consider this "inversion"? Isn't this just a "rotated" version of the same triangle?

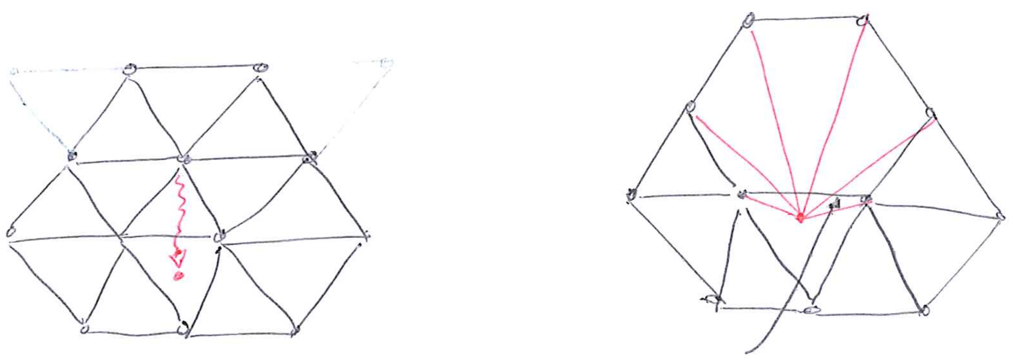
The reason that this "inversion" is problematic, is that it may cause triangles to "overlap"

e.g.

2 triangles overlapping



or



3 triangles overlapping.

(The situation is a bit different for surfaces in 3D, which don't really "overlap", but rather "fold" over one another. Still, this is a problem).

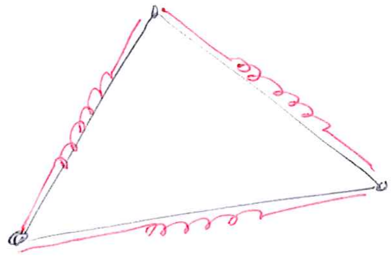
Solutions?

I. Use something different than mass/spring systems (e.g. Finite Elements, in lectures to follow)

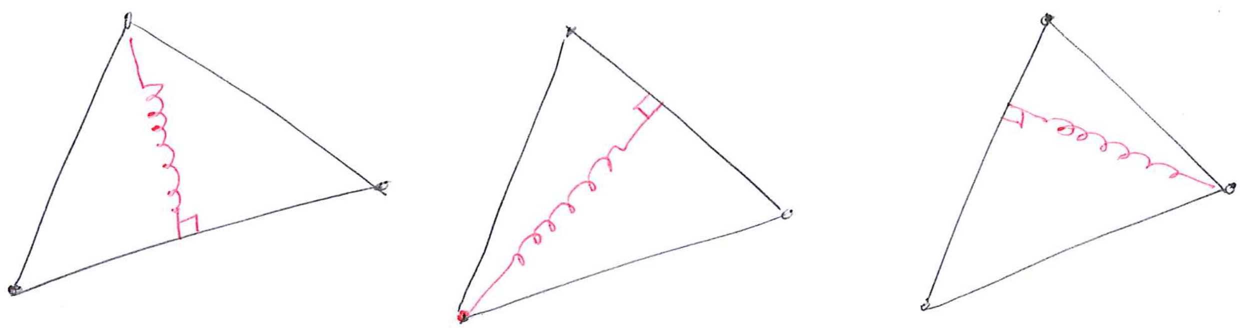
II. Use "auxiliary" springs to fight compression

# Basic altitude springs

→ Springs positioned along triangle edges := "Edge Springs"



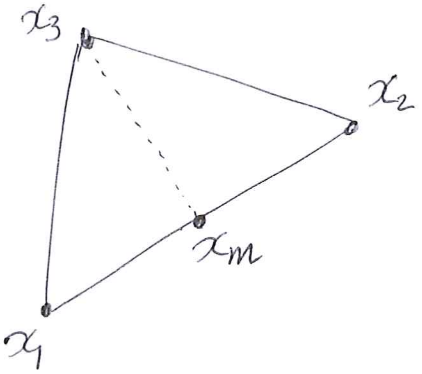
→ "Altitude Springs" Connect a triangle vertex to its projection on the opposite edge



## Implementation

→ The "projected" endpoint of an altitude spring is not fixed; it is "sliding" on the opposite edge

→ At every time step, we compute an "interpolation ratio" as follows

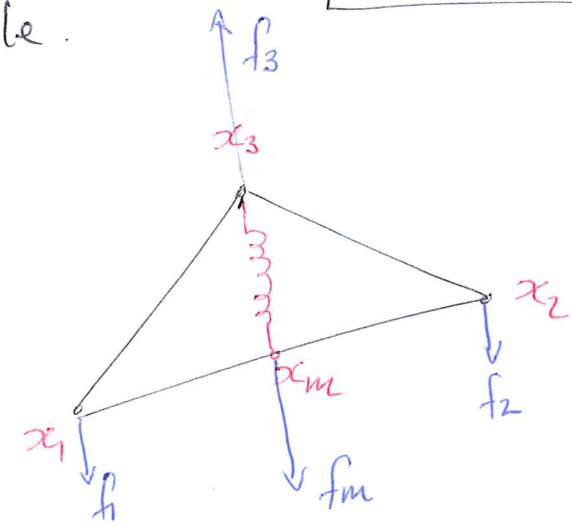



$$\vec{x}_m = \text{Project\_Pt\_On\_Segment}(x_3 ; (x_1, x_2))$$

$$\lambda = \frac{\| x_m - x_1 \|}{\| x_2 - x_1 \|}$$

Thus $\vec{x}_m = (1-\lambda)\, \vec{x}_1 + \lambda\, \vec{x}_2$

→ Restlength: Computed from length of altitude in the "natural" shape of the triangle.




→ Spring force

$$f_3^a = -k\left(\frac{\|x_3 - x_m\|}{l_0} - 1\right)\frac{x_3 - x_m}{\|x_3 - x_m\|}$$

$$f_m^a = -f_3^a$$

⇒ But, $x_m$ is not a "real" particle!

↳ Distribute the force $f_m$ to endpoints $x_1$ & $x_2$ proportionately, according to the interpolation fraction

$$f_1^a = (1-\lambda) f_m^a$$
$$f_2^a = \lambda f_m^a$$

→ Damping force

Step 1: Compute velocity of projected point

$$V_m = (1-\lambda) v_1 + \lambda v_2$$

Step 2:

Compute damping force

$$f_3^d = -bnn^T (V_3 - V_m) \qquad n = \frac{x_m - x_3}{\|x_m - x_3\|}$$

$$f_m^d = -f_3^d$$

Step 3: Distribute:

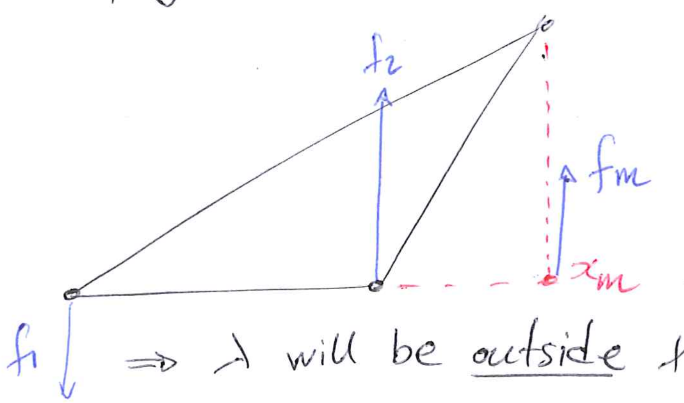
$$f_1^d = (1-\lambda) f_m^d$$
$$f_2^d = \lambda f_m^d$$

# Which springs to use?

## A. All three

**Pros:**
- Forces are continuous as a function of time (no switching on/off)
- Simple, uniform treatment

**Cons:**
- If projection falls outside edge ⇒ possible problems




⇒ $\lambda$ will be outside the range $(0,1)$

⇒ $f_1, f_2$ will point in opposite directions

⇒ One of $f_1/f_2$ will be larger in magnitude than $f_m$, itself.

- If several altitudes are compressed/inverted, there may be conflicting forces in how to "recover" a reasonable shape
- The triangle may become "too rigid"

## B. Shortest altitude only

**Pros:**
- Shortest altitude always projects inside an edge!
- No conflict in which direction to un-compress/un-invert.

**Cons:** Switching springs on/off may result in noticeable jitter!