WHAT IS THE "GEOMETRY" OF CONTACT?

by
Kenny Erleben
2014

A PRACTICAL DEFINITION?

- Contact "surfaces" are represented (sampled) by a discrete set of contact points that each has
  - A spatial position
  - A sense of direction
  - A labelling of the objects making up the contact
  - A measure of proximity (distance) between the objects
- Practical problem: How do we compute such representations?

SOME PRACTICAL EXAMPLES

- Closed form solutions for simple primitive shapes: SAT-like, box vs box, sphere vs. sphere.

- Iterative methods for convex shapes: GJK-type of algorithms, EPA, sampling of direction tables.

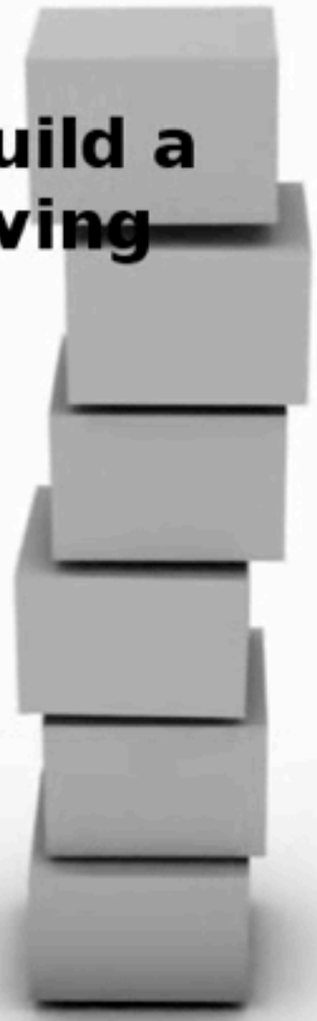
- Feature based approaches: vertex-face and edge-edge pairs, I-collide, V-clip, "point in element" style, polygon clipping.

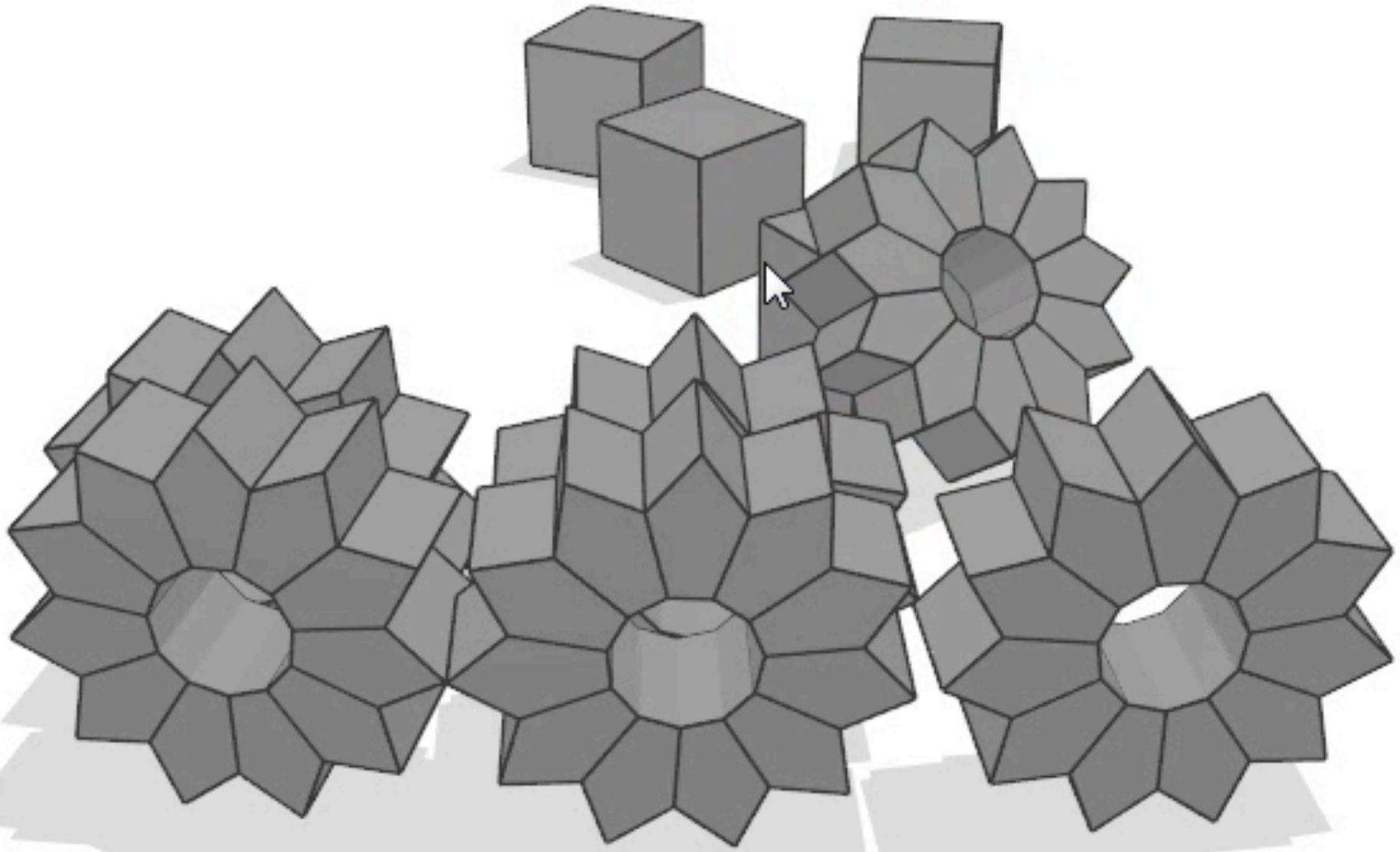
- Volume based approaches: Computing the actual volume intersection (LDI's using GPUs), signed distance fields

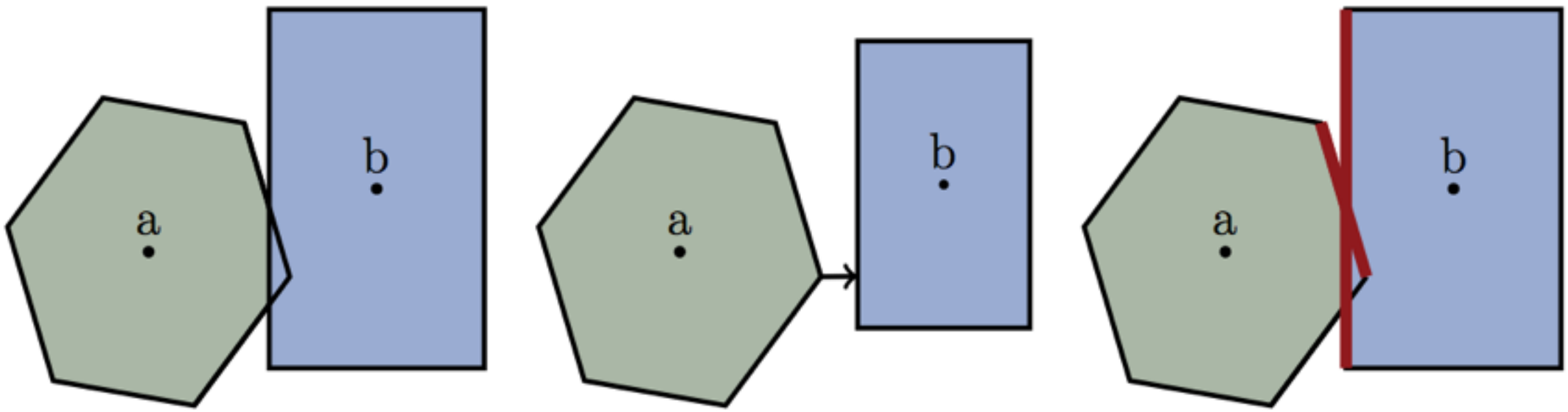
HOW DO THEY "WORK"?

- They use a notion of "soft/fuzzy" geometry to deal with numerical precision and round off problems
- Many assumes a state of interpenetrating objects
- They need to be computationally fast so they are often based on local information
- Local information is not always well-posed so heuristic assumptions is often needed

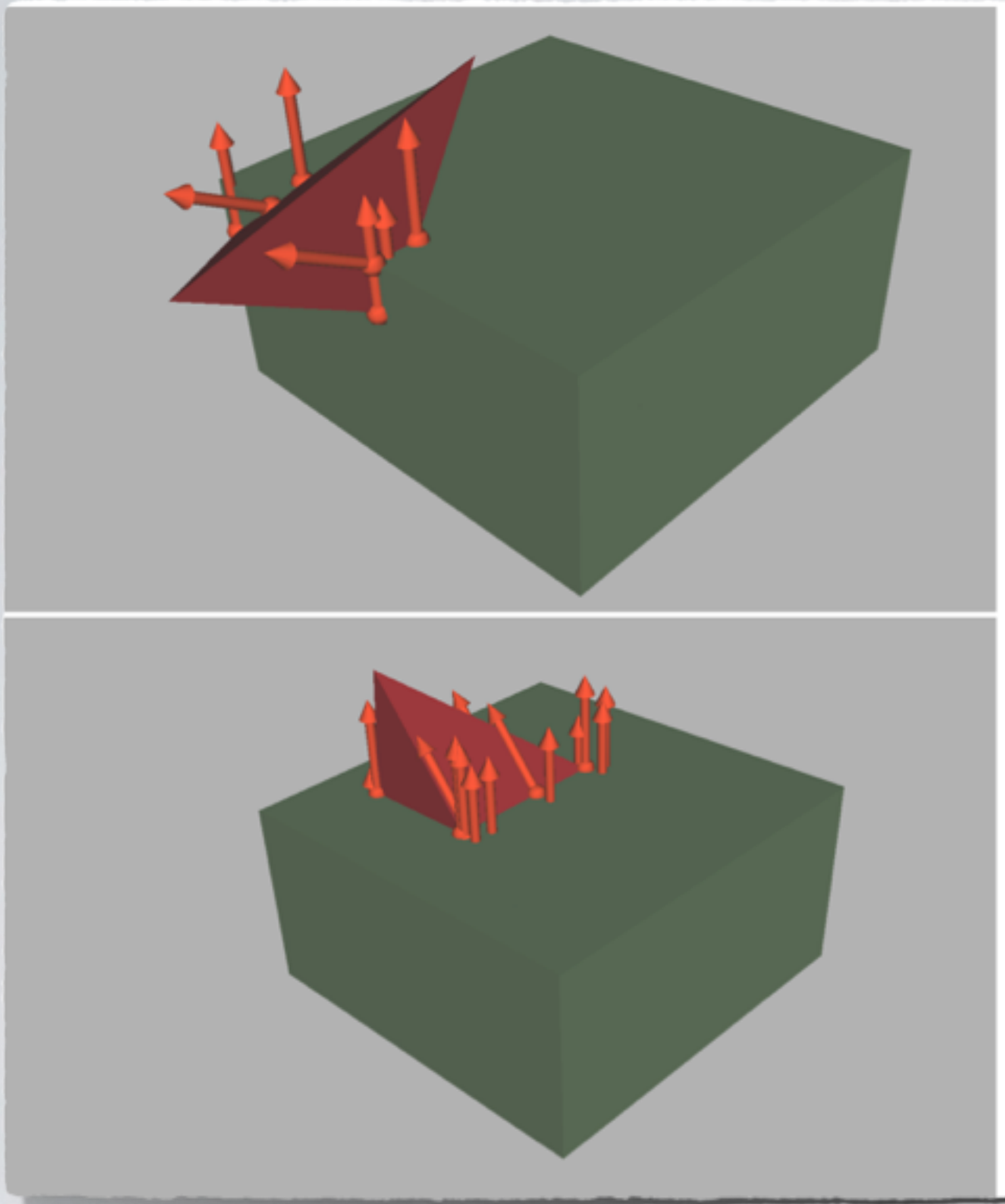
Interactive test: Box stacking. User must build a stack and tilt it by moving the topmost box.

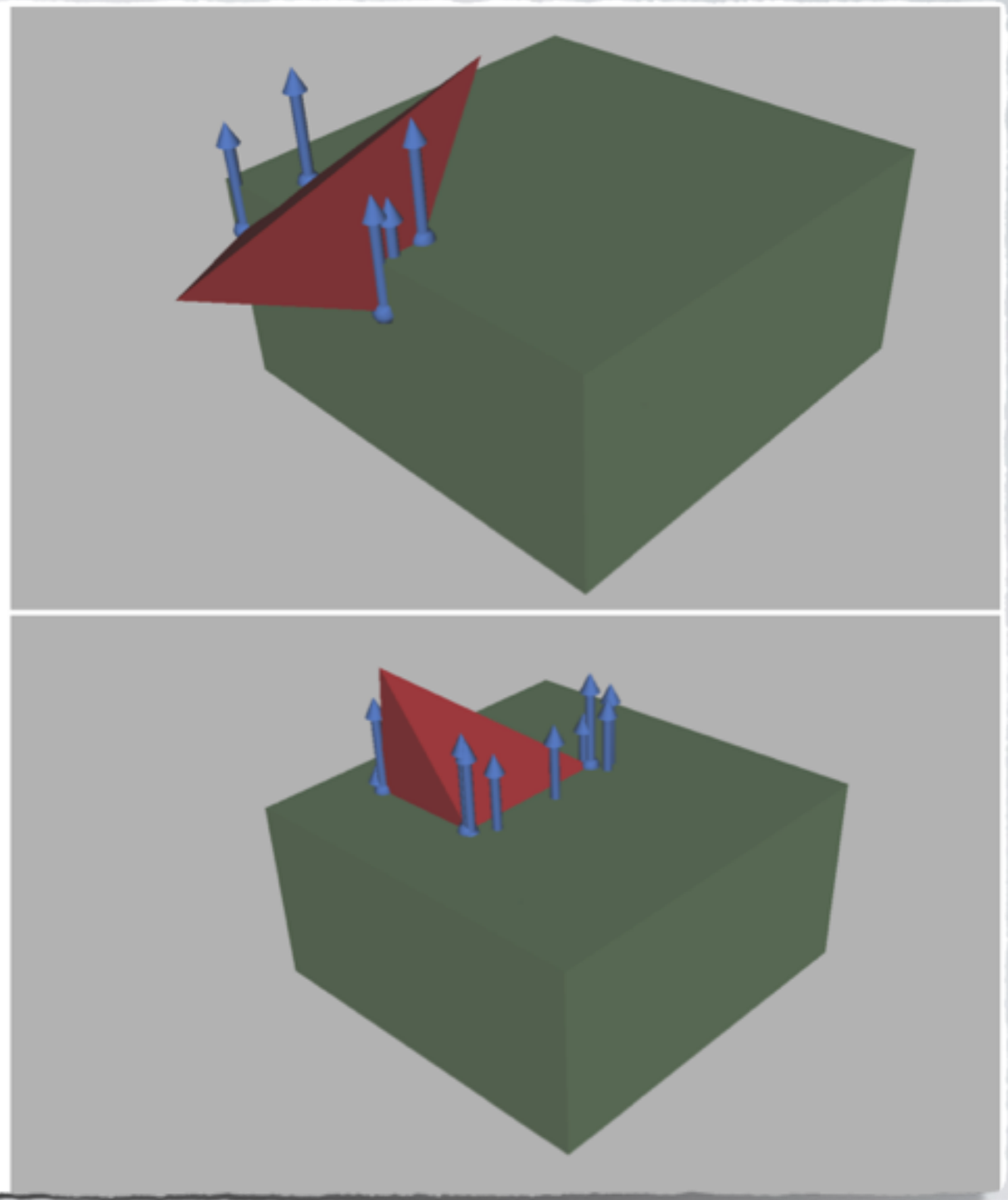


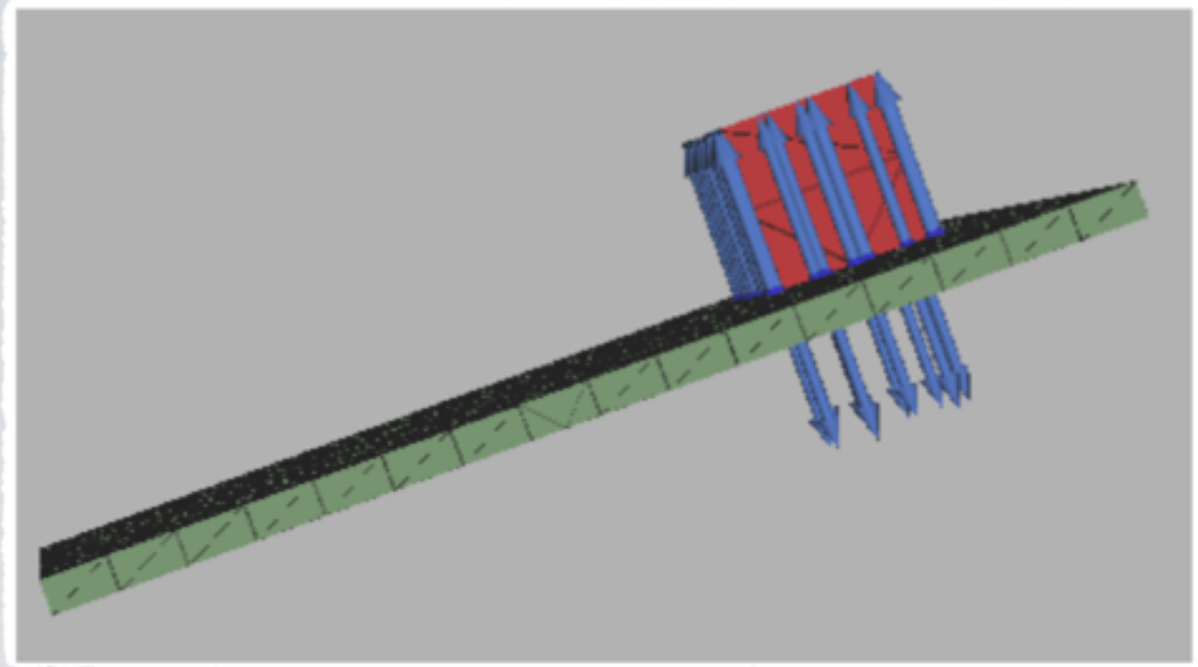


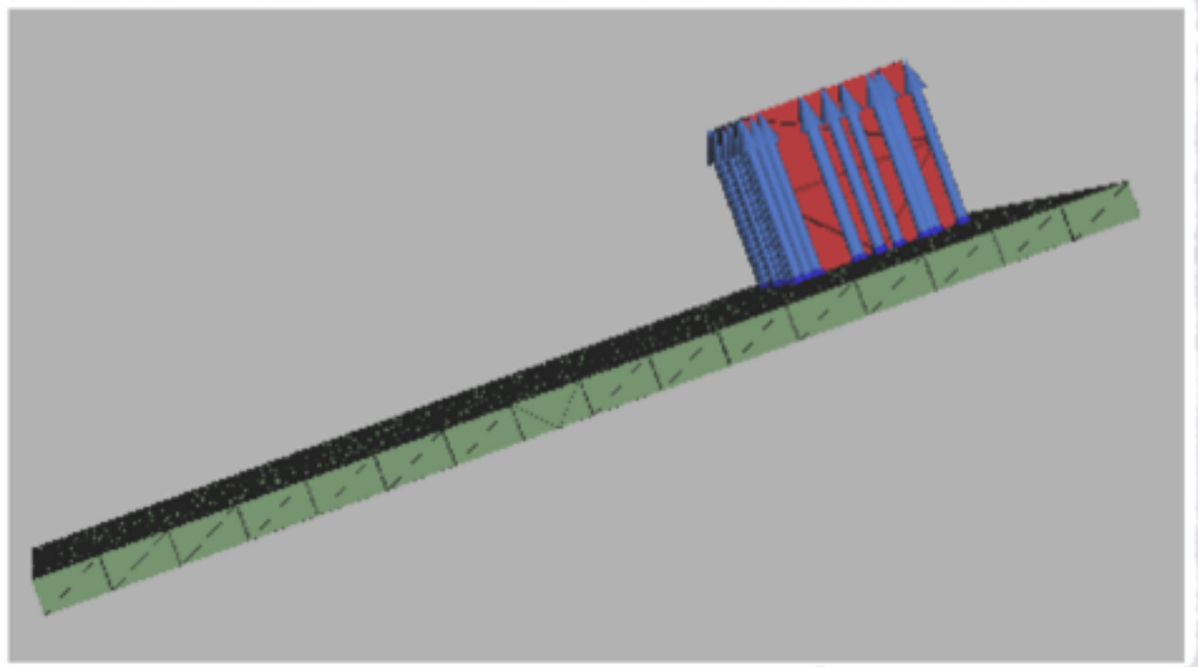


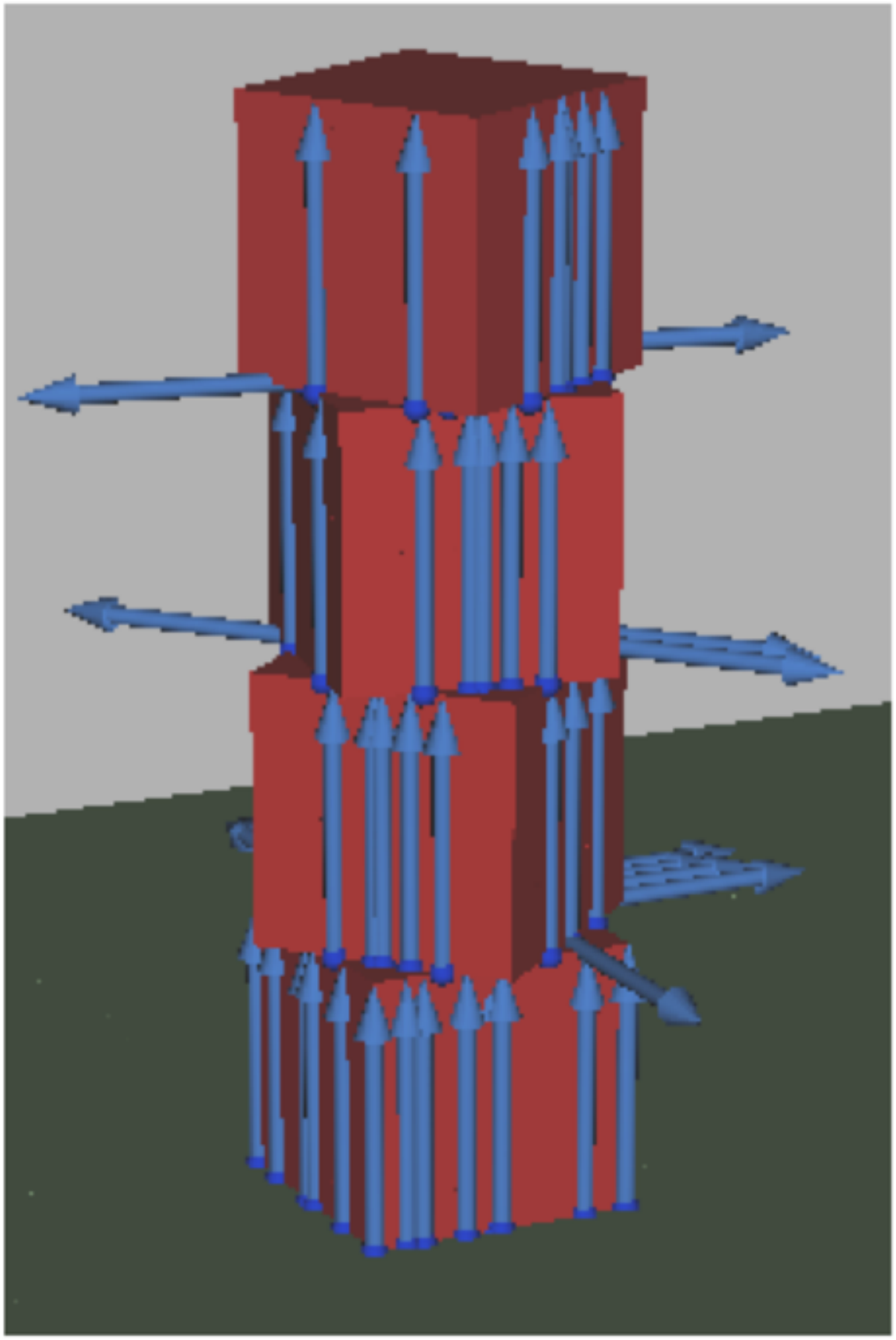
(a) (b) (c)

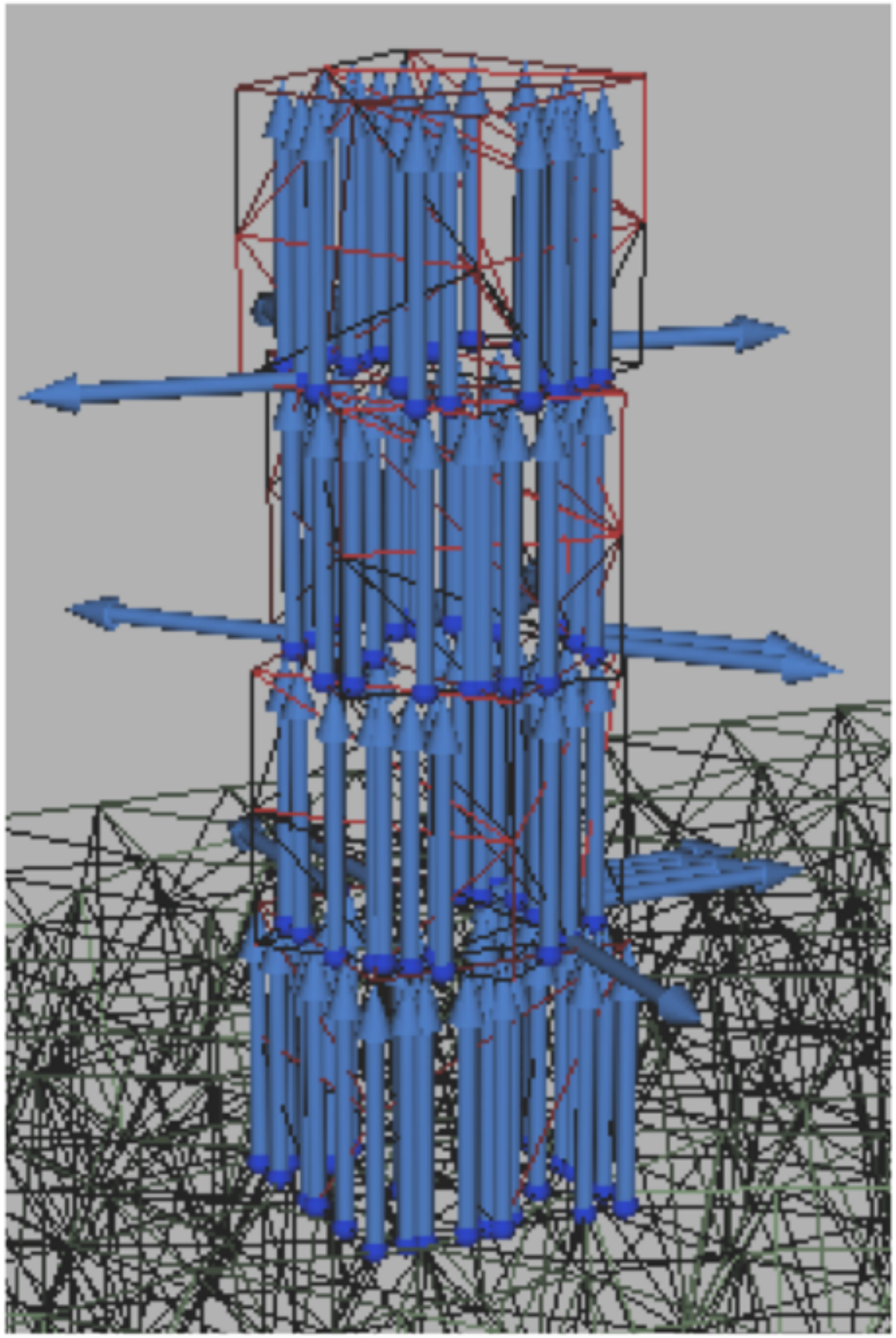


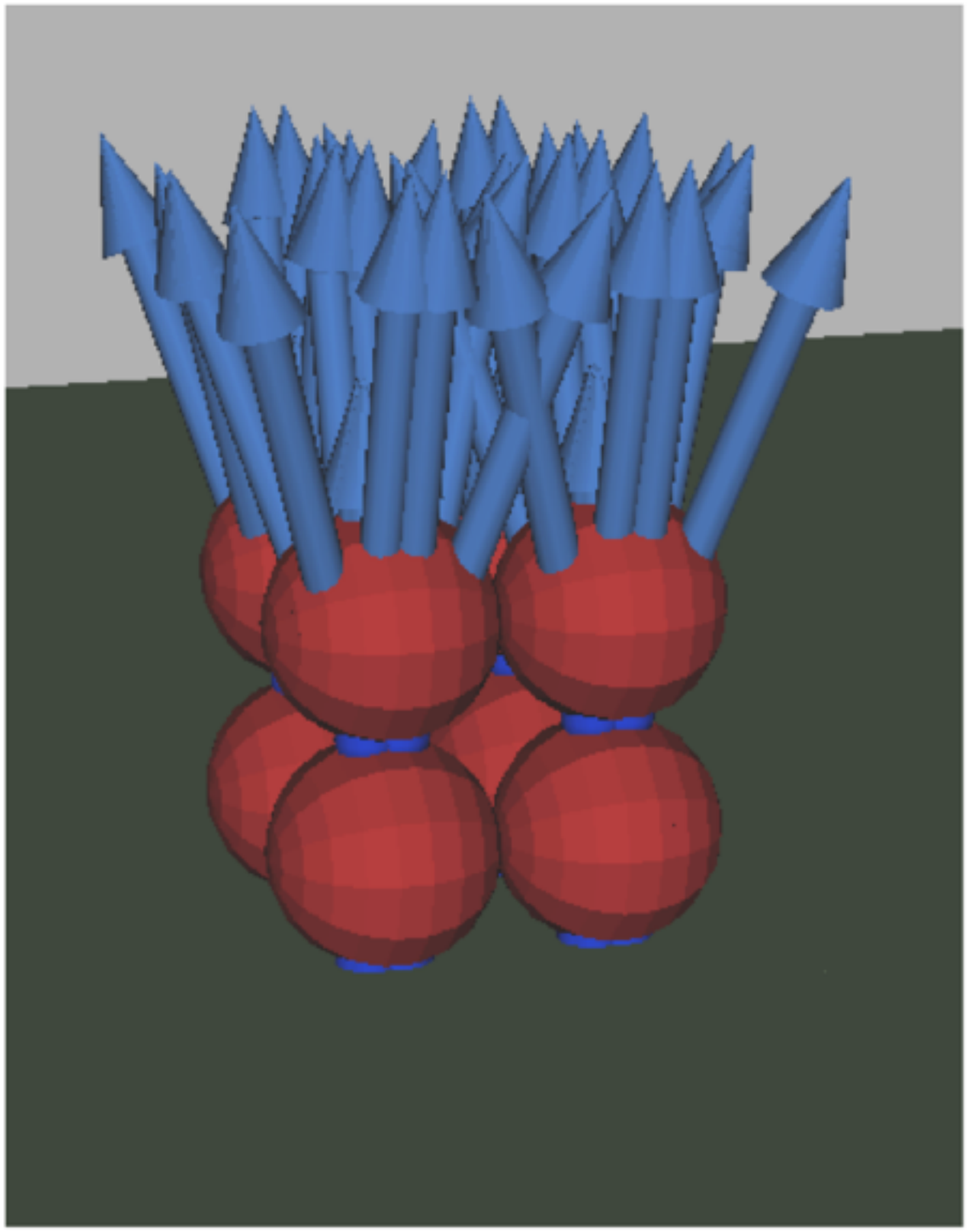


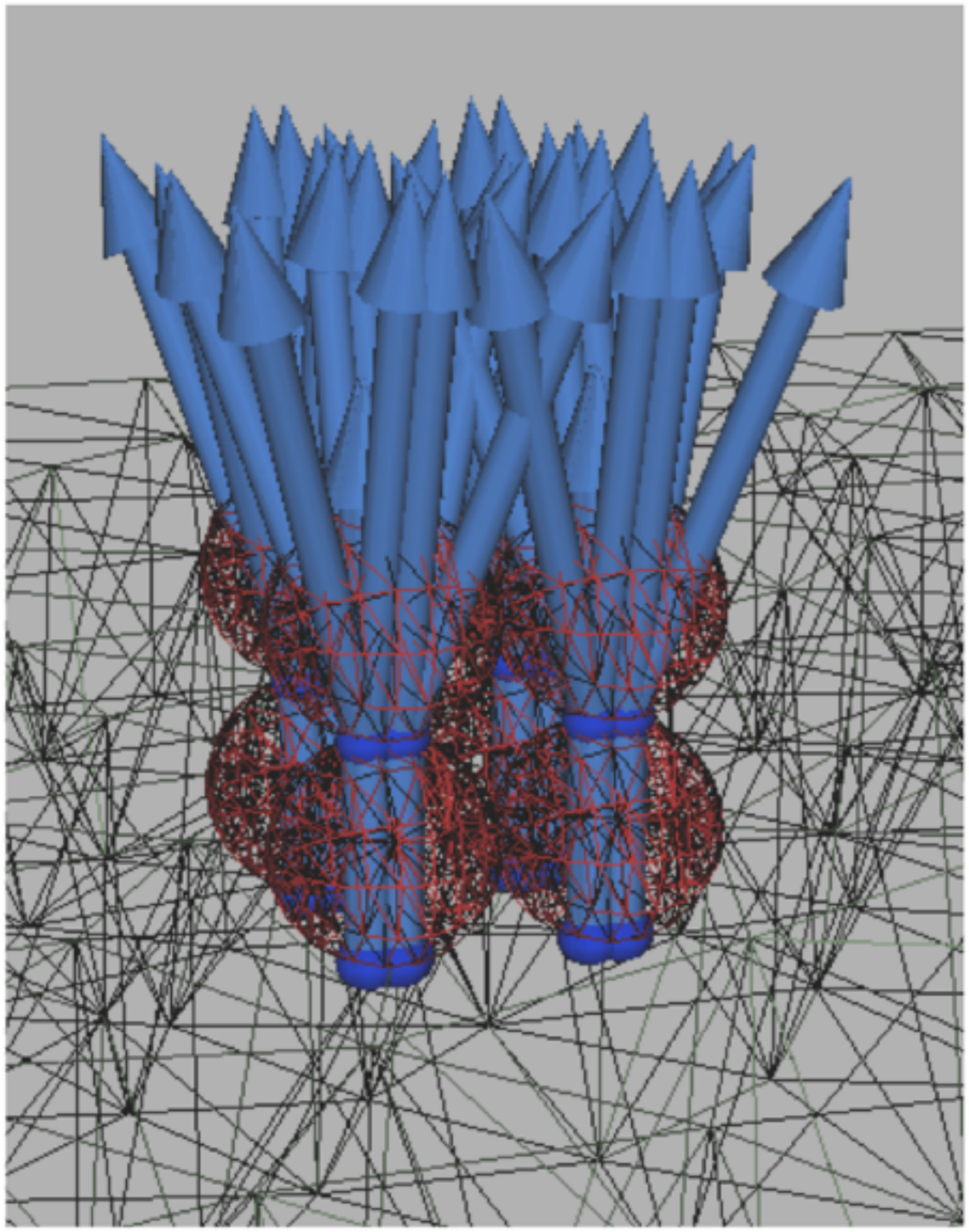


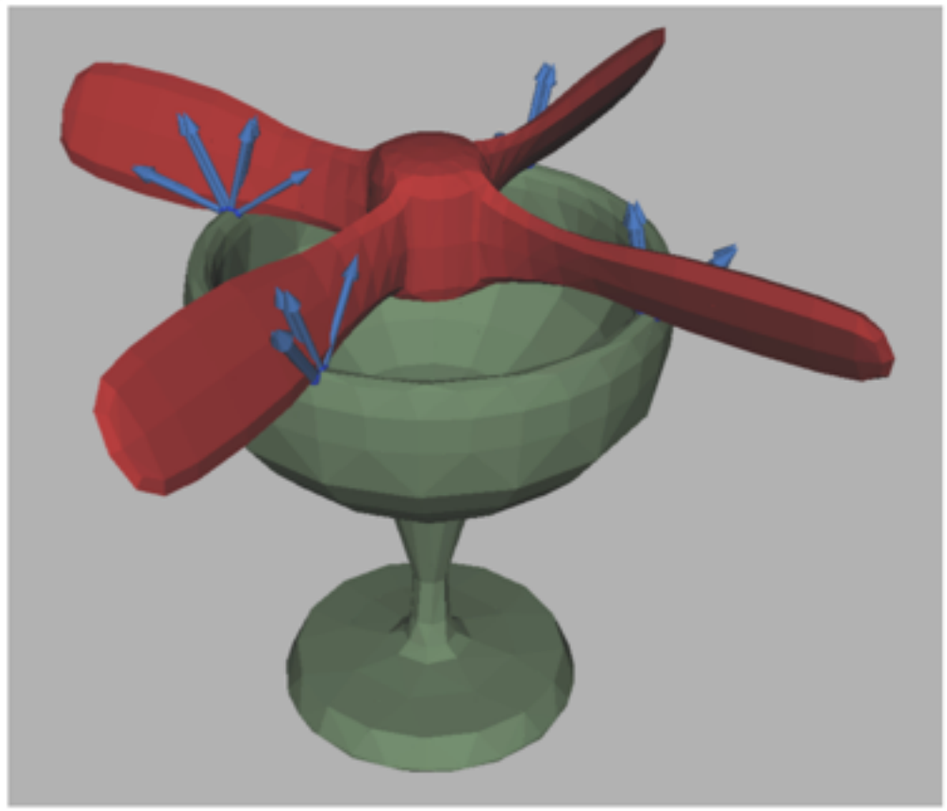


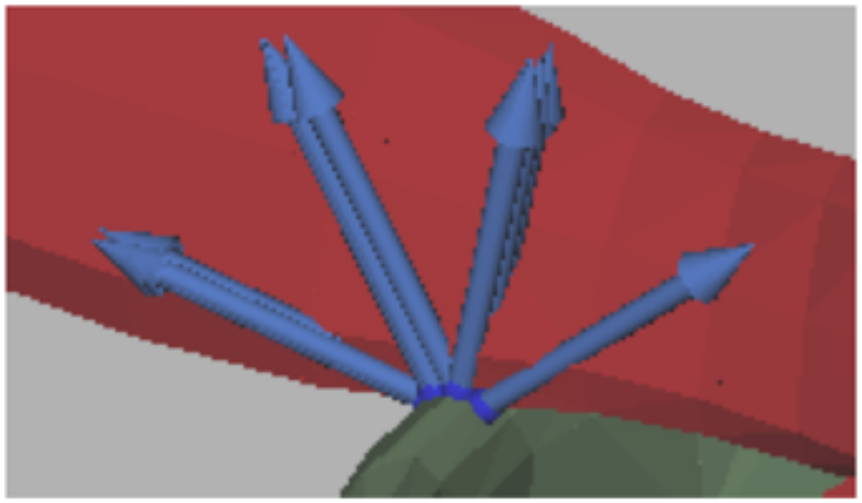


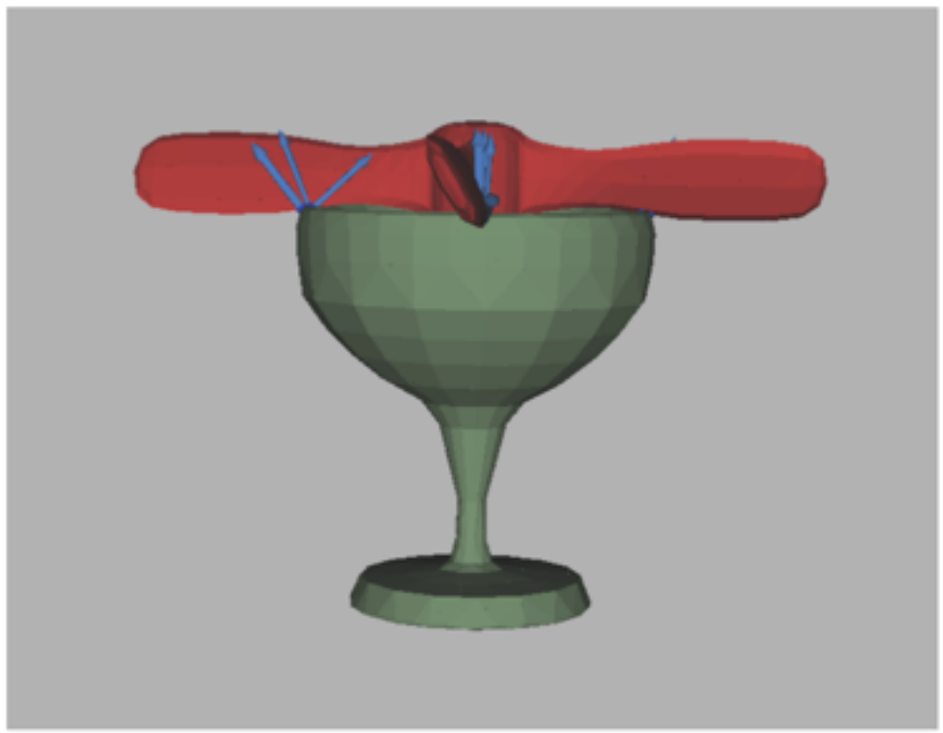


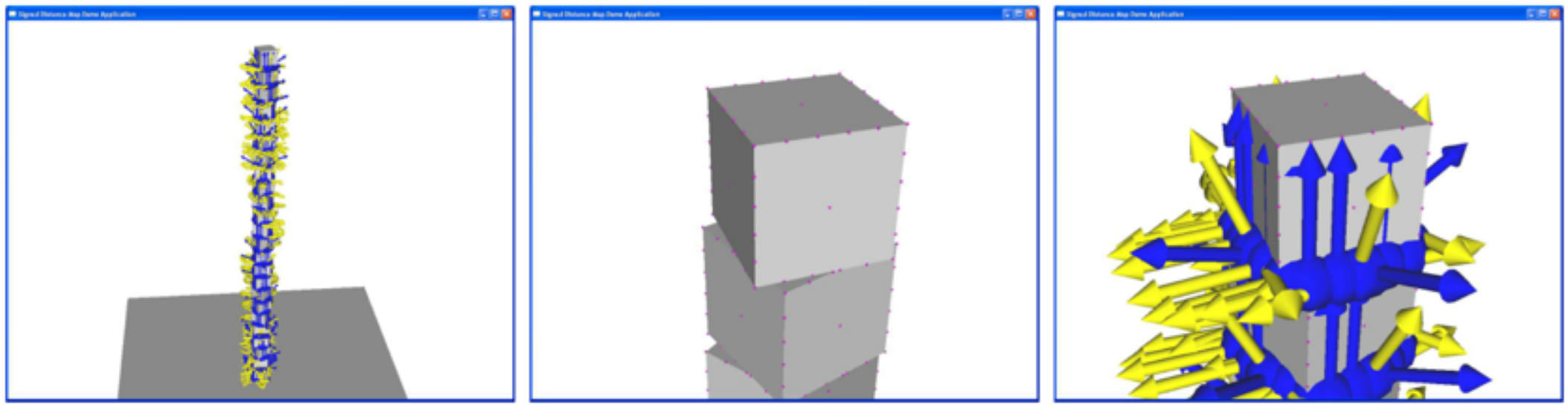




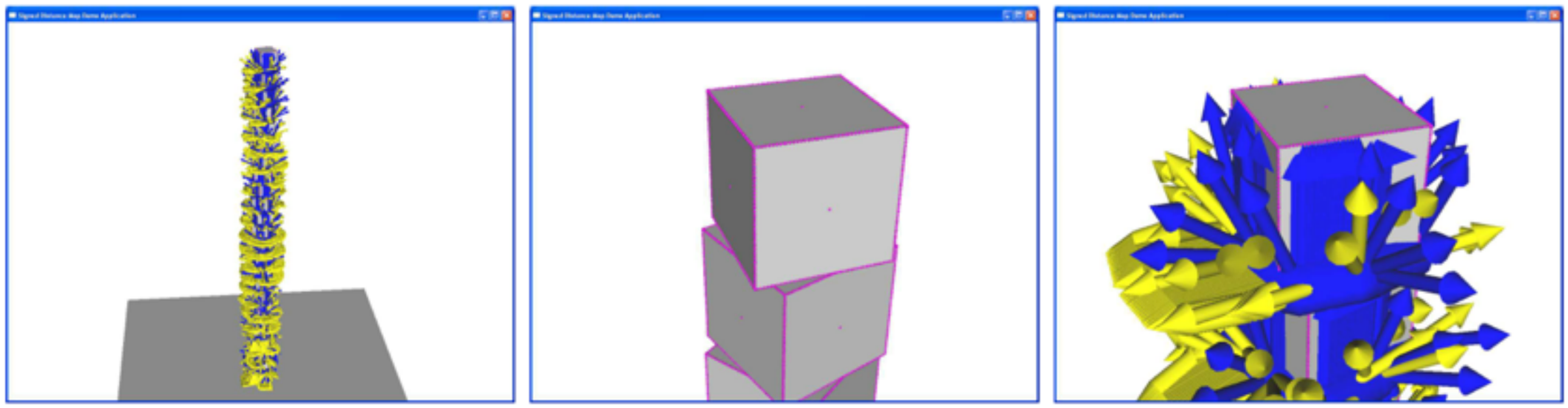


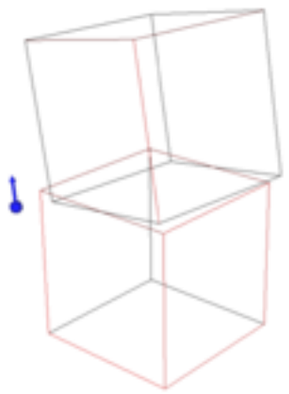




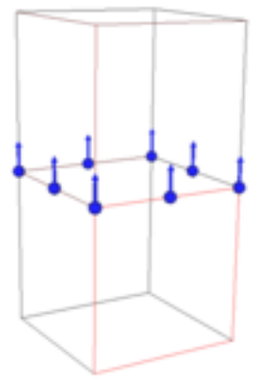


(a) Sensible sampling.

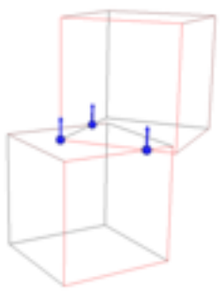


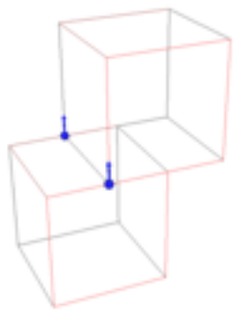


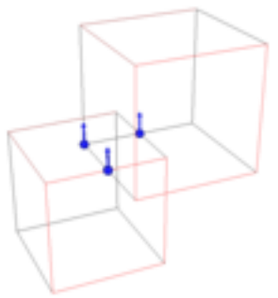
(a) Edge-edge case is picked instead of face-face case, generating a faulty contact point outside the contact region.



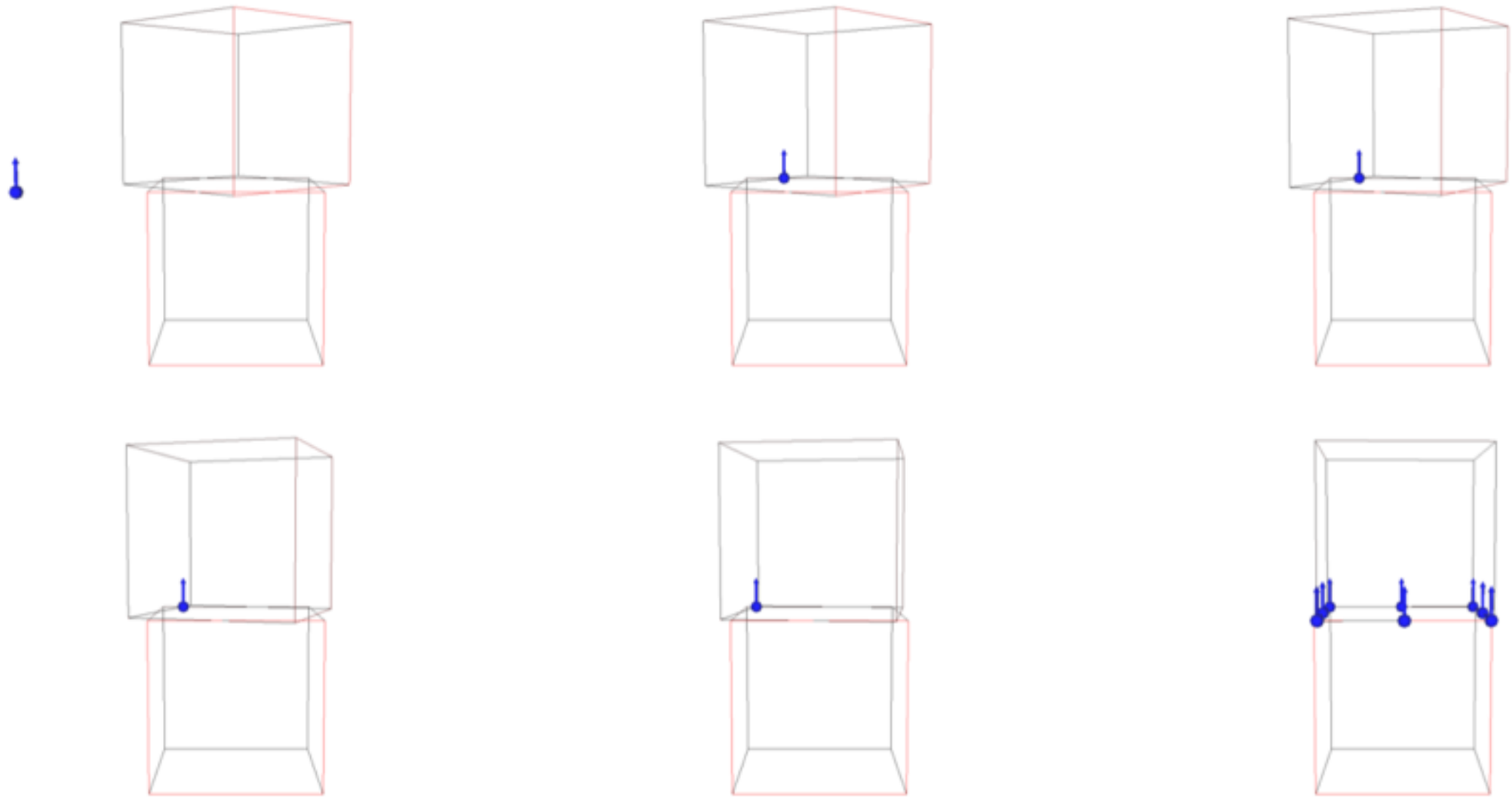
(b) Contact state of corners are ignored during 2D intersection testing, causing edge-edge crossings to generate contact points.

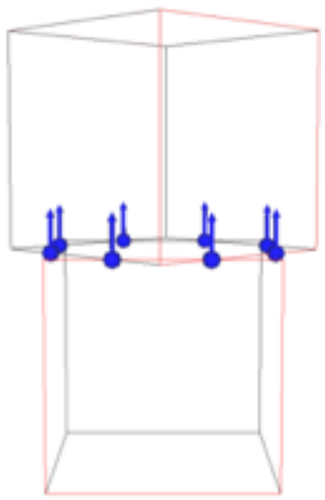
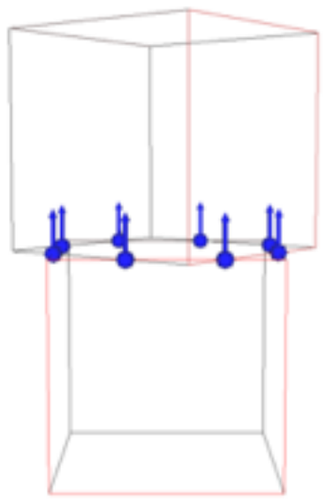
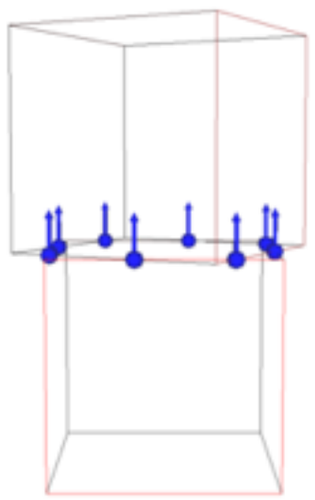
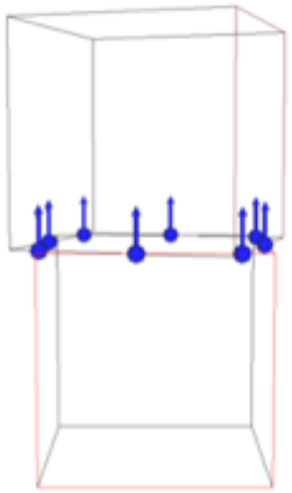
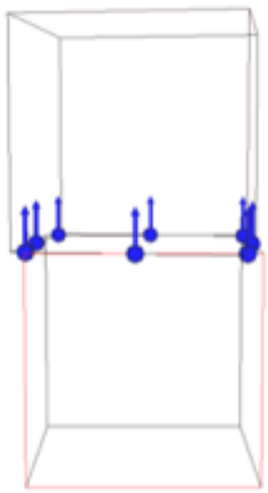
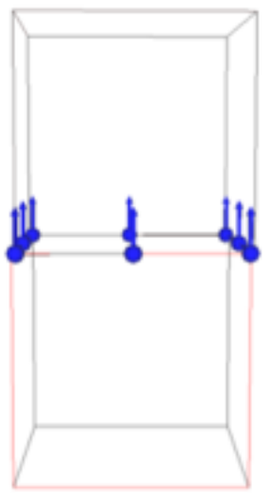






(c) Missing corners, due to badly (or too lazy) 2D intersection testing.



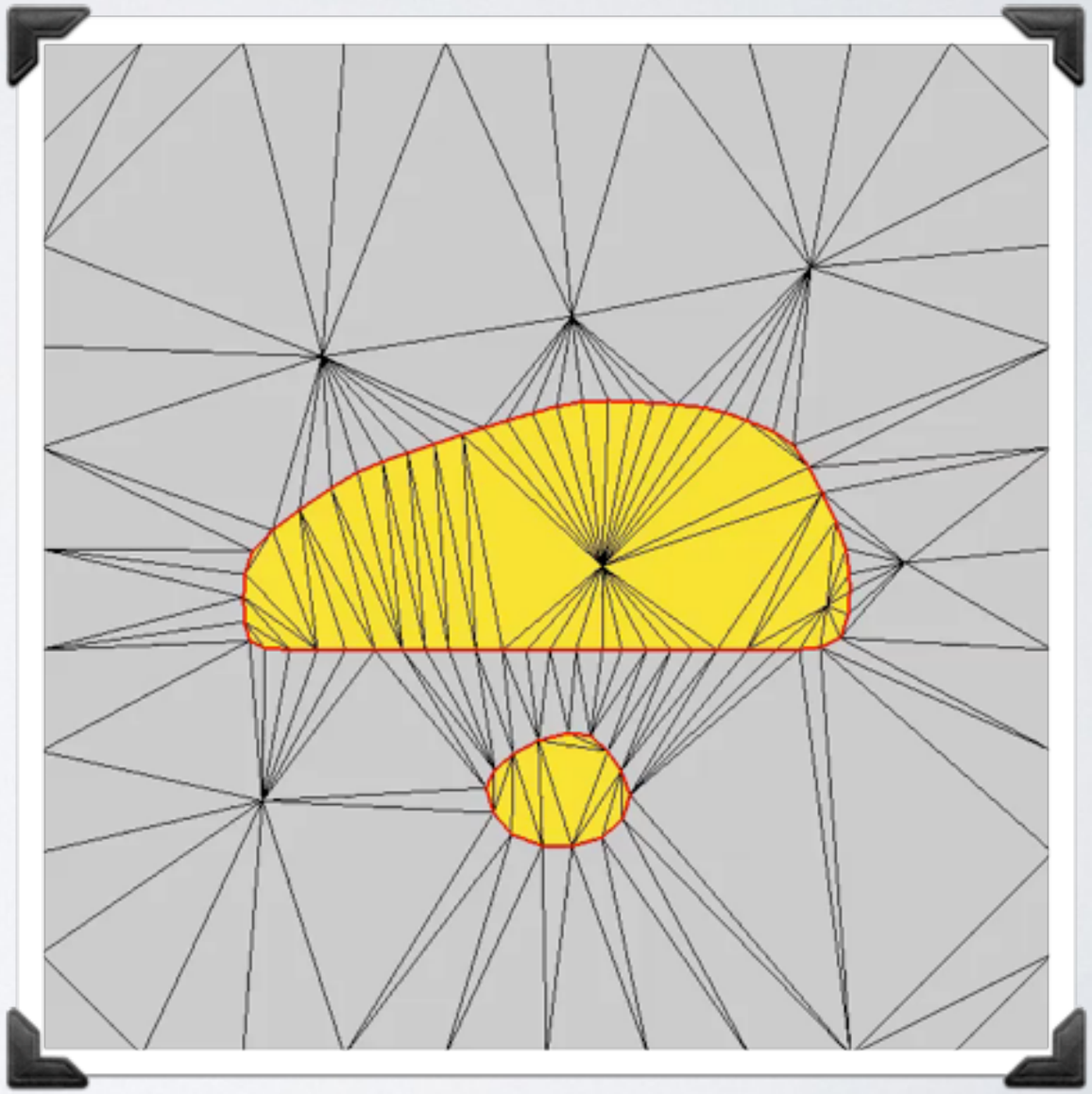







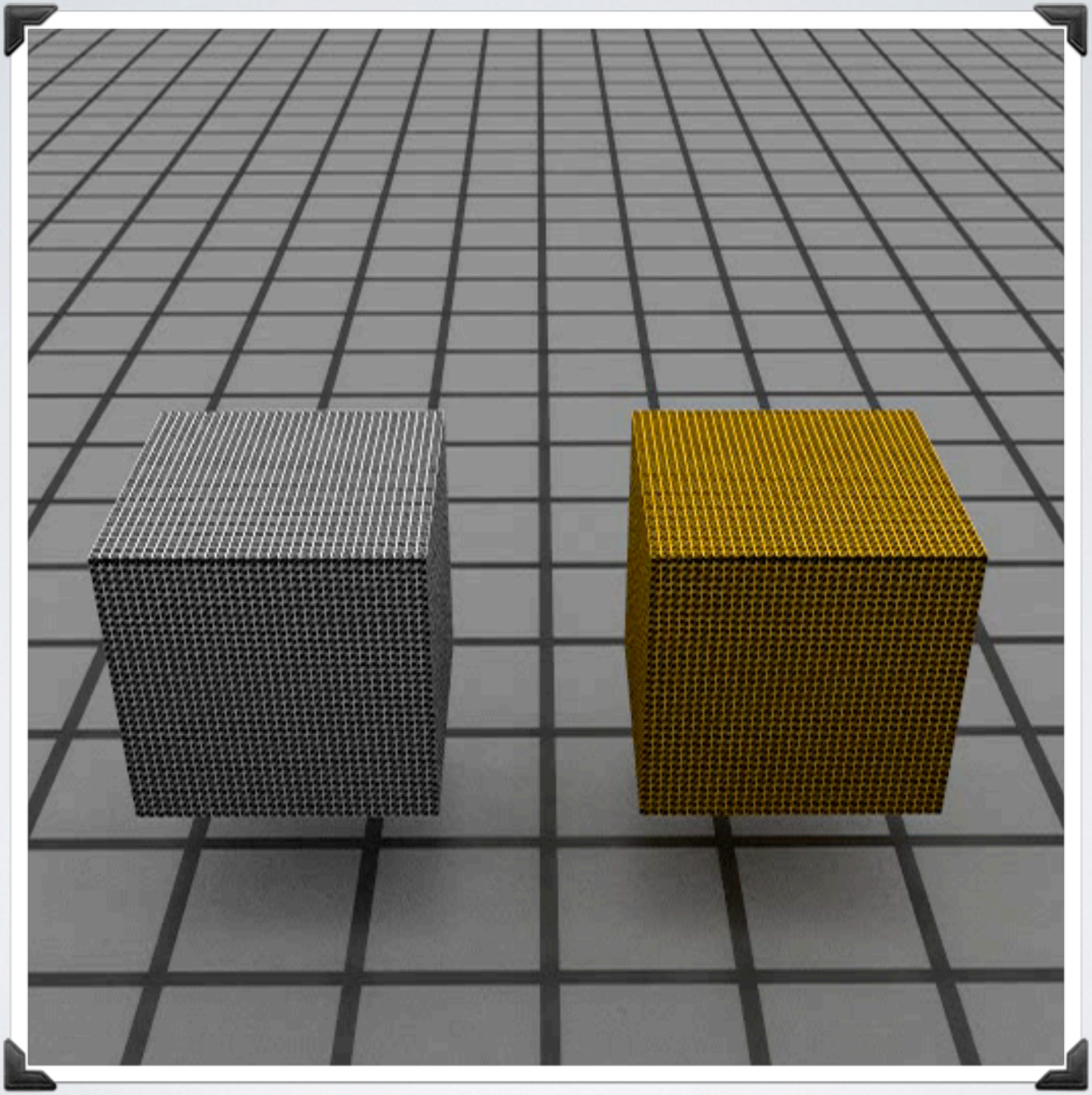
TRICKY TO MAKE A FORMAL DEFINITION?

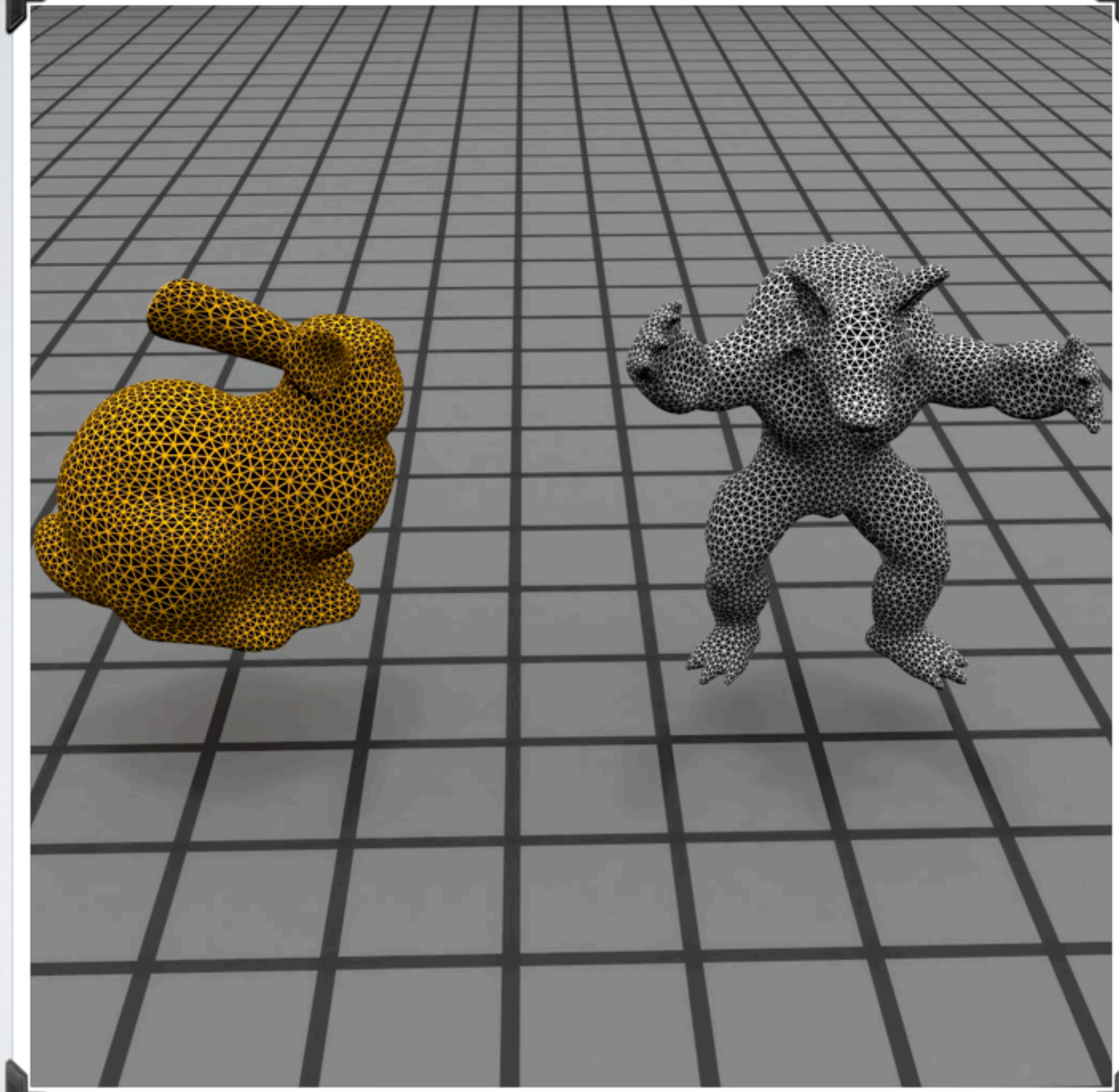
- Discrete time-stepping artefacts results in
  - Penetrating geometry which is unphysical
    - How do we define normals?
    - How do we define proximity measures?
  - Or tunnelling geometry which is funny physics
- We need robustness and small sensitivity to small changes in time and space

WHAT IS A MOVING MESH?

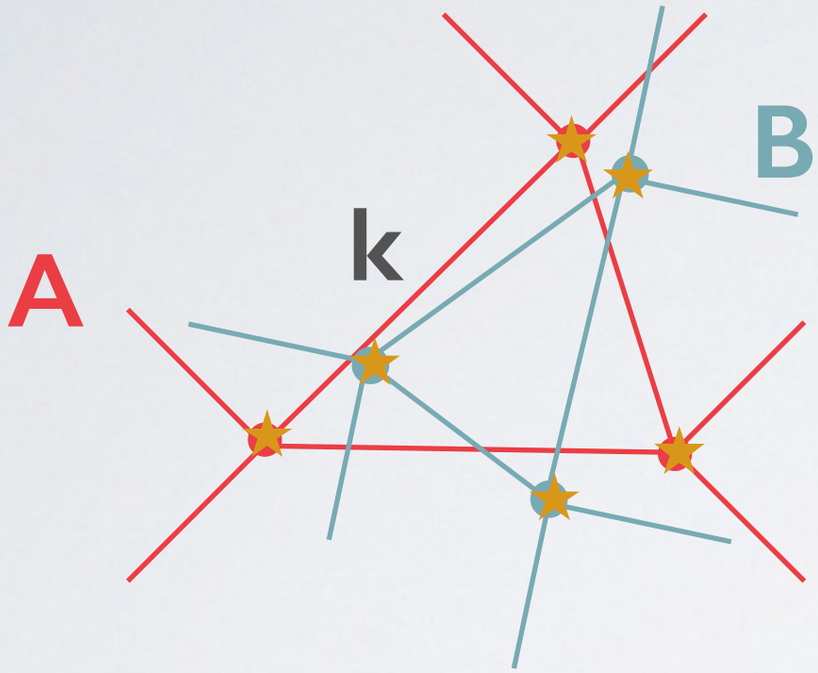
- Handle boundary conditions that dynamically change shape and topology over time
- Handle disjoint domains that change or move over time
- Providing a (optimal) computational mesh at any time
- Minimal numerical dissipation due to Lagrangian meshing property

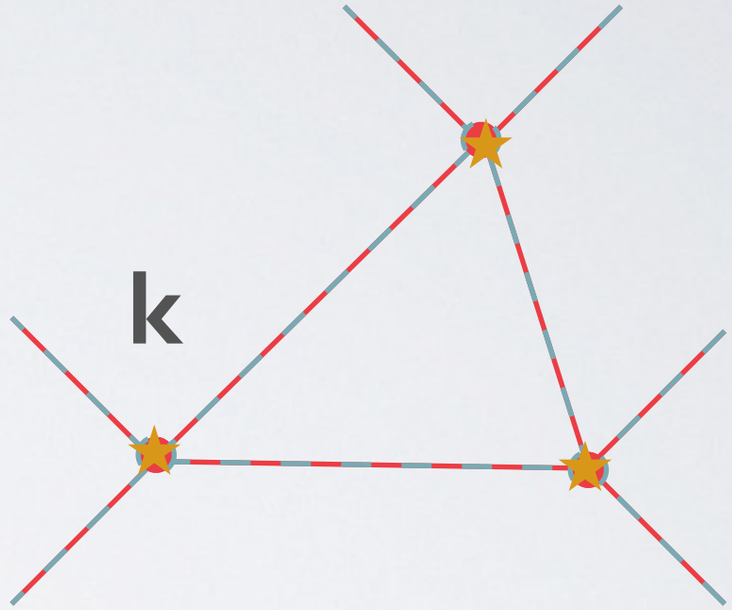






CONFORMING MANIFOLD



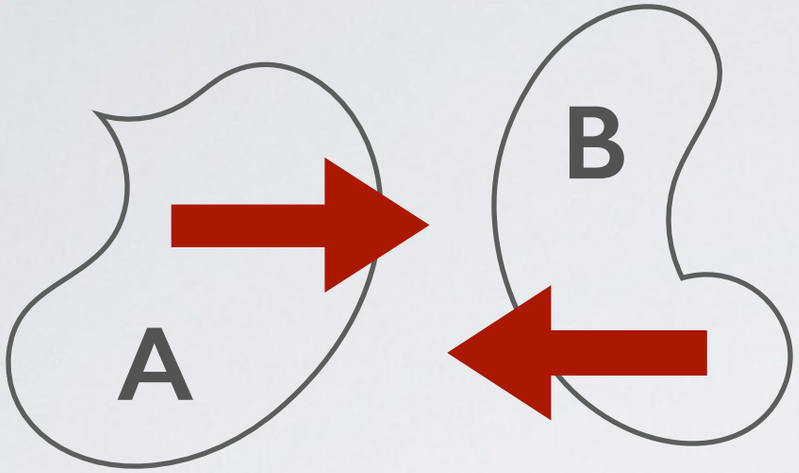


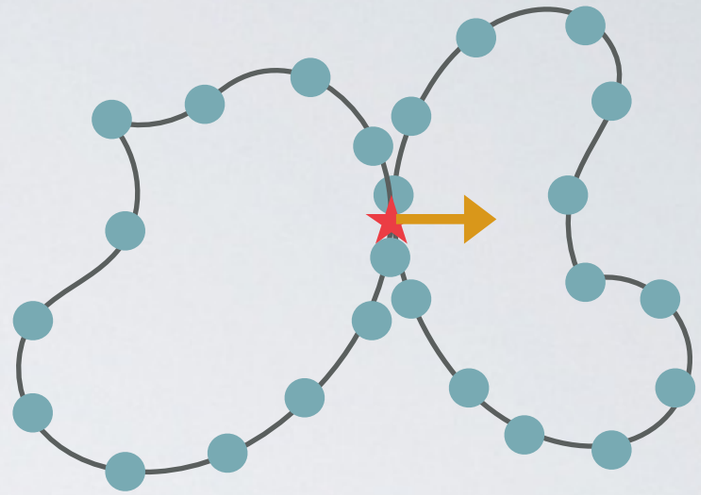
$$\omega_a \vec{f}^{A}_{k,a} + \omega_b \vec{f}^{A}_{k,b} + \omega_c \vec{f}^{A}_{k,c} = -\vec{f}^{B}_{k}$$

$$0 \leq \omega_a, \omega_b, \omega_c \geq 1$$

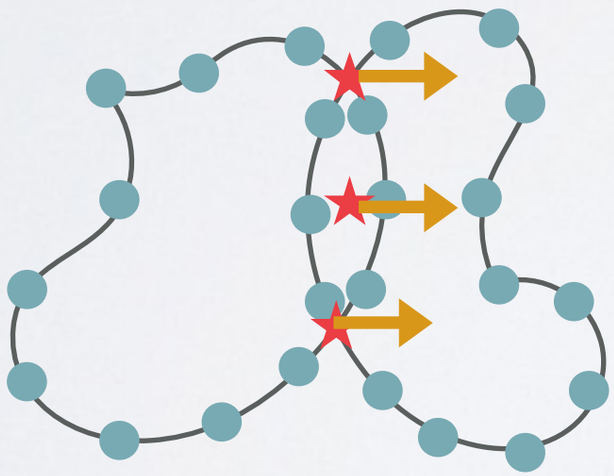
$$\omega_a + \omega_b + \omega_c = 1$$

$$\vec{f}^{A}_{k} = -\vec{f}^{B}_{k}$$

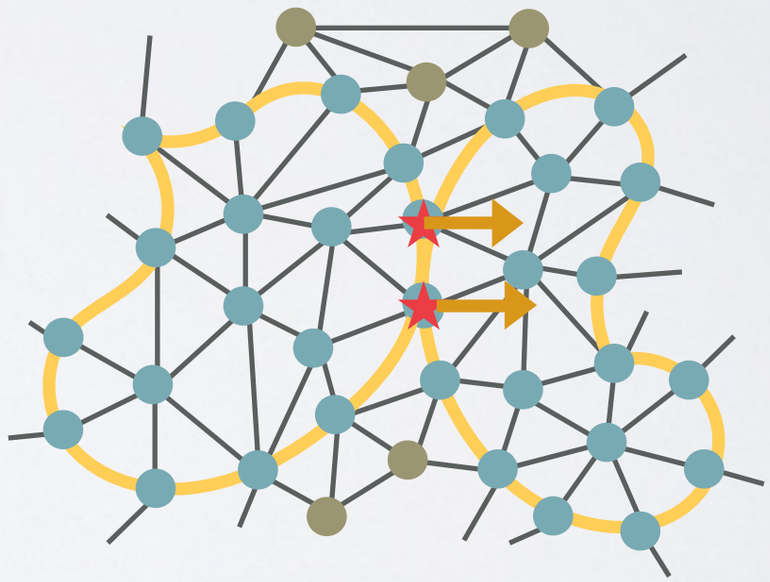




Continuous
Collision Detection



Discrete
Collision Detection



Moving Meshes

	DCD	CCD	MOM
Exact geometry	No	Yes	Yes
Normals	No	Yes	Yes
Penetration	Yes	No	No
Tunneling	Yes	No	No
Conforming	No	No	Yes
Quality Meshing	No	No	Yes

A FORMAL CONTINUOUS DEFINITION?

$$\mathbf{A}, \mathbf{B} \subseteq \mathbb{R}^3$$

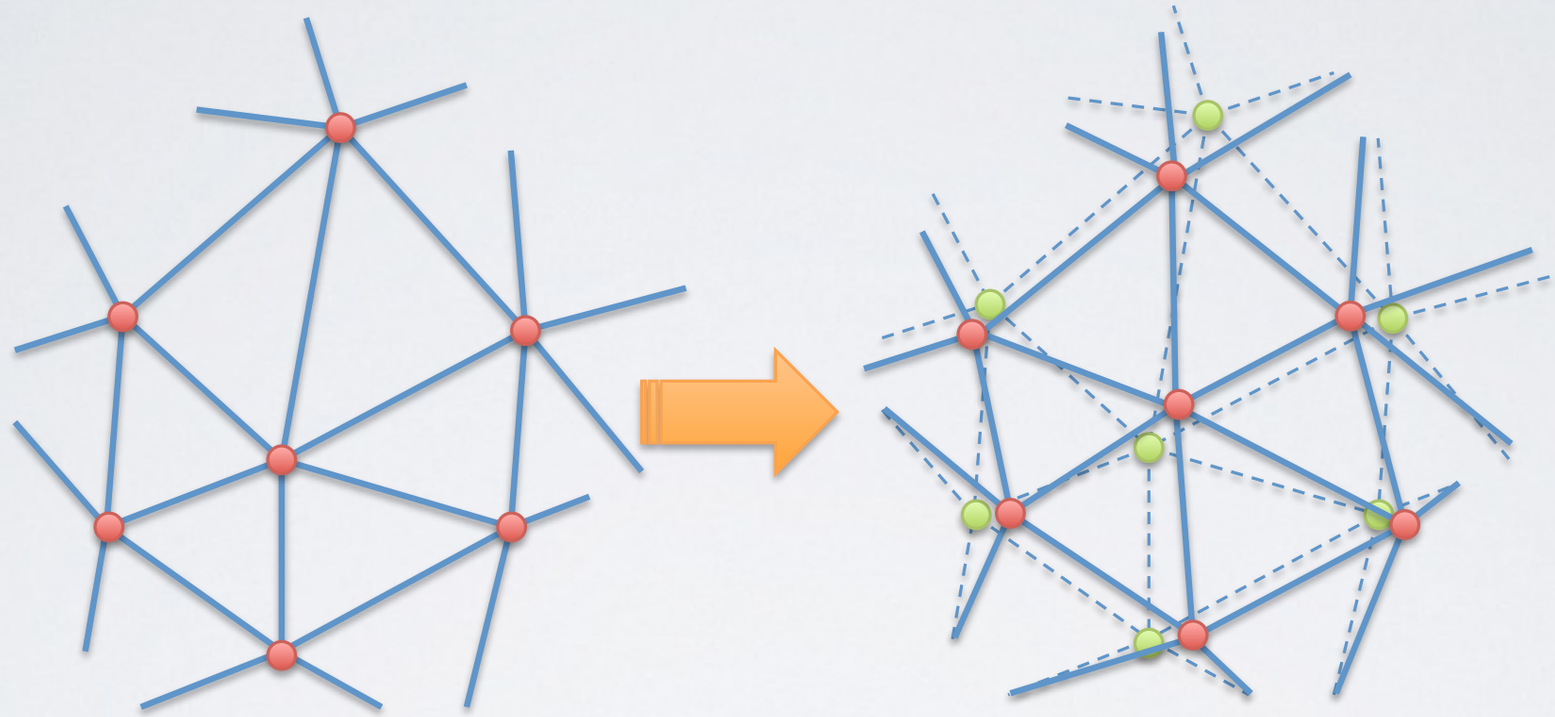
$$\mathbf{A} \cap \mathbf{B} = \mathbf{C}$$

$$\vec{p} \in \mathbf{C}$$

$$\vec{n}(\vec{p}) \in \mathcal{N}_{\mathbf{A}}(\vec{p})$$

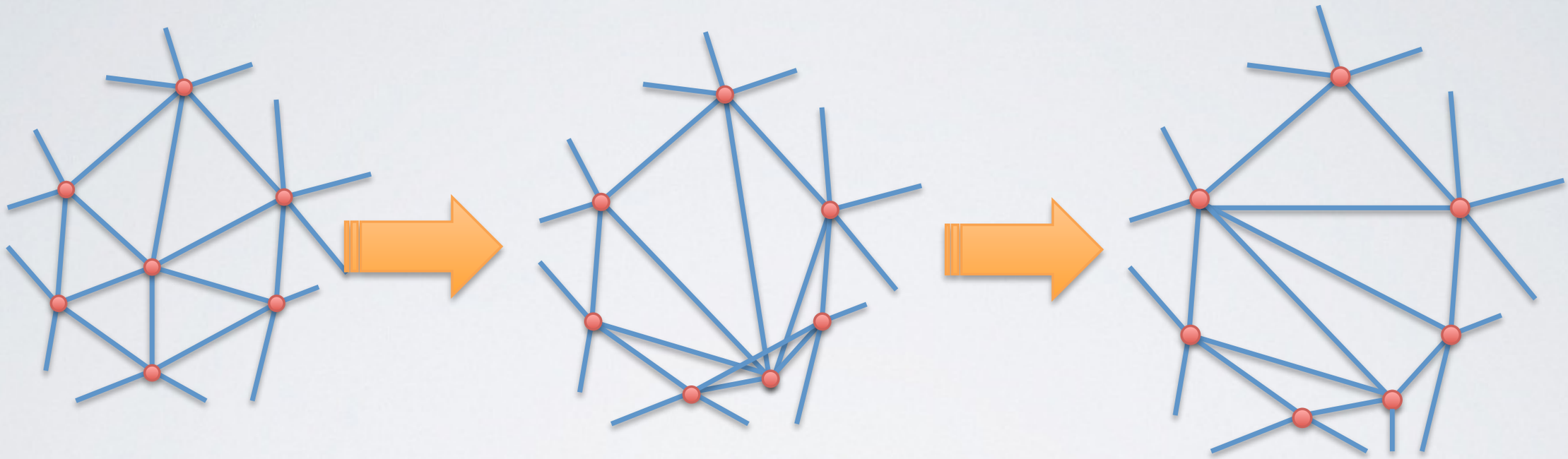
$$-\vec{n}(\vec{p}) \in \mathcal{N}_{\mathbf{B}}(\vec{p})$$

MOVING THE MESH



- Move all vertices
- Keep values stored in vertices "fixed"

MOVING THE MESH



- Problem: Displacements can cause collapsed or inverted mesh elements
- Insight: Connectivity changes can not be avoided

CONSERVATIVE ADVANCEMENT STRATEGY

Algorithm Advance
  While exist vertex not at target position
    For each vertex do
      move vertex as far as possible without collapsing or
      inverting any elements and keep all other vertices frozen
    Next vertex
    For each nearly collapsed or inverted element do
      Change connectivity locally
    Next element
  End while
End algorithm

CONSERVATIVE ADVANCEMENT STRATEGY

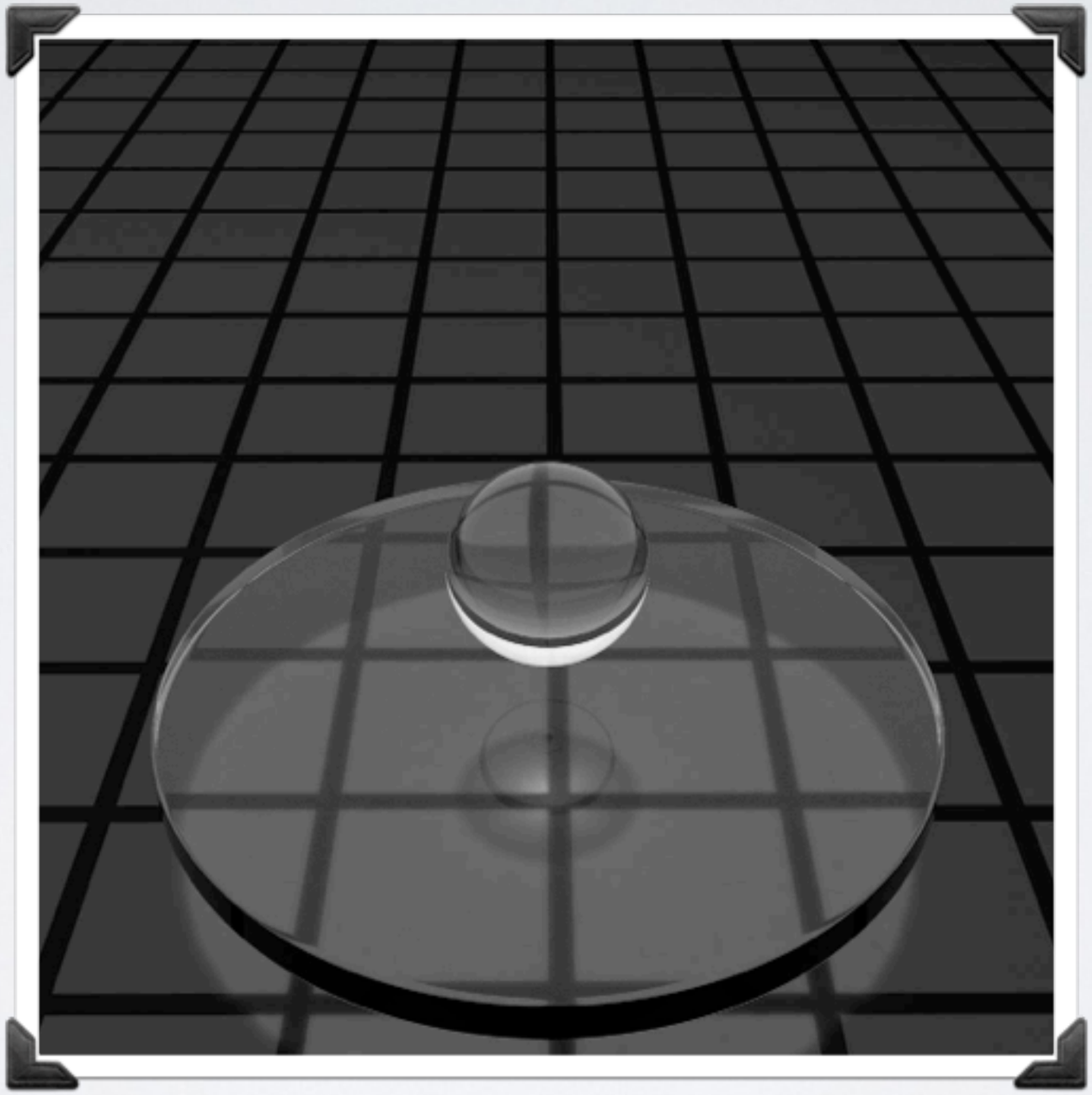
Algorithm Advance
  While exist vertex not at target position
    For each vertex do
      move vertex as far as possible without collapsing or inverting any elements and keep all other vertices frozen
    Next vertex
    For each nearly collapsed or inverted element do
      Change connectivity locally
    Next element
  End while
End algorithm

Sequential

and not easy to parallelize

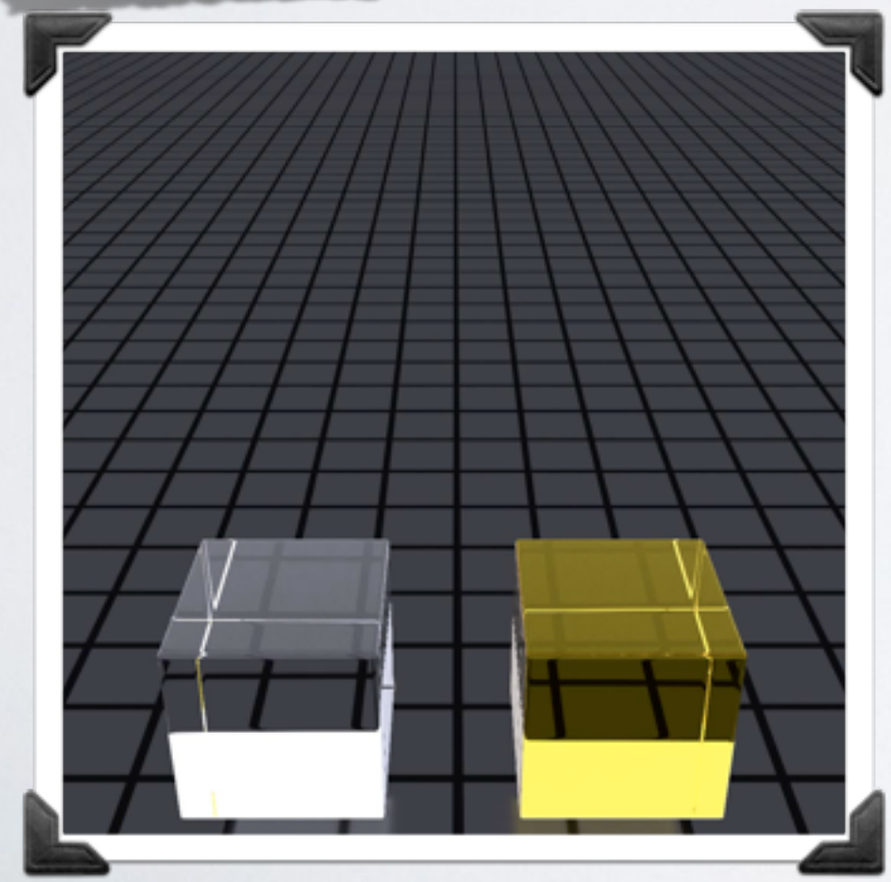
Vertex order dependency may results in non-symmetry

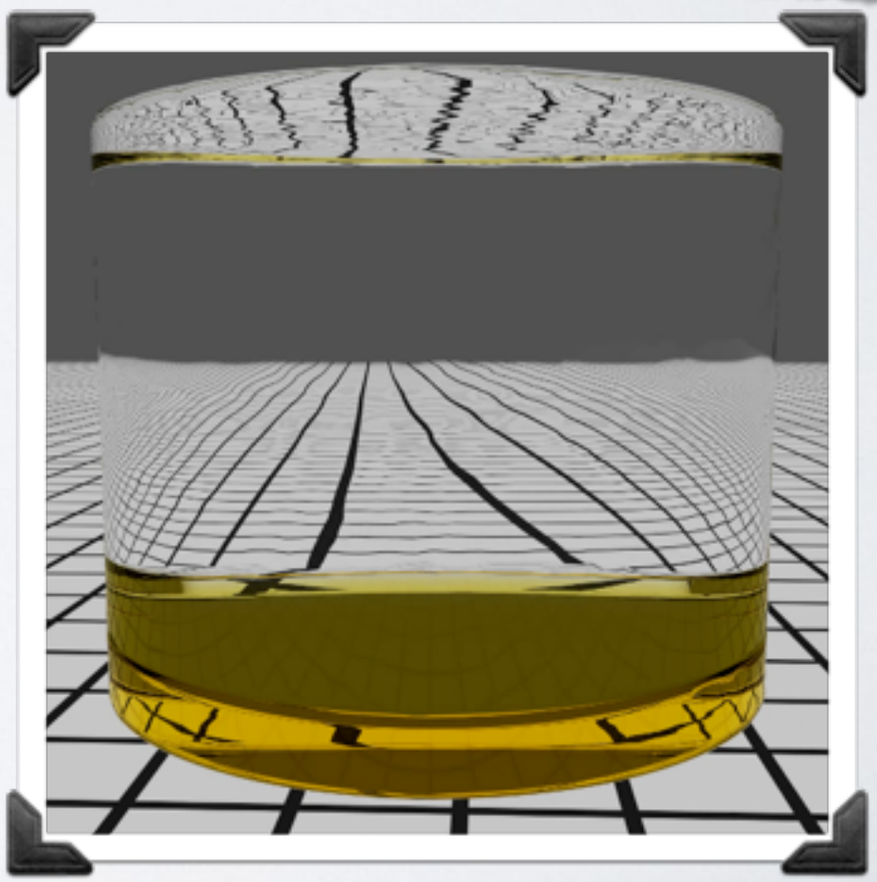
WATER CROWN EXAMPLE



PERFORMANCE RESULTS

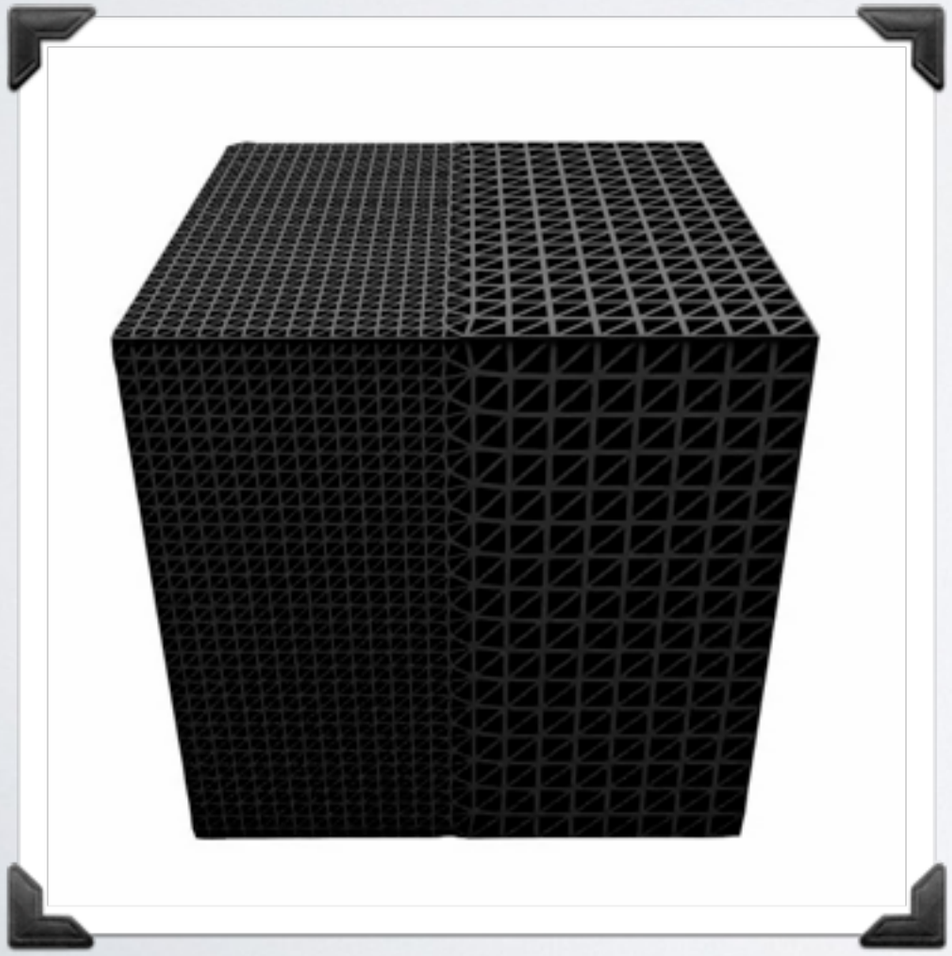
example	#tets initial/max/median	#triangles initial/max/median	average time per iteration (seconds) assembly	GMRES	advection	total
CUBE N.U.	20345 / 20345 / 17403	7616 / 7616 / 7610	6.41	3.48	3.87	13.9
OIL/WATER	42319 / 42319 / 24538	8832 / 9532 / 9040	8.91	6.57	4.19	19.9
TEASER	24770 / 92893 / 43983	11840 / 37089 / 20575	22.0	11.2	14.4	48.1
CROWN	33810 / 94789 / 81983	15104 / 38054 / 32160	32.7	19.6	19.8	72.8
DAM BREAKING	84852 / 108518 / 98722	33370 / 42736 / 38291	40.2	21.4	23.1	85.6

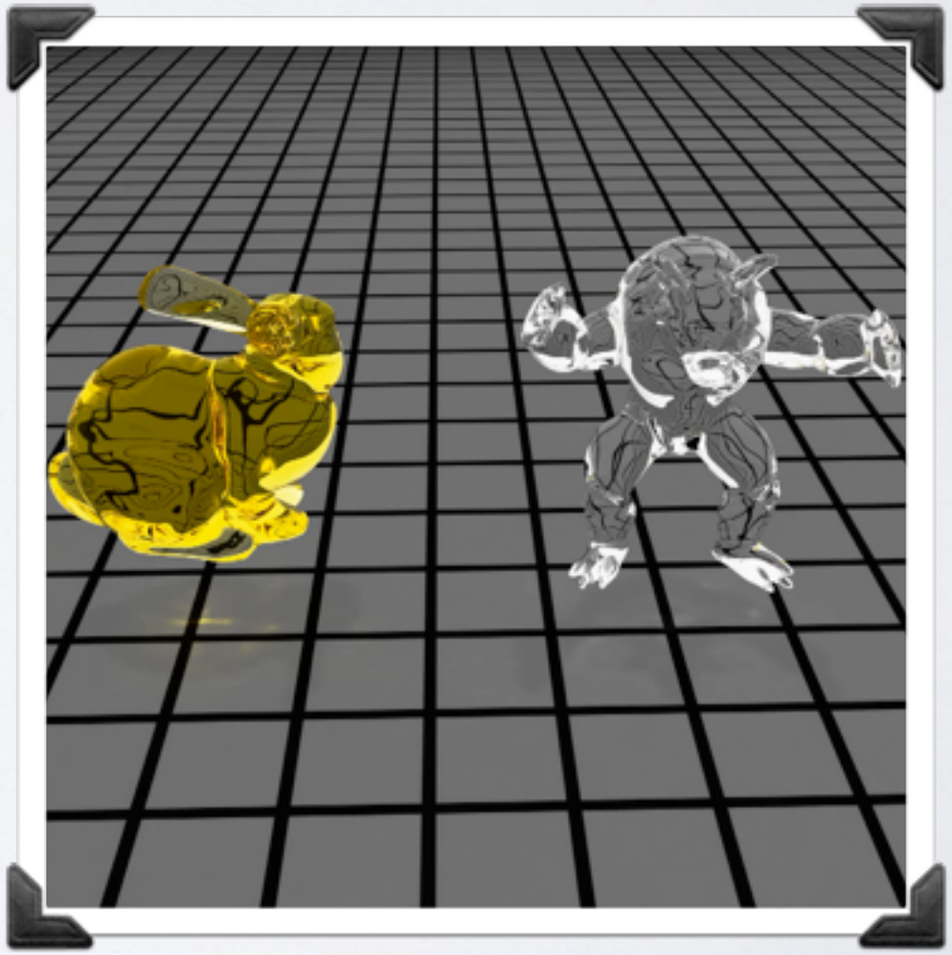




PERFORMANCE RESULTS

example	#tets initial/max/median	#triangles initial/max/median	average time per iteration (seconds) assembly	CR solver	advection	total
CUBE N.U.	20349 / 20349 / 19247	7616 / 7916 / 7715	2.01	3.01	2.32	7.4
BUNNY/ARMADILLO	43252 / 43252 / 40825	18572 / 18527 / 17987	4.56	6.25	5.47	16.5

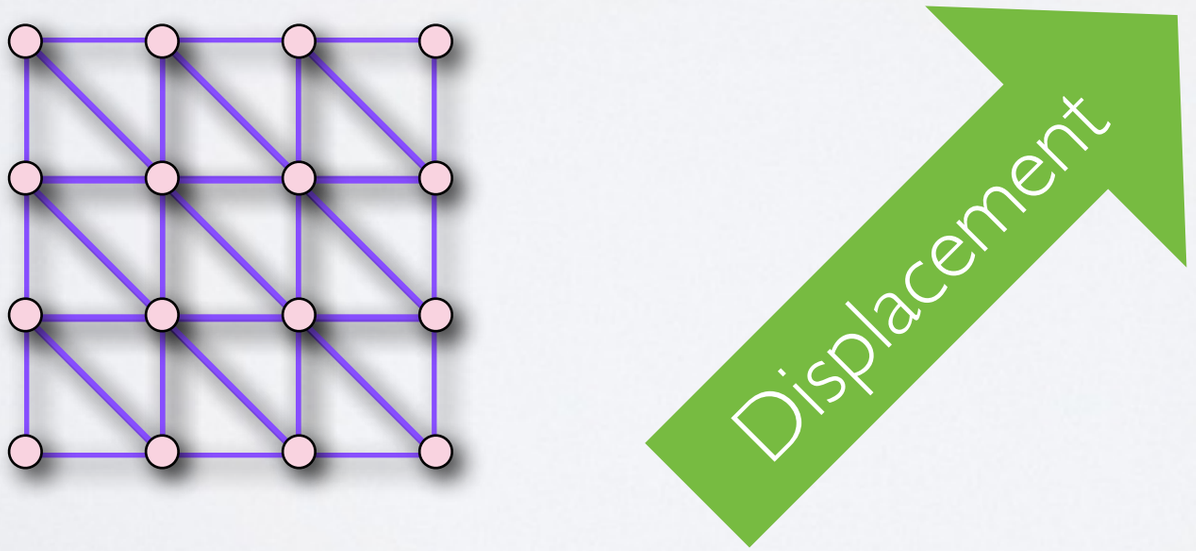




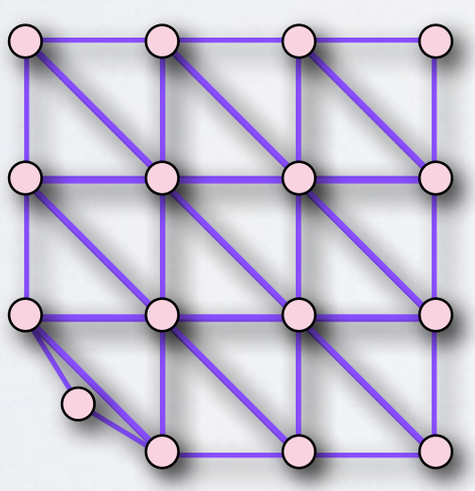
PARTIAL CONCLUSIONS

- The conservative advancement is the computational bottleneck
- The conservative advancement algorithm does not preserve symmetry
- The conservative advancement algorithm can not be easily parallellized

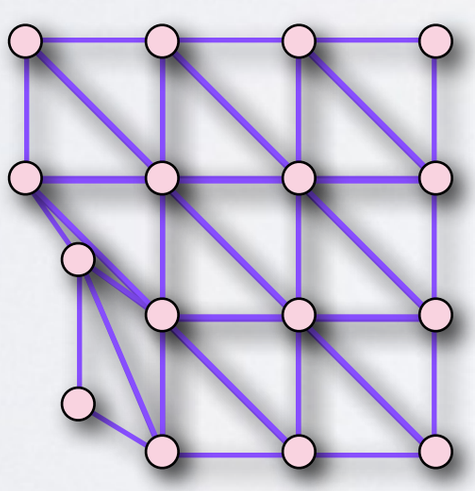
WHY IS IT A BOTTLENECK?



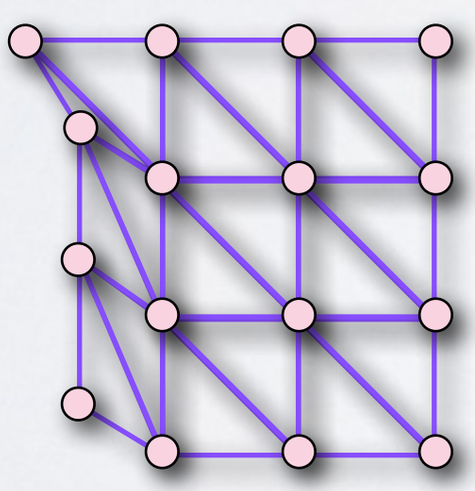
WHY IS IT A BOTTLENECK?



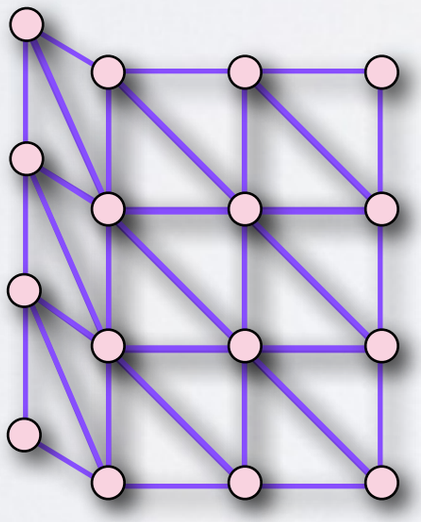
WHY IS IT A BOTTLENECK?



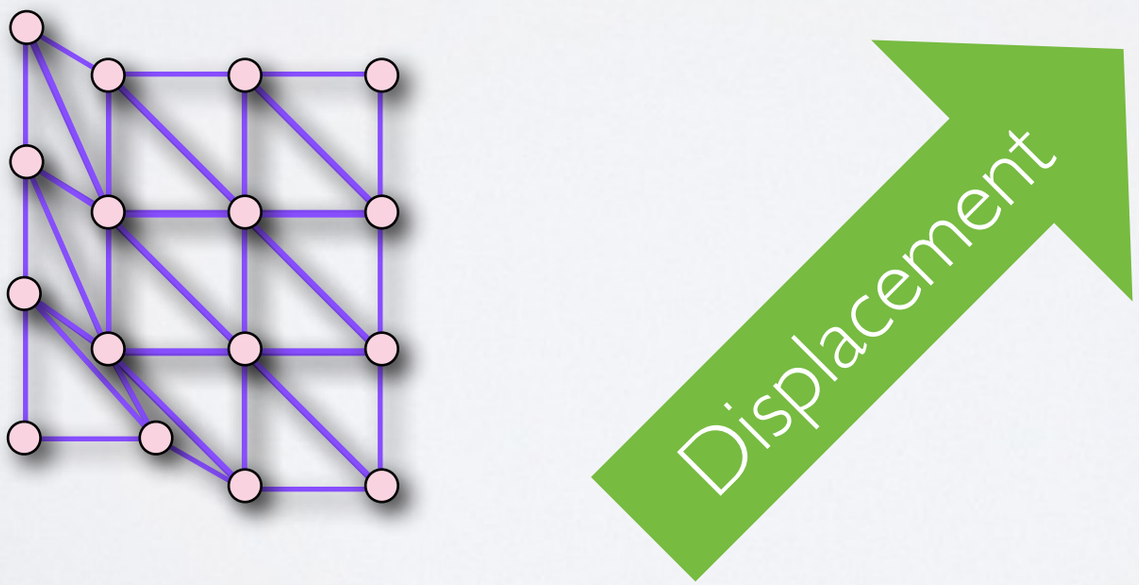
WHY IS IT A BOTTLENECK?



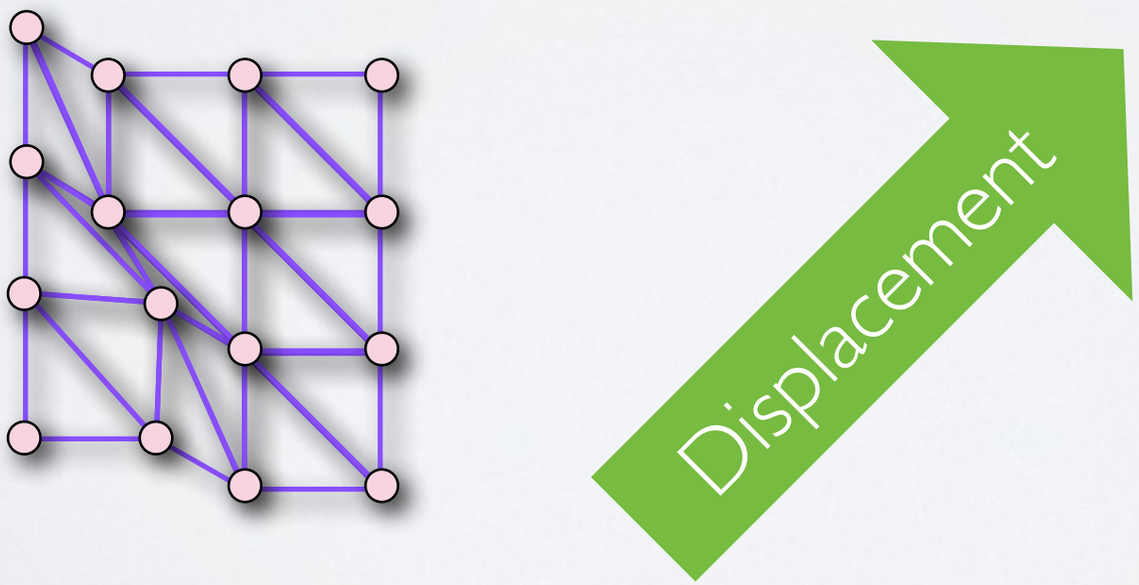
WHY IS IT A BOTTLENECK?



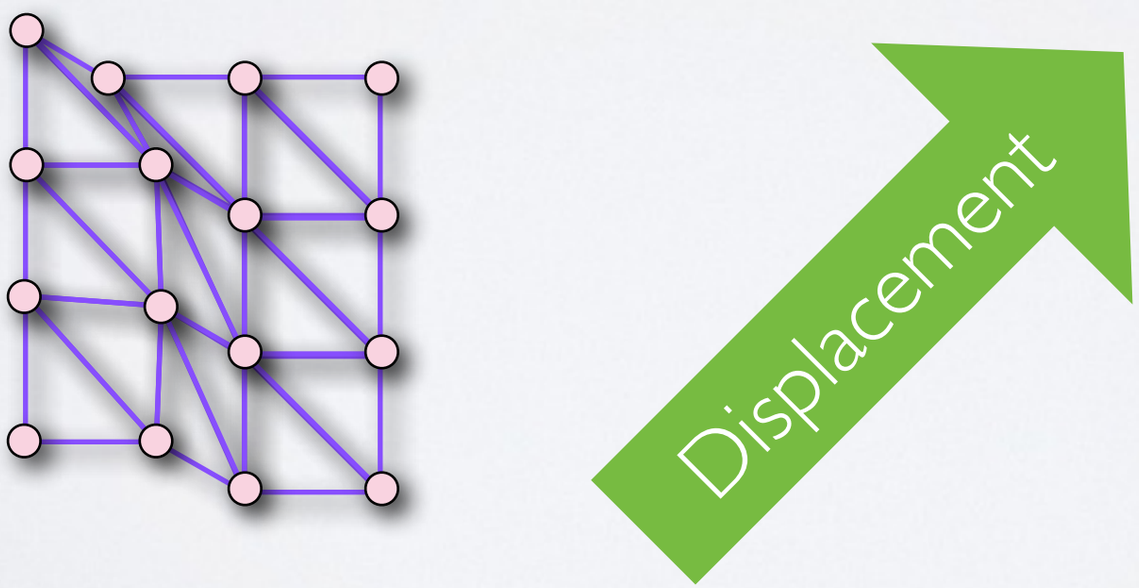
WHY IS IT A BOTTLENECK?



WHY IS IT A BOTTLENECK?

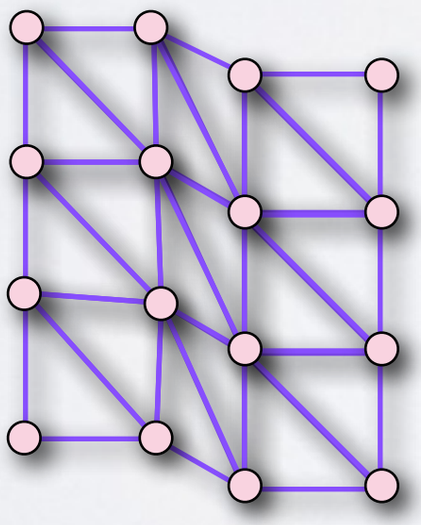


WHY IS IT A BOTTLENECK?



WHY IS IT A BOTTLENECK?

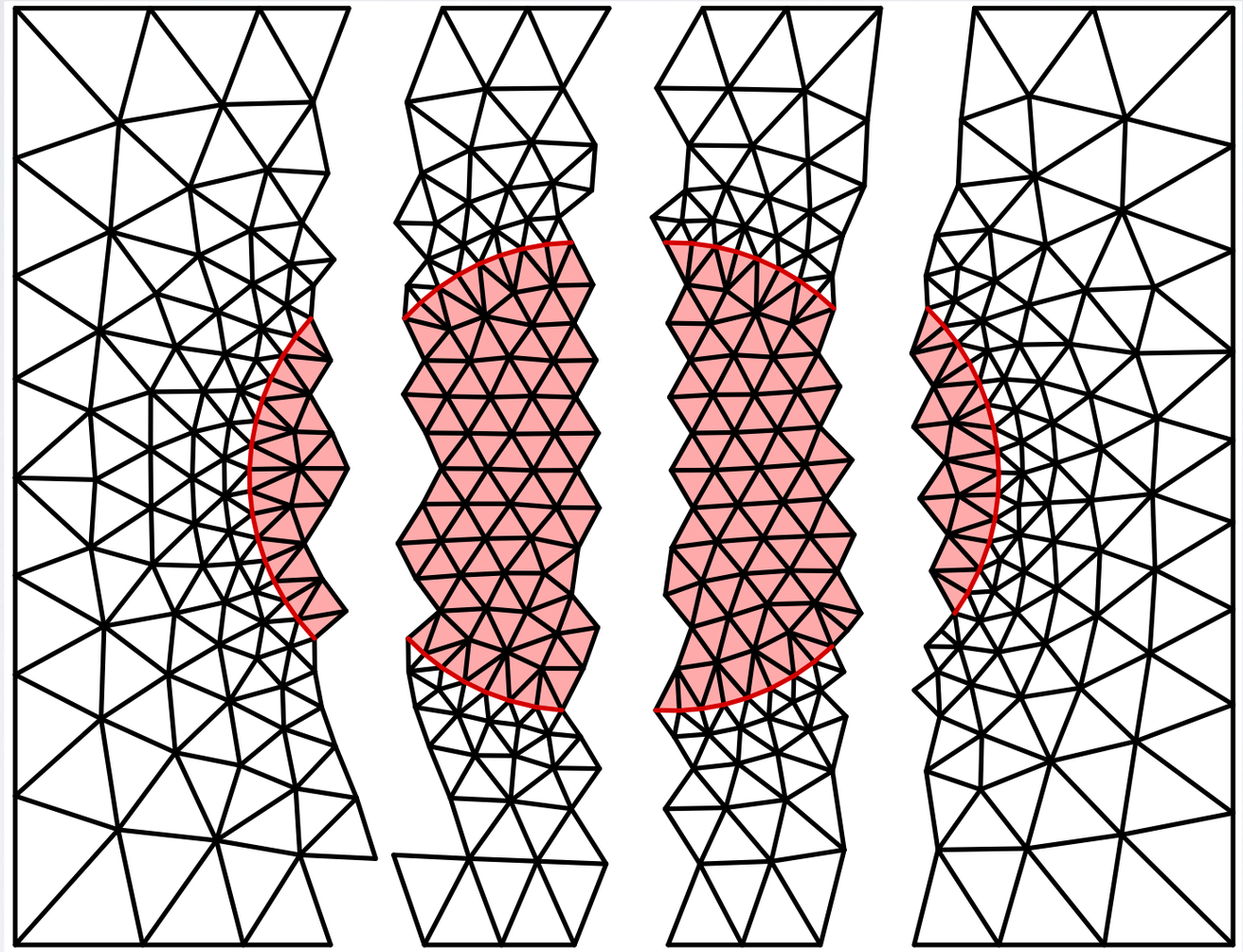
And so on....



WE HAD TO INVENT A NEW MOVING MESH METHOD

- Block and batch all mesh operations into six computational building blocks
- Each block highly configurable in terms of quality and stop criteria
- Apply domain decomposition to increase parallelism

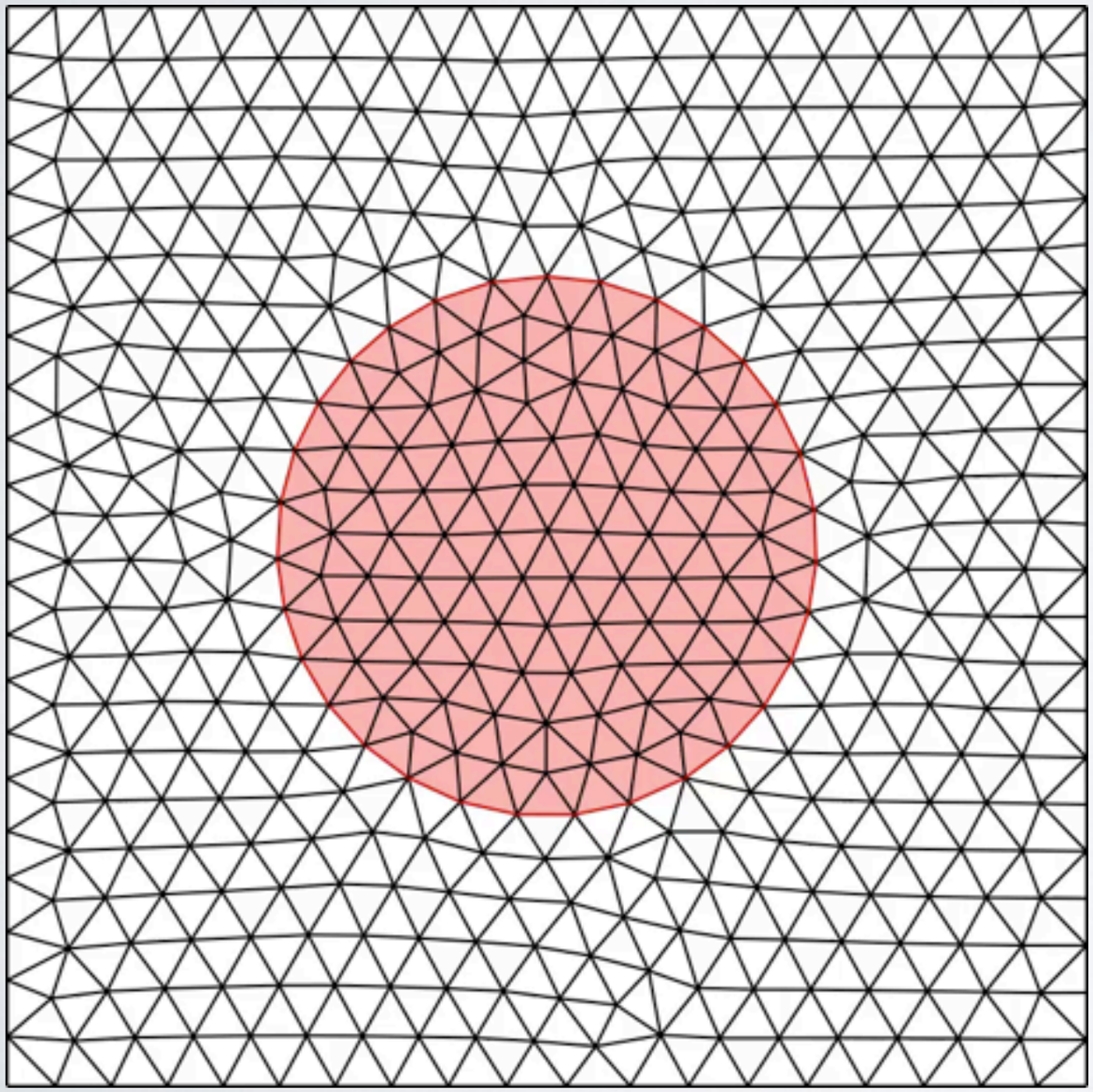
DOMAIN DECOMPOSITION

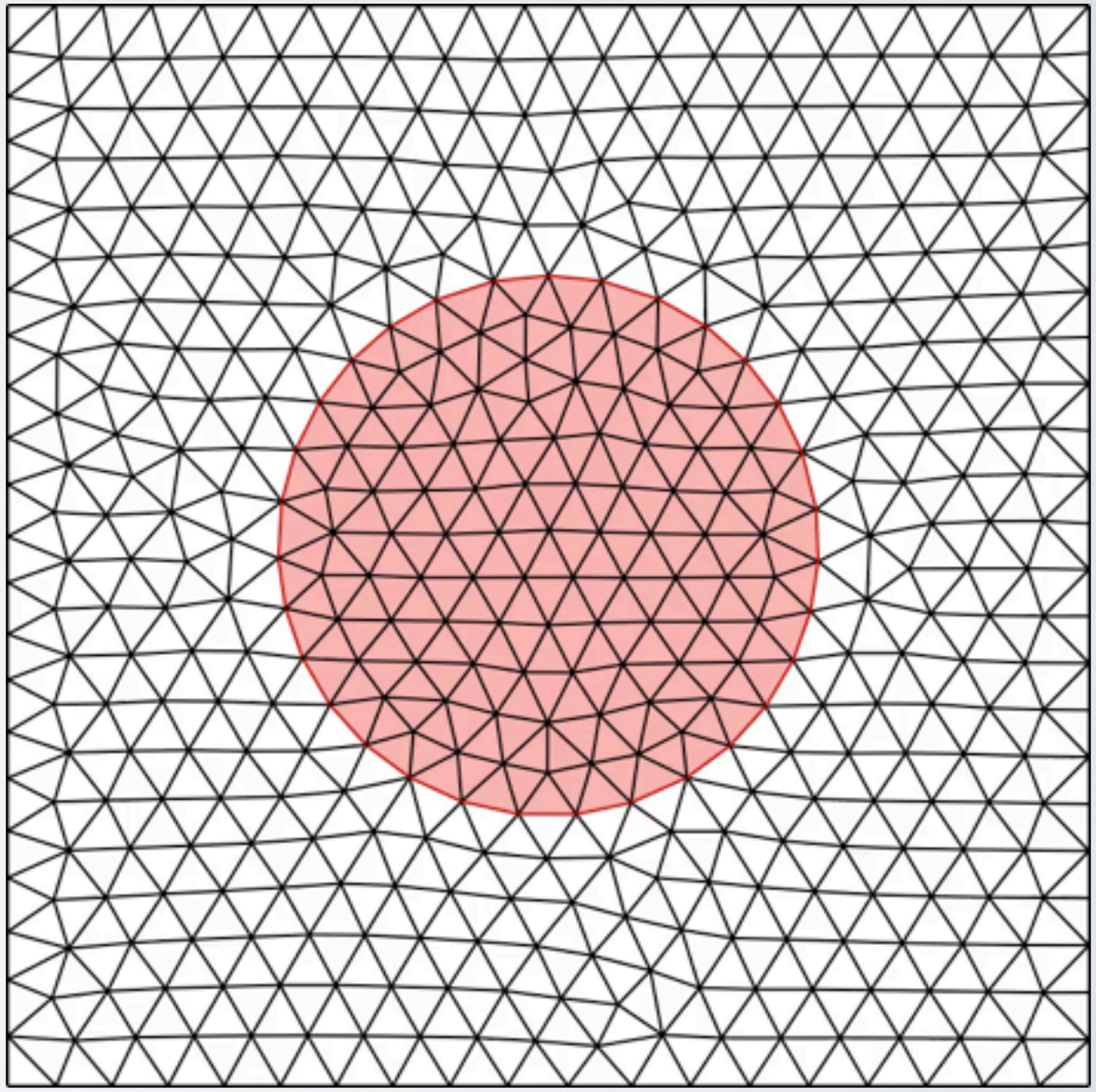


A FEW DETAILS

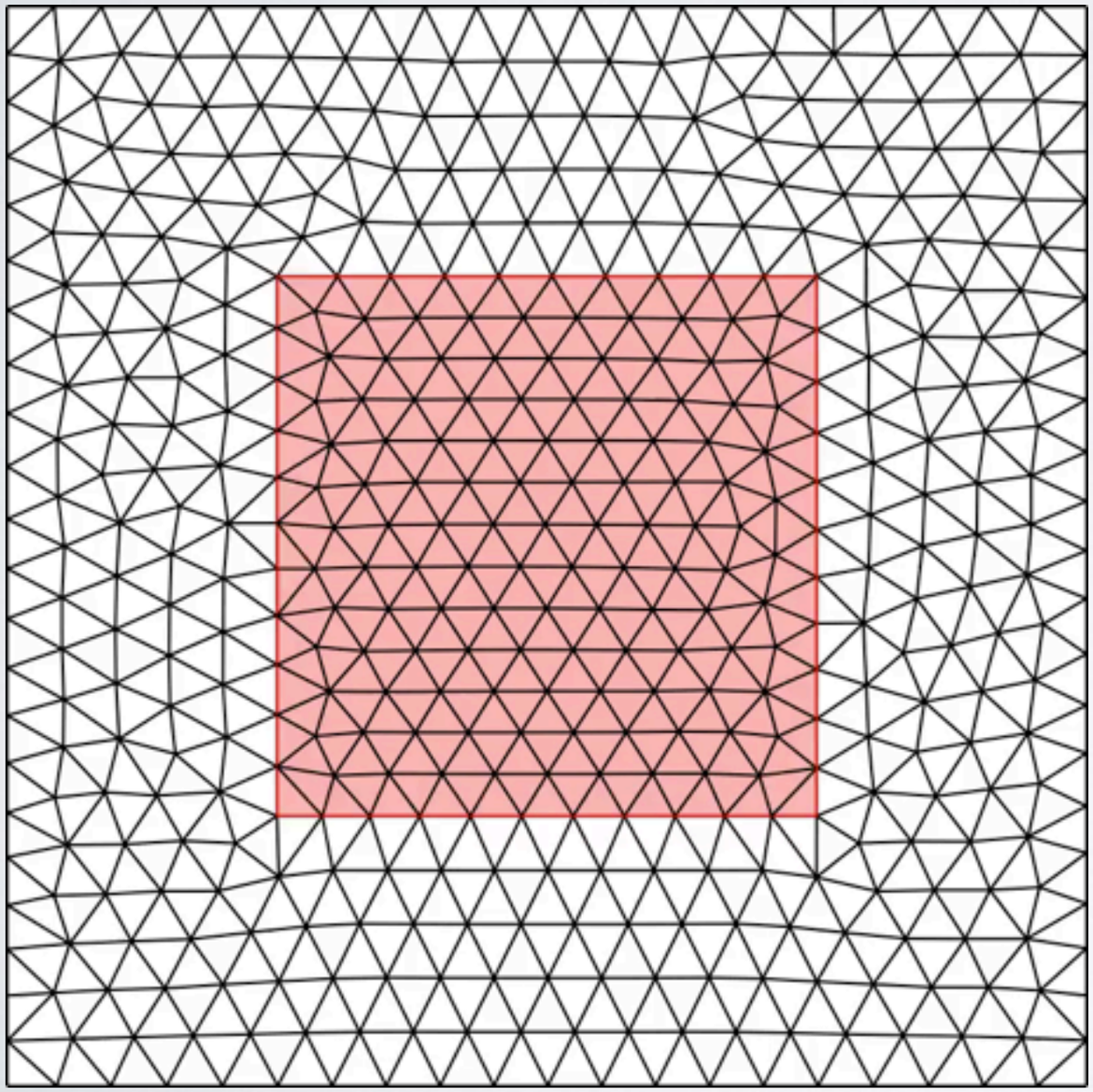
- Dynamic slabbing to better support load balancing
- Shift slabs during outer iteration to avoid boundary artefacts
- Replacement copy strategy with locked borders instead of ghost cells
- Run six computational blocks on each slab until stop criteria

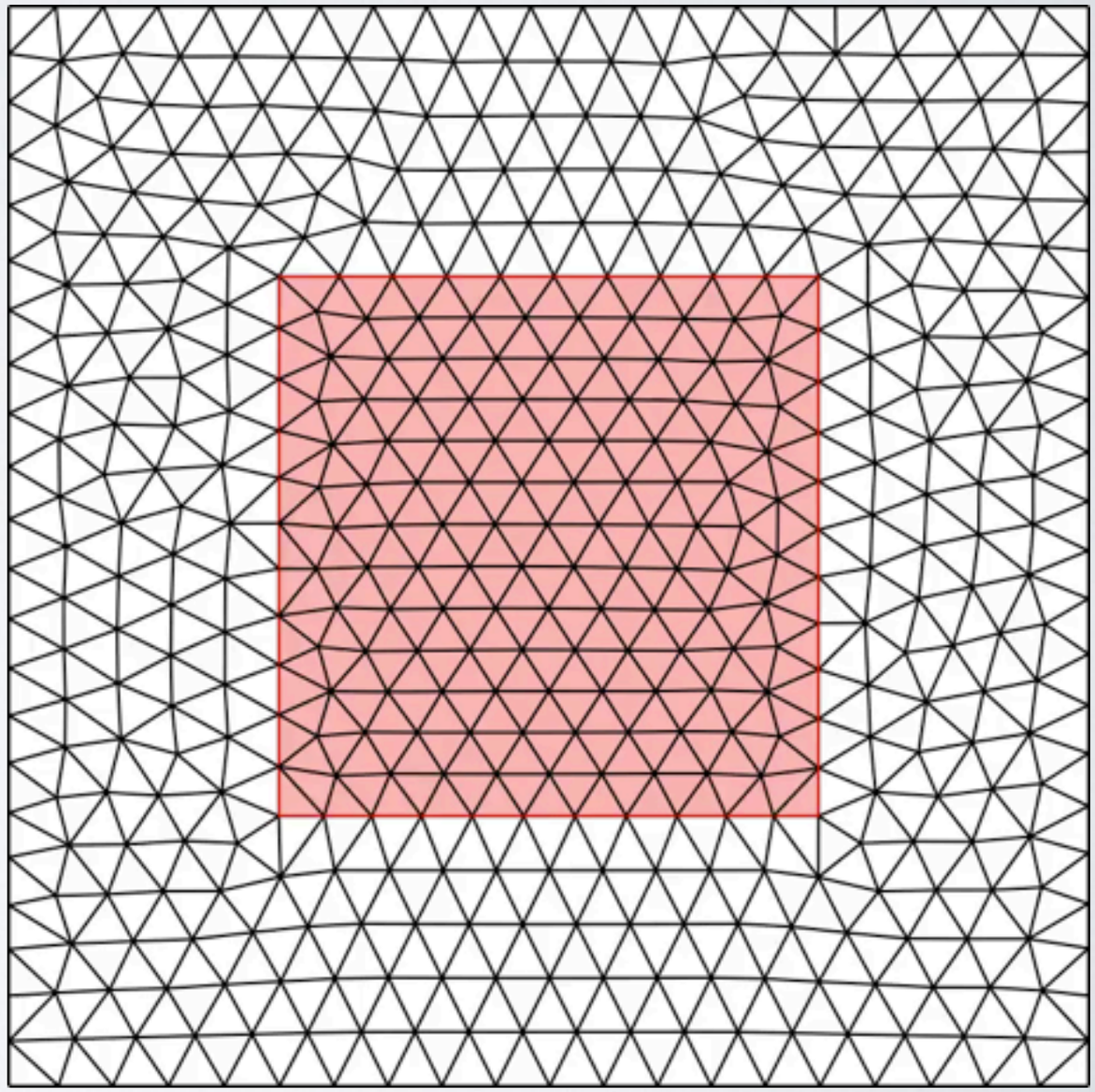
PARALLEL TRANSLATE



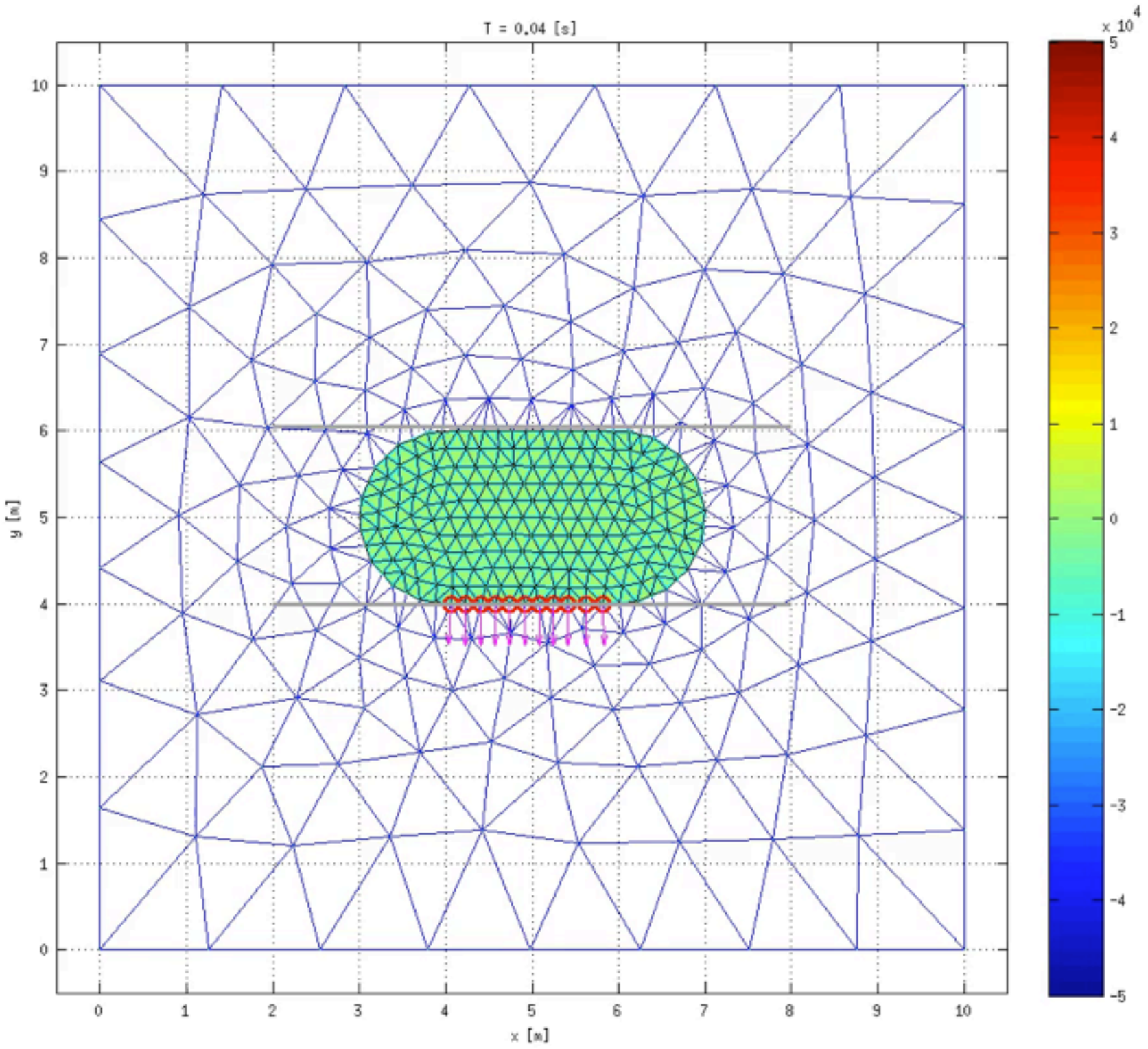


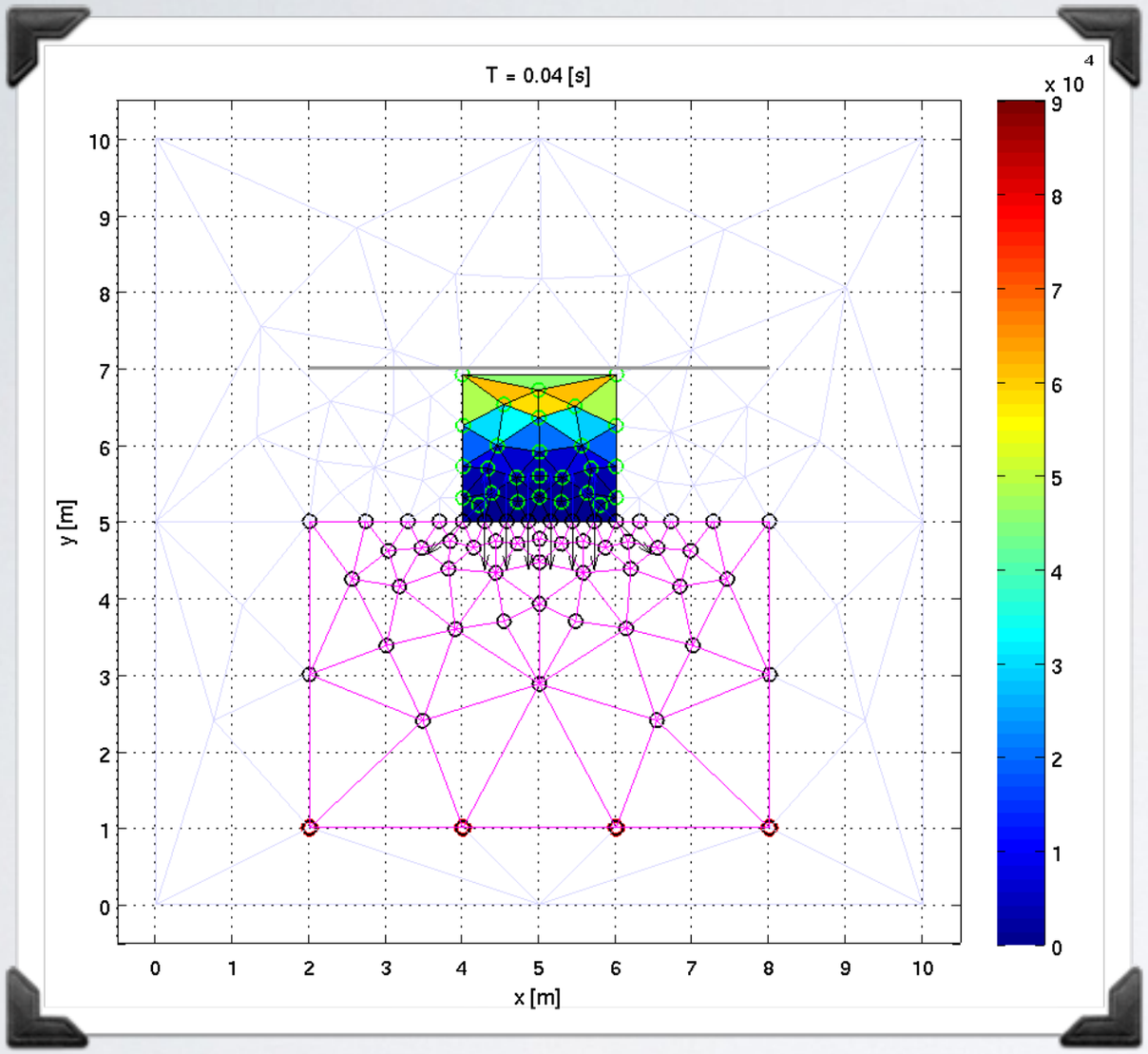
PARALLEL VORTEX EXAMPLE

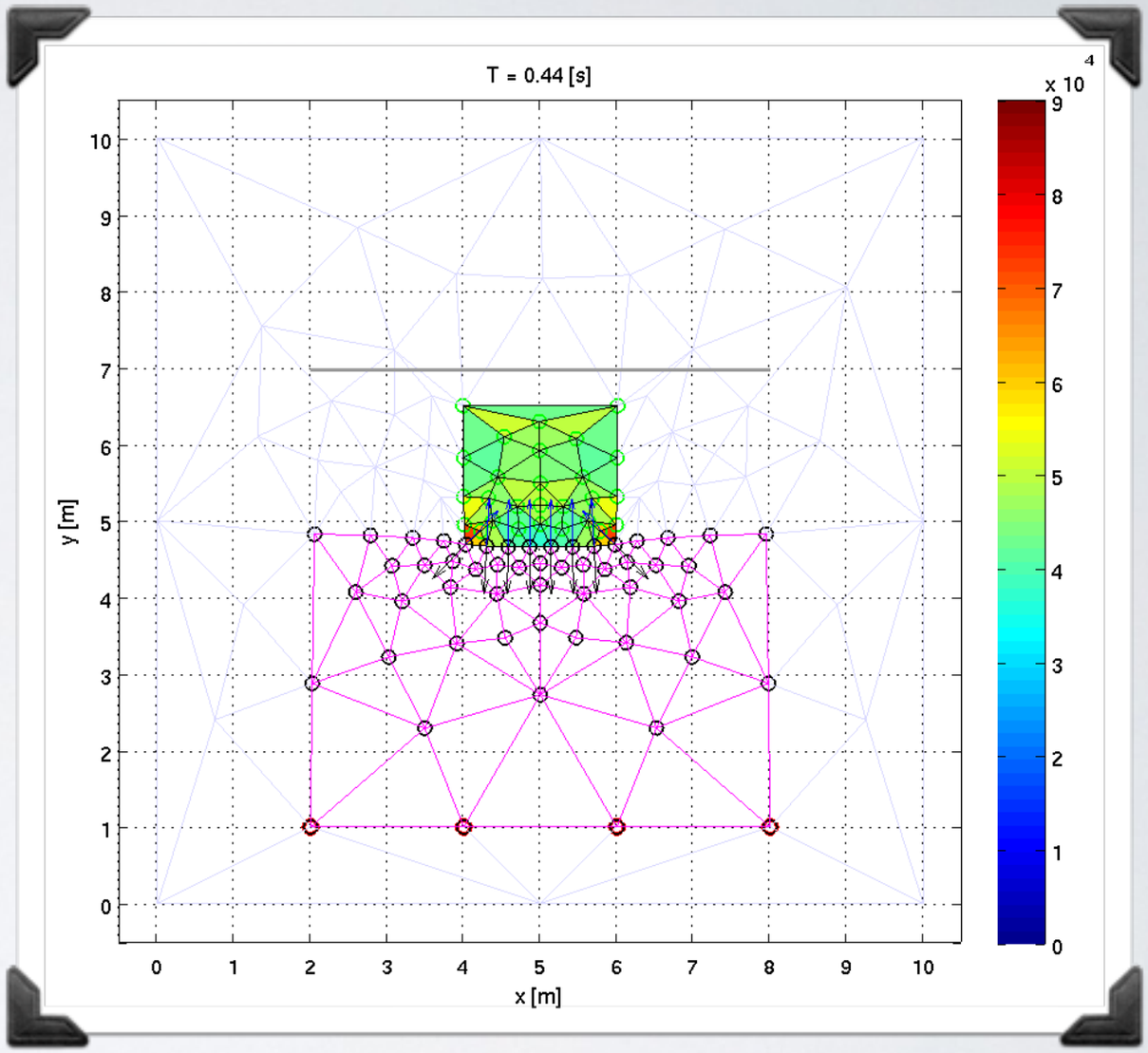




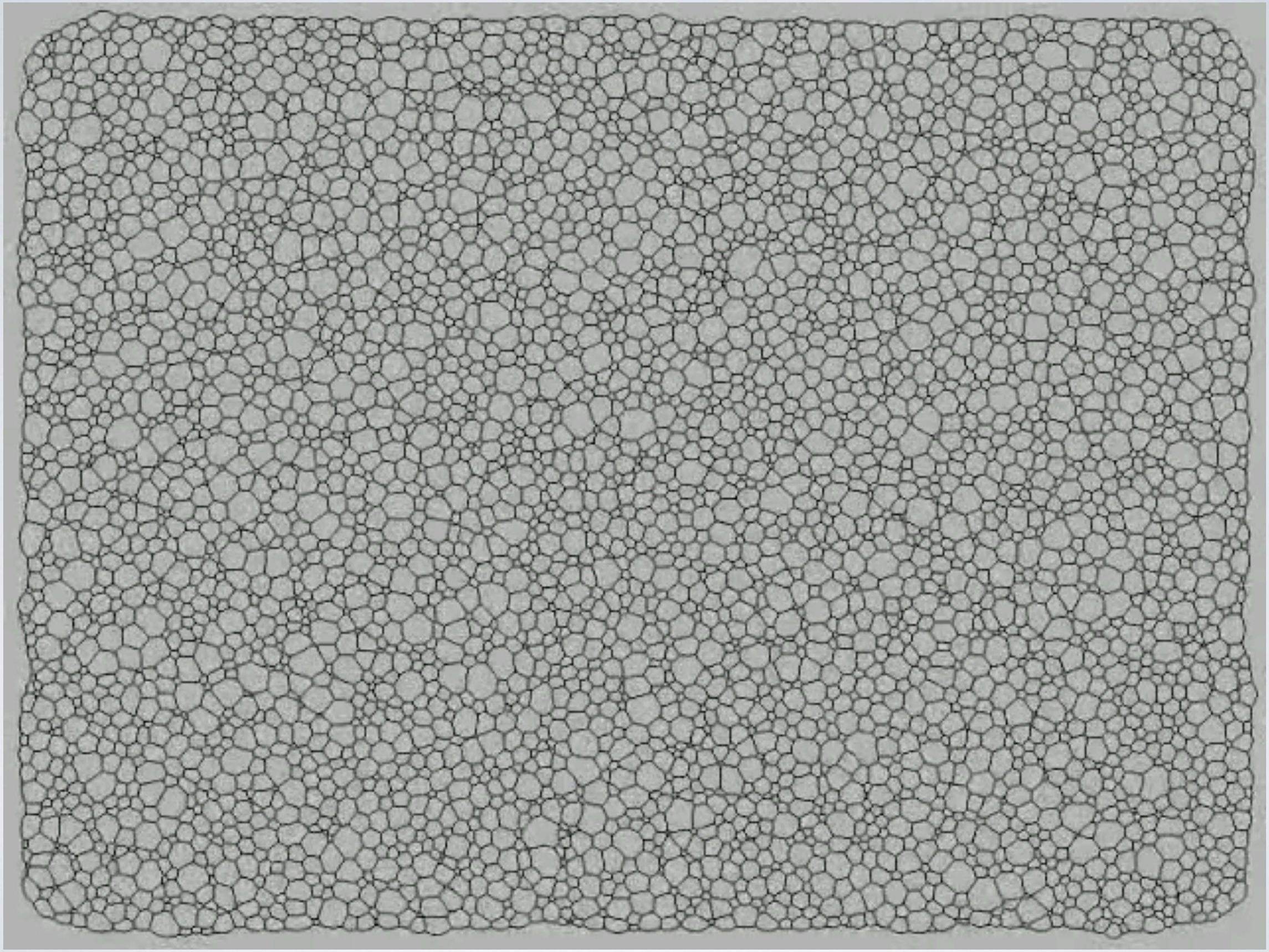
PATCH WORK EXAMPLES FOR BIOMECHANICS MODELLING


T = 0.04 [s]
x 10^4
5
4
3
2
1
0
-1
-2
-3
-4
-5
y [m]
x [m]
0
1
2
3
4
5
6
7
8
9
10


T = 0.04 [s]
x 10^4
9
8
7
6
5
4
3
2
1
0
y [m]
x [m]
0 1 2 3 4 5 6 7 8 9 10


T = 0.44 [s]
x 10^4
9
8
7
6
5
4
3
2
1
0
y [m]
x [m]
0 1 2 3 4 5 6 7 8 9 10

BACKTRACKING



QUESTIONS?