Geometric Predicates for Unconditionally Robust Elastodynamics Simulation

Daniele Panozzo
NYU Geometric Computing Lab

NYU | COURANT INSTITUTE OF MATHEMATICAL SCIENCES

Catastrophic Failures

- Wikipedia: "A **catastrophic failure** is a sudden and total failure from which recovery is impossible. *Catastrophic failures often lead to cascading systems failure.*"

- **Catastrophic failures** require user interaction to fix them.

- We usually do not tolerate catastrophic failures, but we usually accepts them and think of them as unavoidable for meshing and simulation software.

Many reasons, one is hidden in plain sight!

- We usually assume that floating point numbers are equivalent to real numbers
- This is however not the case, here is an example of statements/properties that are commonly used in numerical algorithms:
    - Multiplication distributes over addition
    - The gradient of a function is an ascend direction
    - The point at the intersection of two non-degenerate, non-parallel intersecting segments AB, CD is a point of both AB and CD
- The statements above are "**mostly**" true, so algorithms using them "**often**" work

Geometric Computing Lab

Faculty


Daniele Panozzo


Denis Zorin

Postdocs


Daniel Zint


Michael Tao

University of Victoria


Teseo Schneider

PhD Students


Leyi Zhu


Siqi Wang


YunFan Zhou

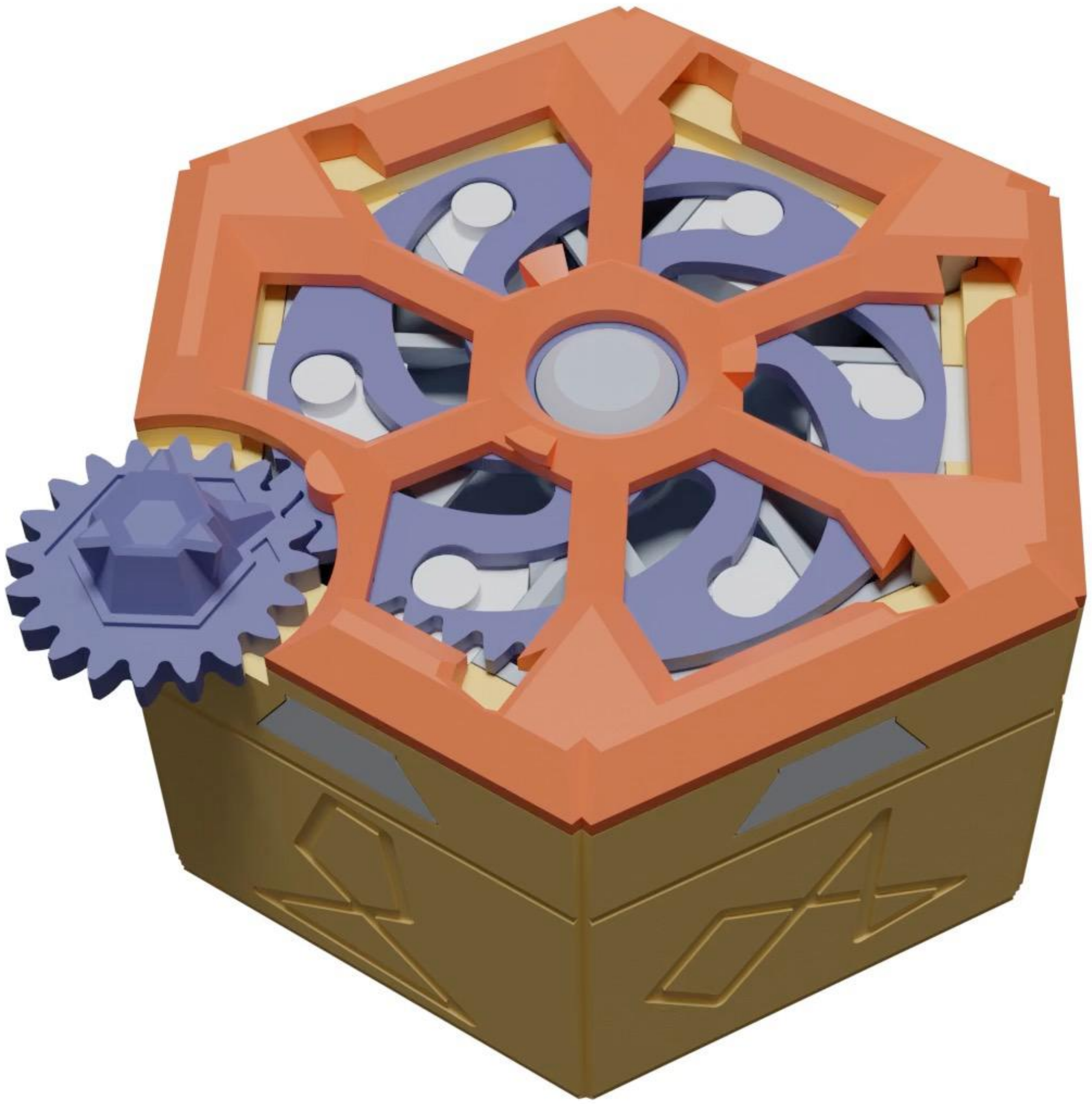

Zizhou Huang


Arvi Gjoka


Ryan Capouellez


Max Paik

Robust Forward Elastodynamics

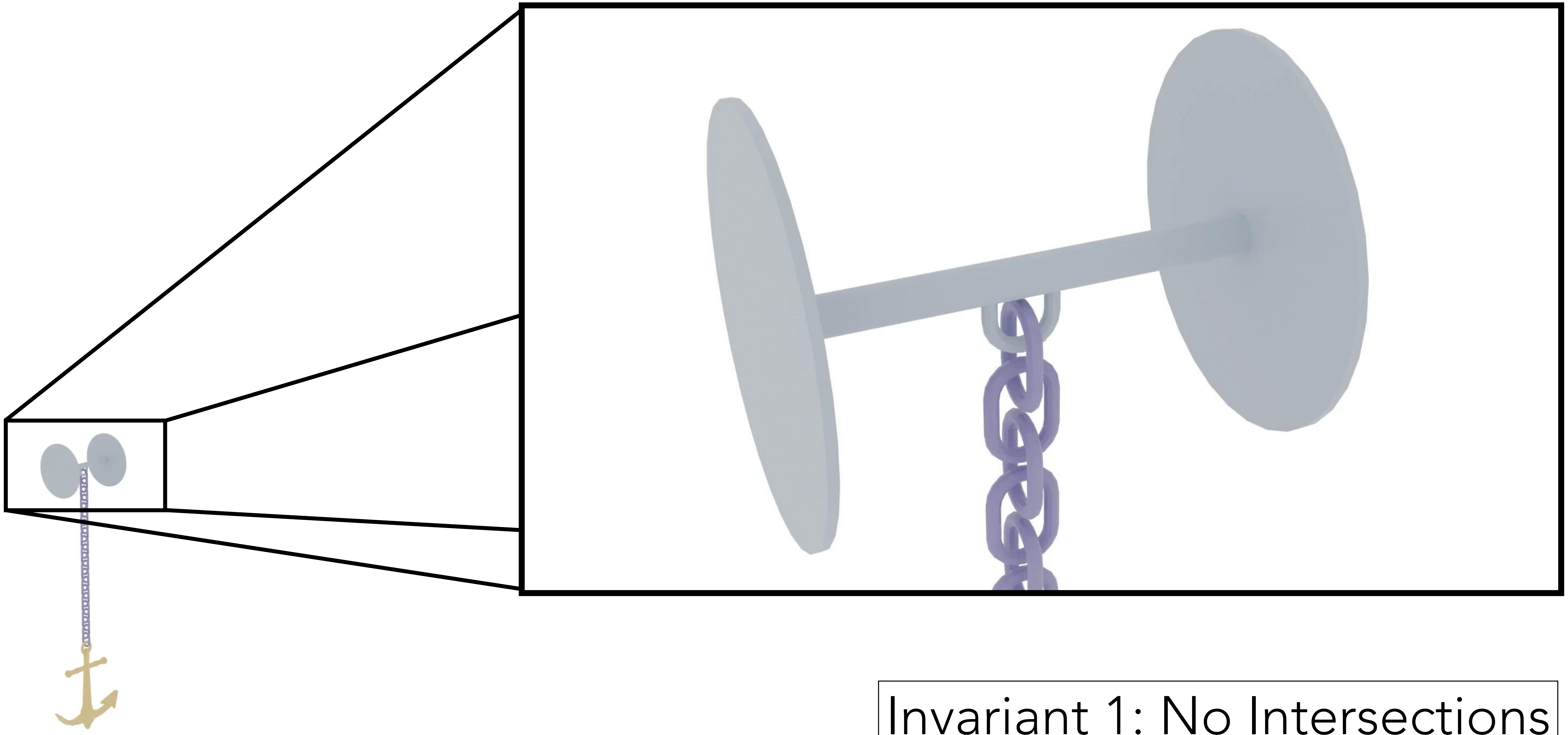


Intersection-free Rigid Body Dynamics

Zachary Ferguson, Minchen Li, Teseo Schneider, Francisca Gil Ureta, Timothy Langlois, Chenfanfu Jiang, Denis Zorin, Danny M. Kaufman, Daniele Panozzo,

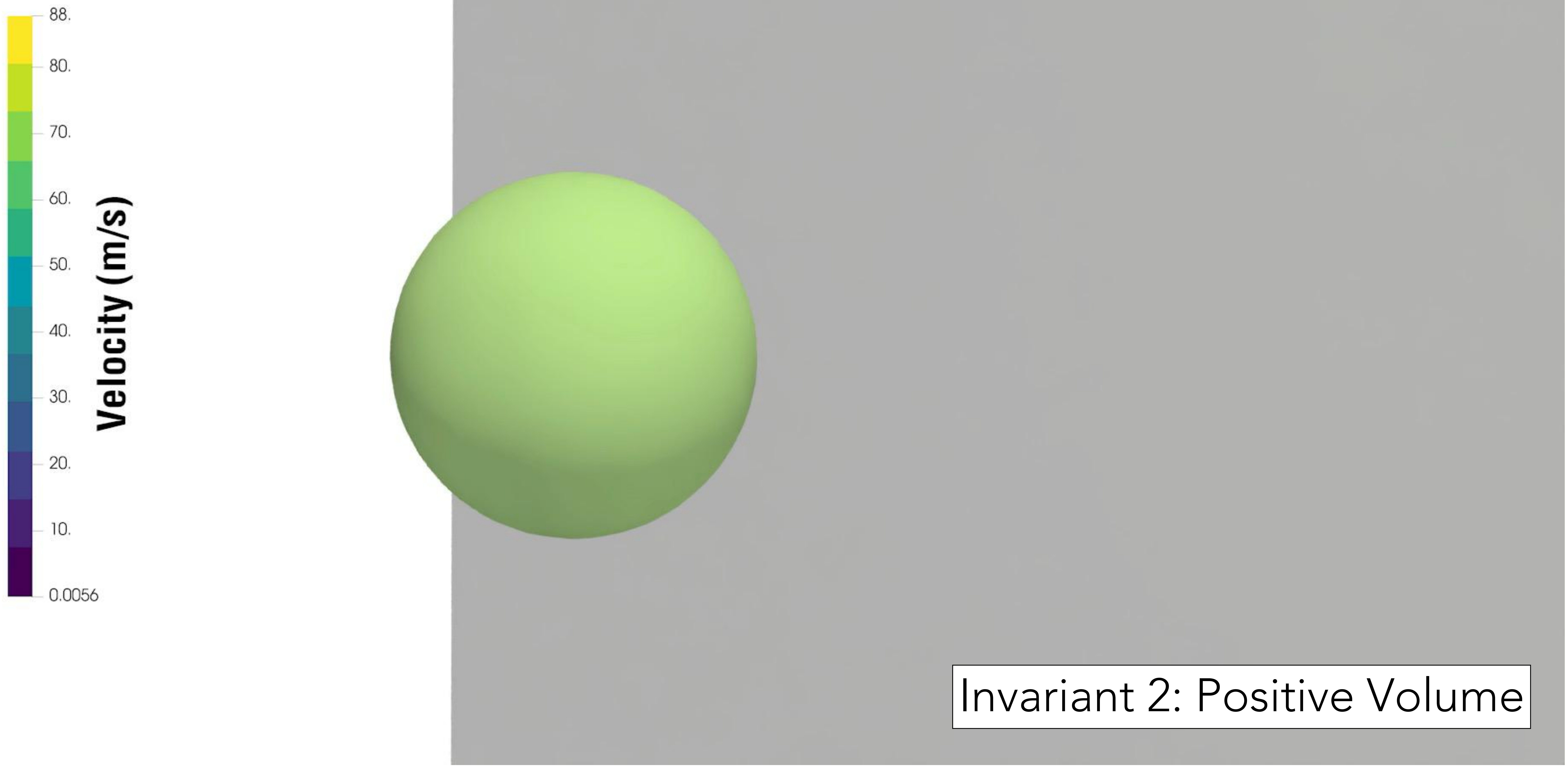
ACM Transaction on Graphics (SIGGRAPH), 2021

[Paper] [Video 1] [Video 2] [Website]



Invariant 1: No Intersections

dt: 2e-5s



Invariant 2: Positive Volume


Persp
cam1
Houdini
Non-Commercial Edition

Copyright Steve Abramowitch


Top
No cam
Non-Commercial Edition
Persp
No cam
Non-Commercial Edition
Front
No cam
Back
No cam
Launch flipbook render
Houdini
Non-Commercial Edition

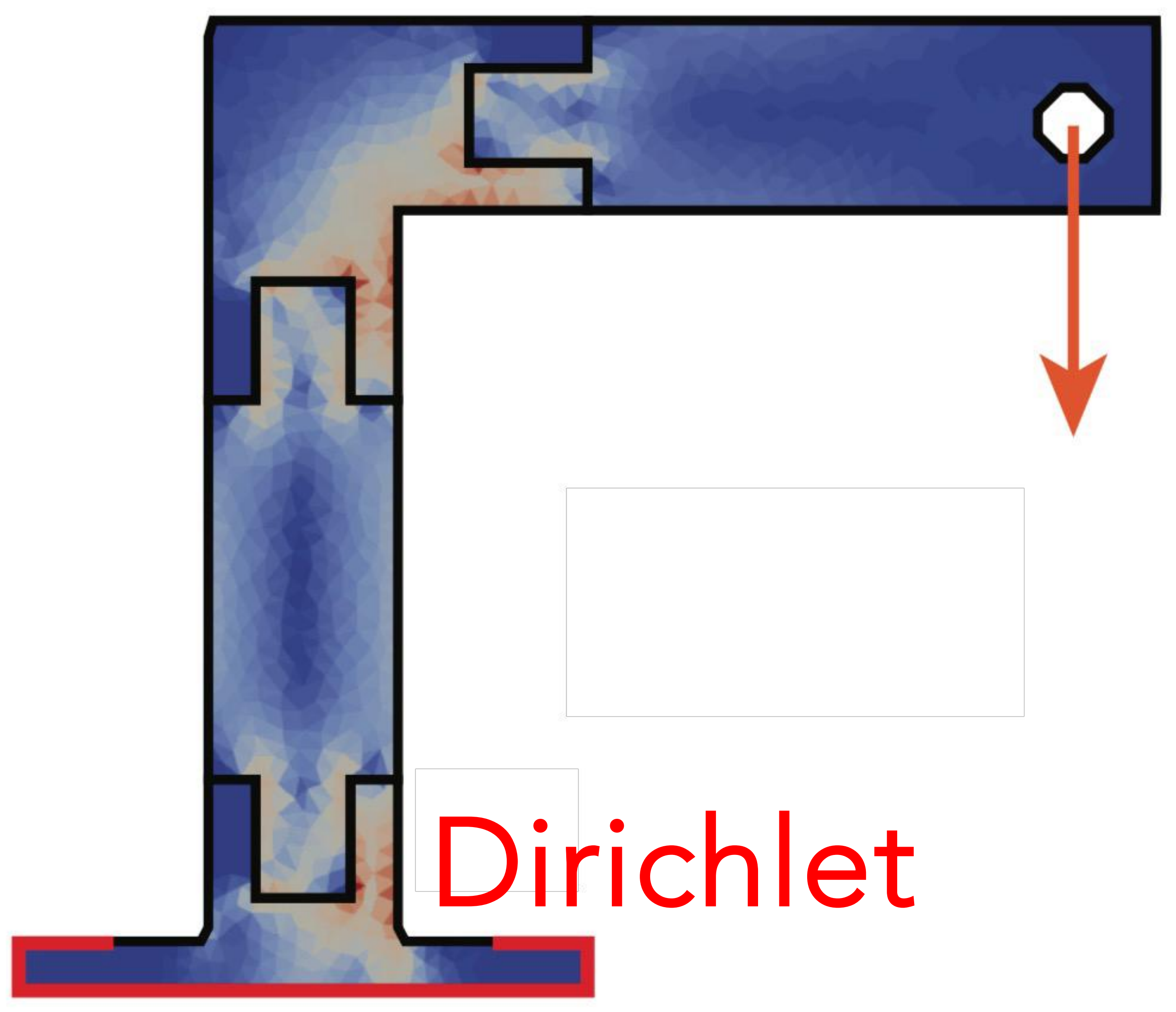
Inverse Problems

Forward Simulation

Input:

Domain:
Object (Mesh)

Boundary conditions:
Forces
Displacements

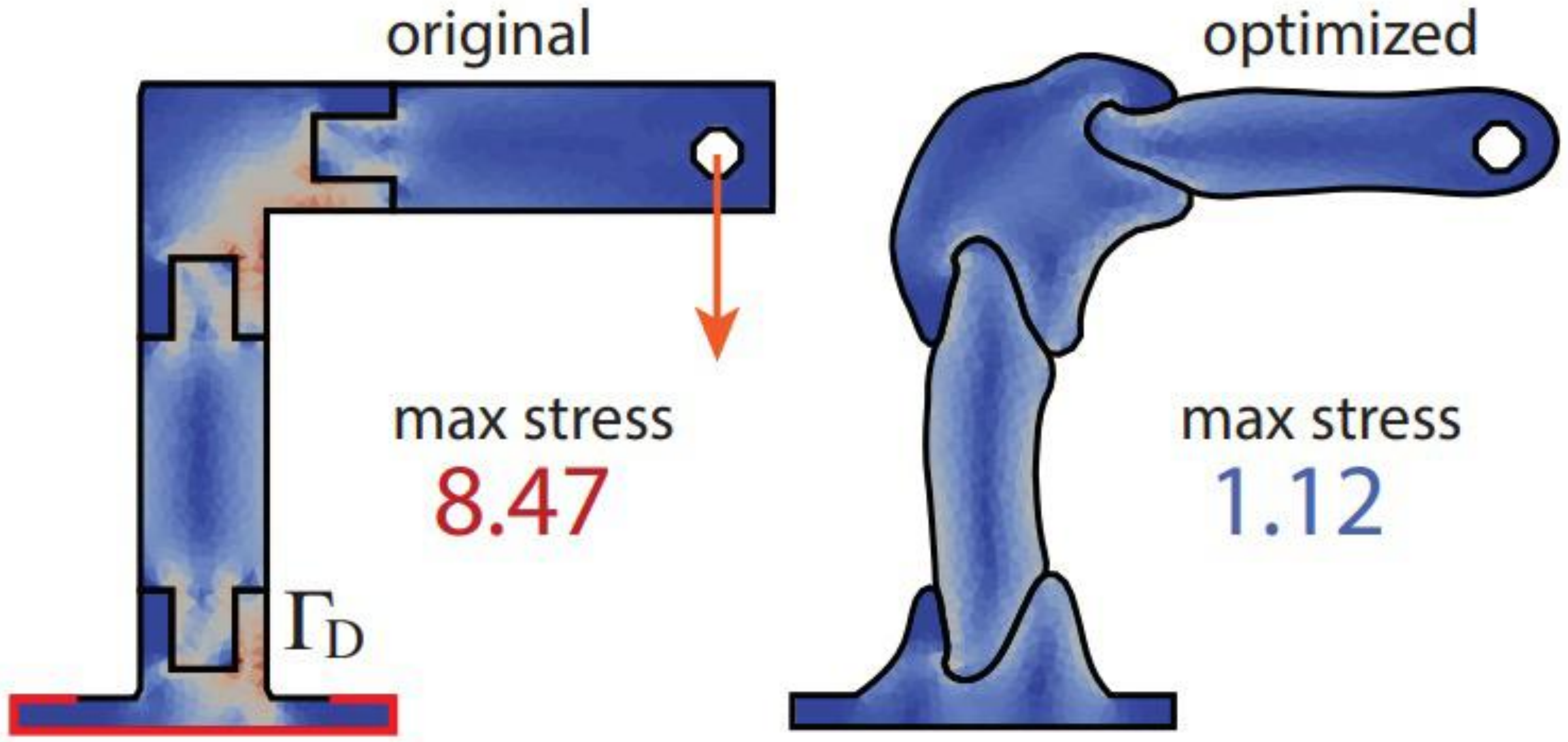


Output: Stress field = $\boldsymbol{\sigma}(x)$

maximize strength ⟺ minimize max stress ⟺ $\min \; J(\Omega) = \left(\int_\Omega (\|\sigma\|_F)^p \, dx \right)^{1/p}$

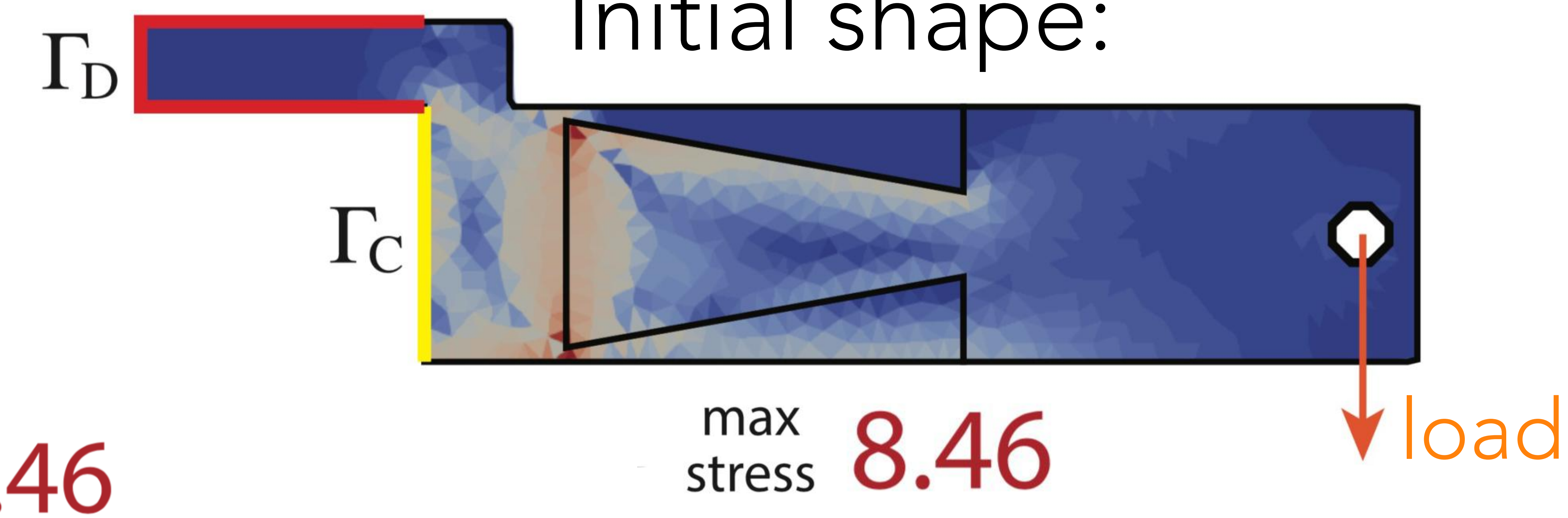
Requirement: Differentiability

PDE-constrained Optimization

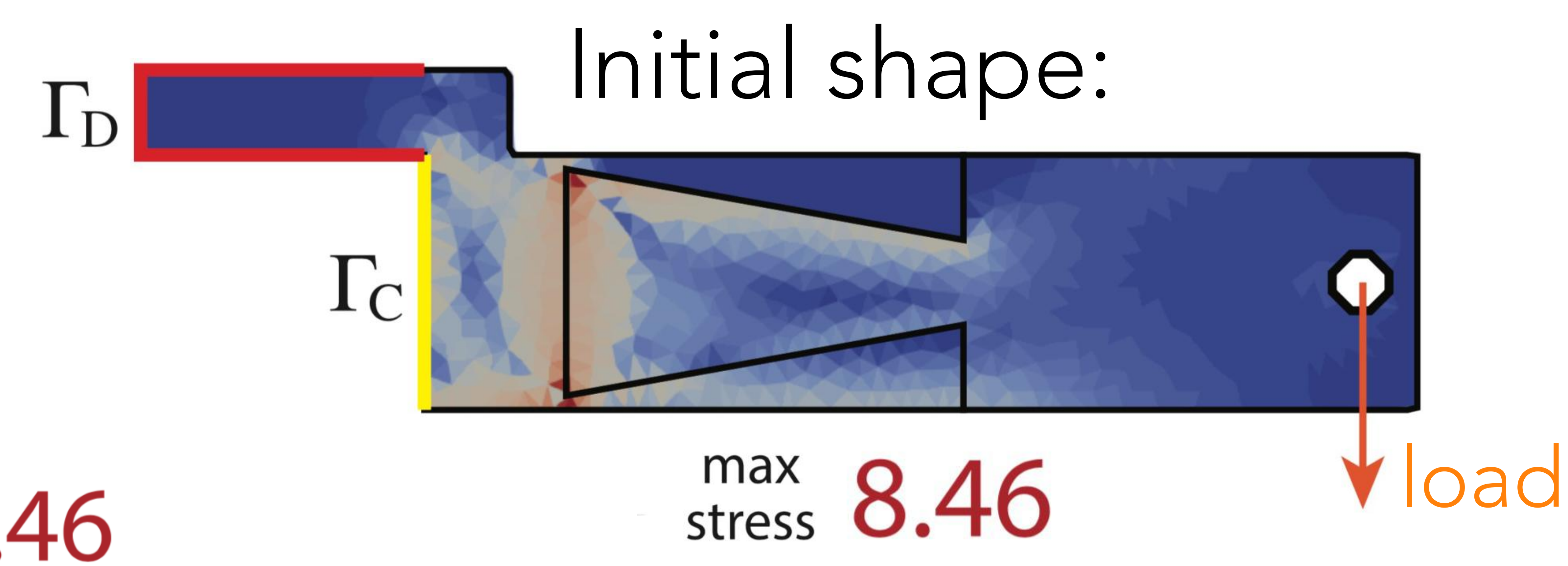


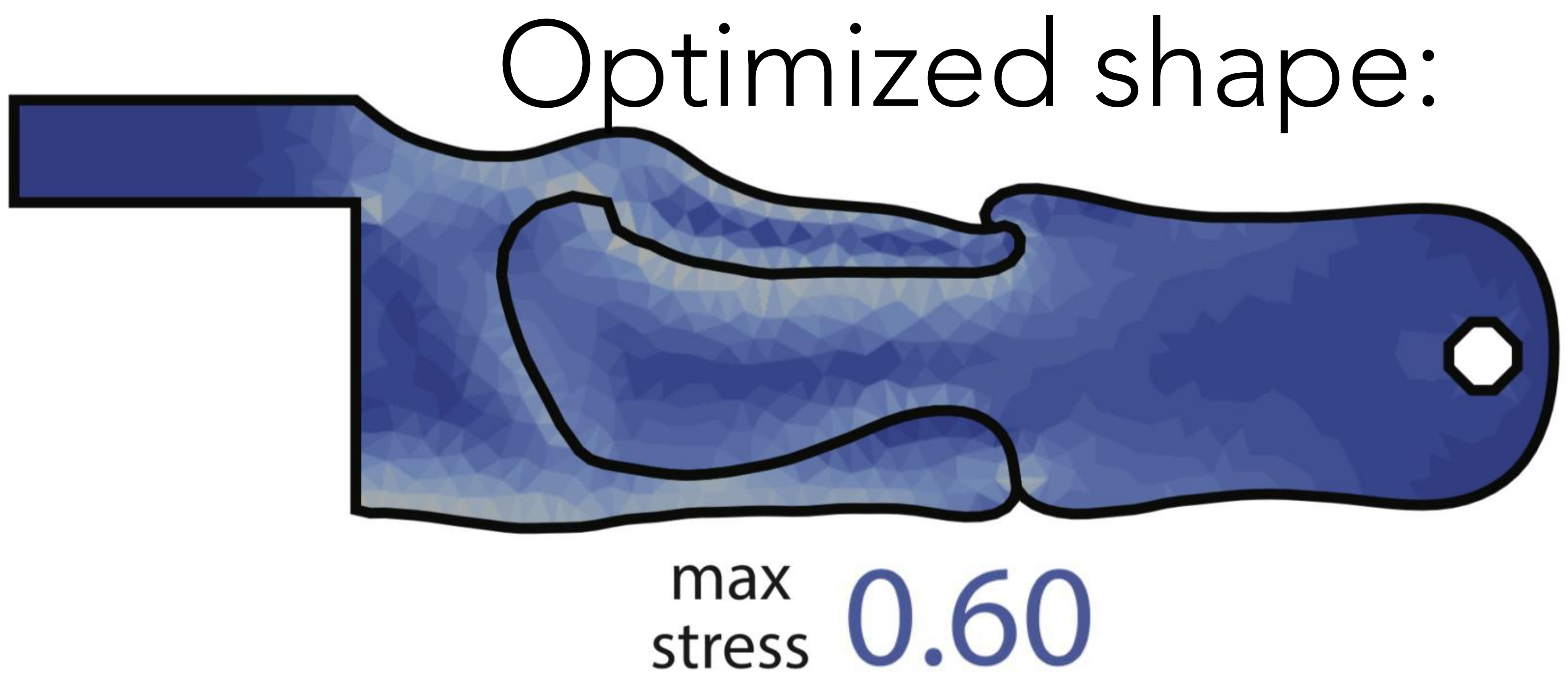
Deformable-Deformable (RD)

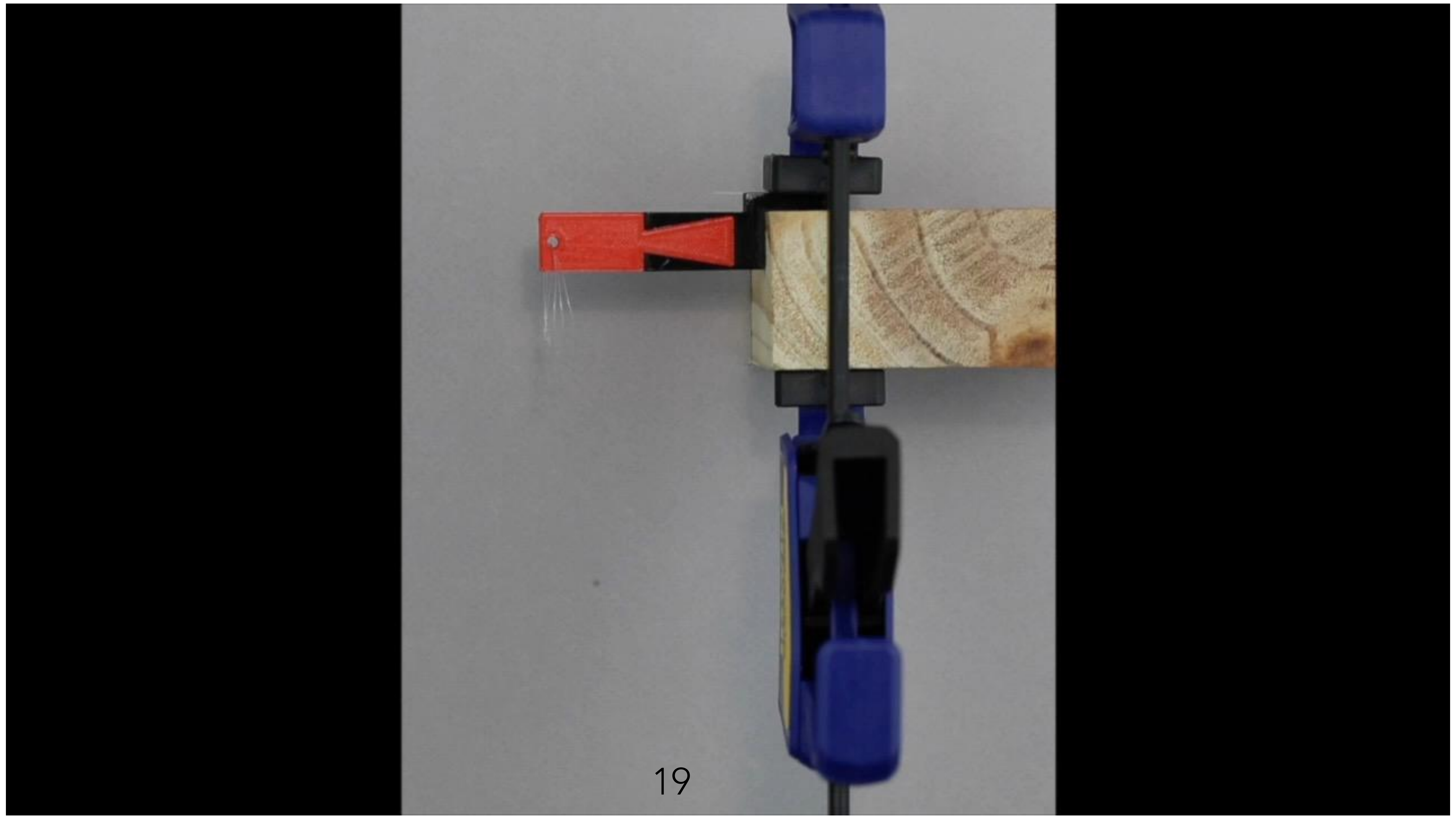
Initial shape:

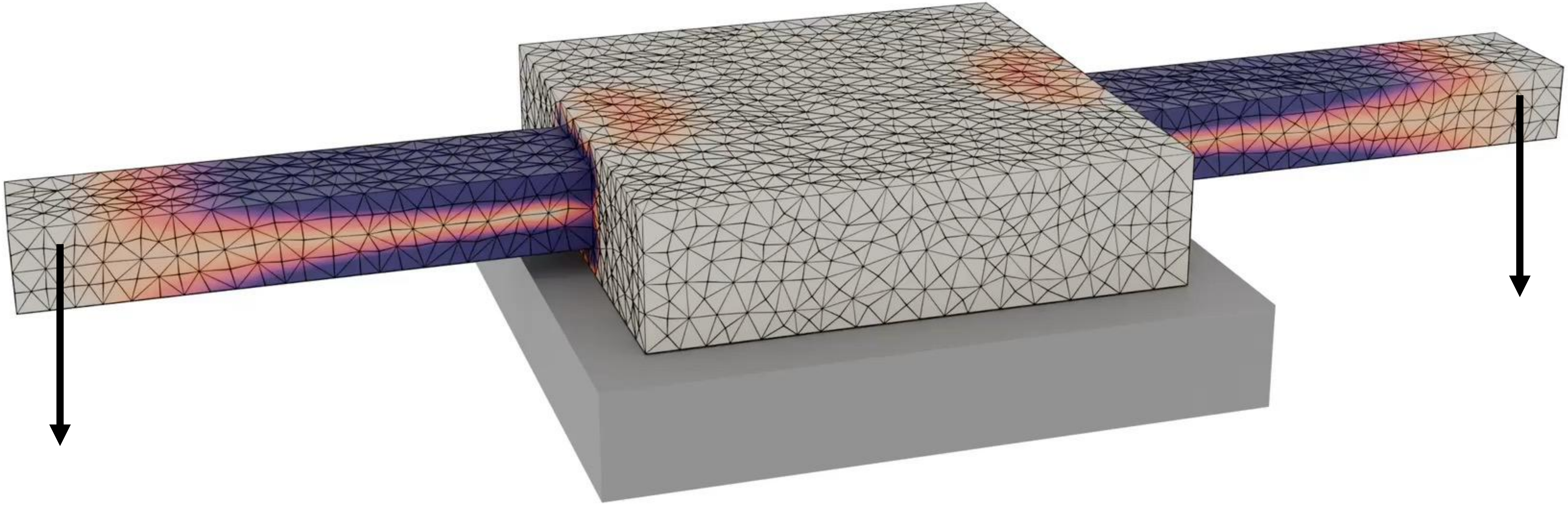


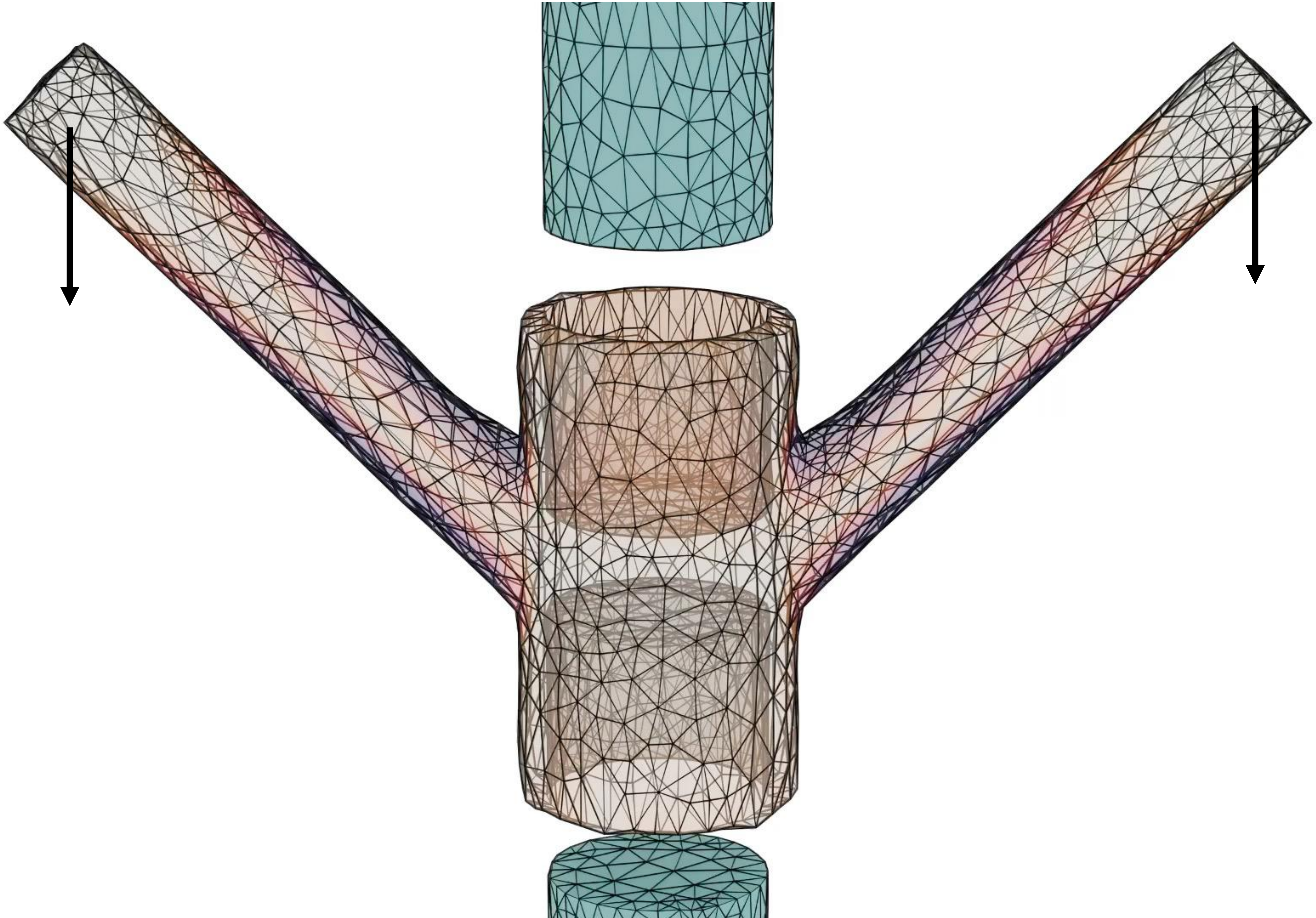
Deformable-Deformable (RD)

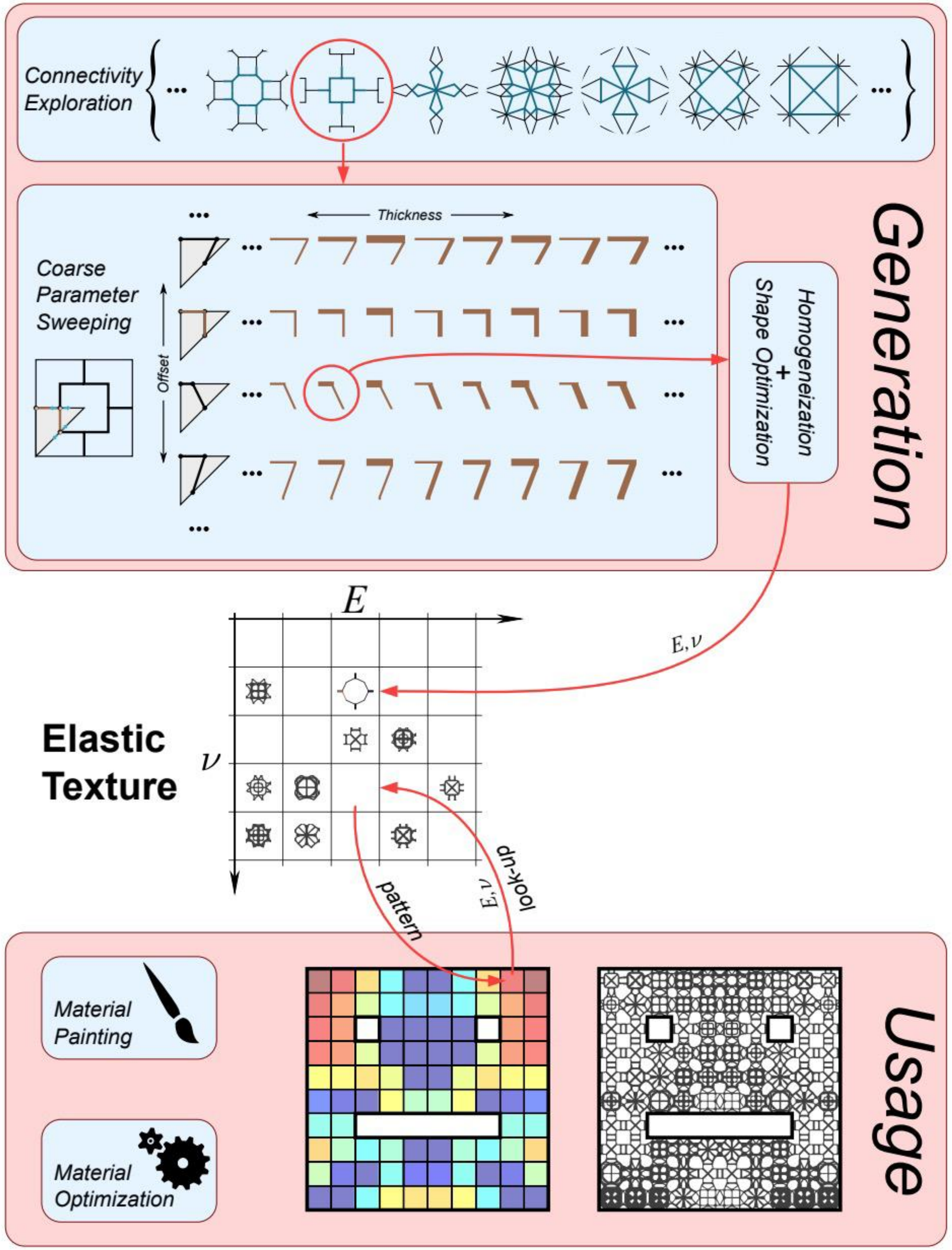


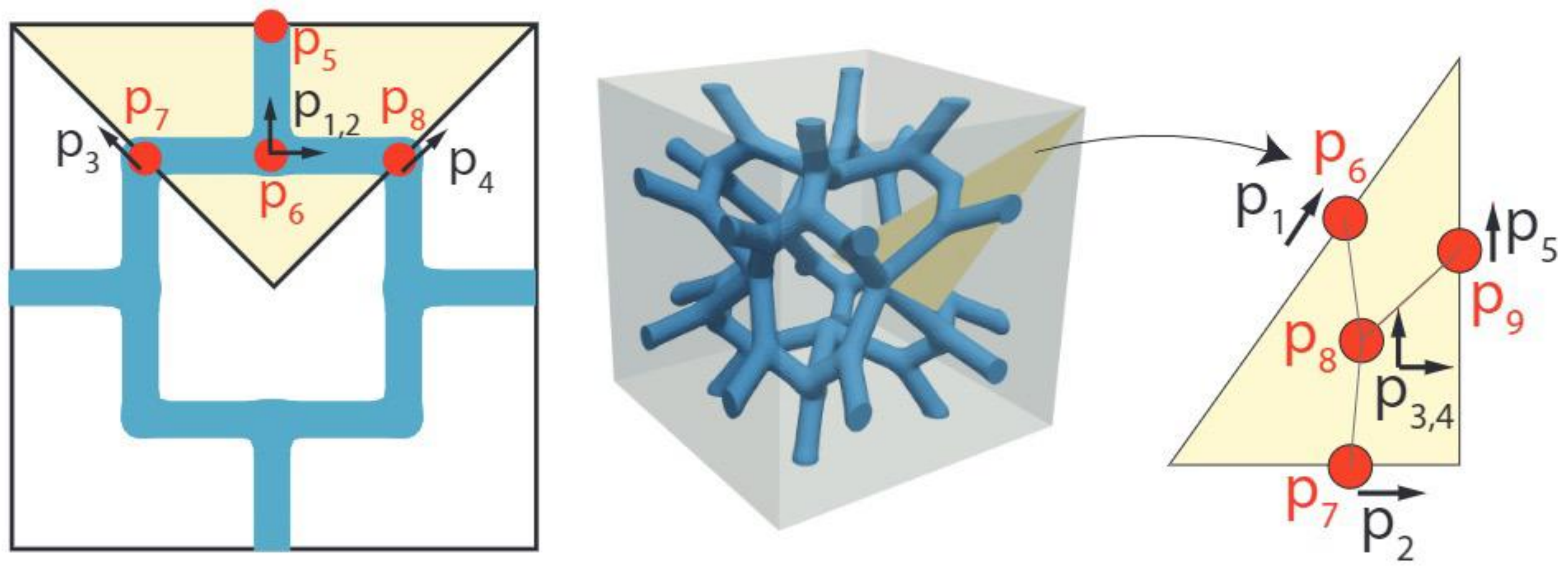




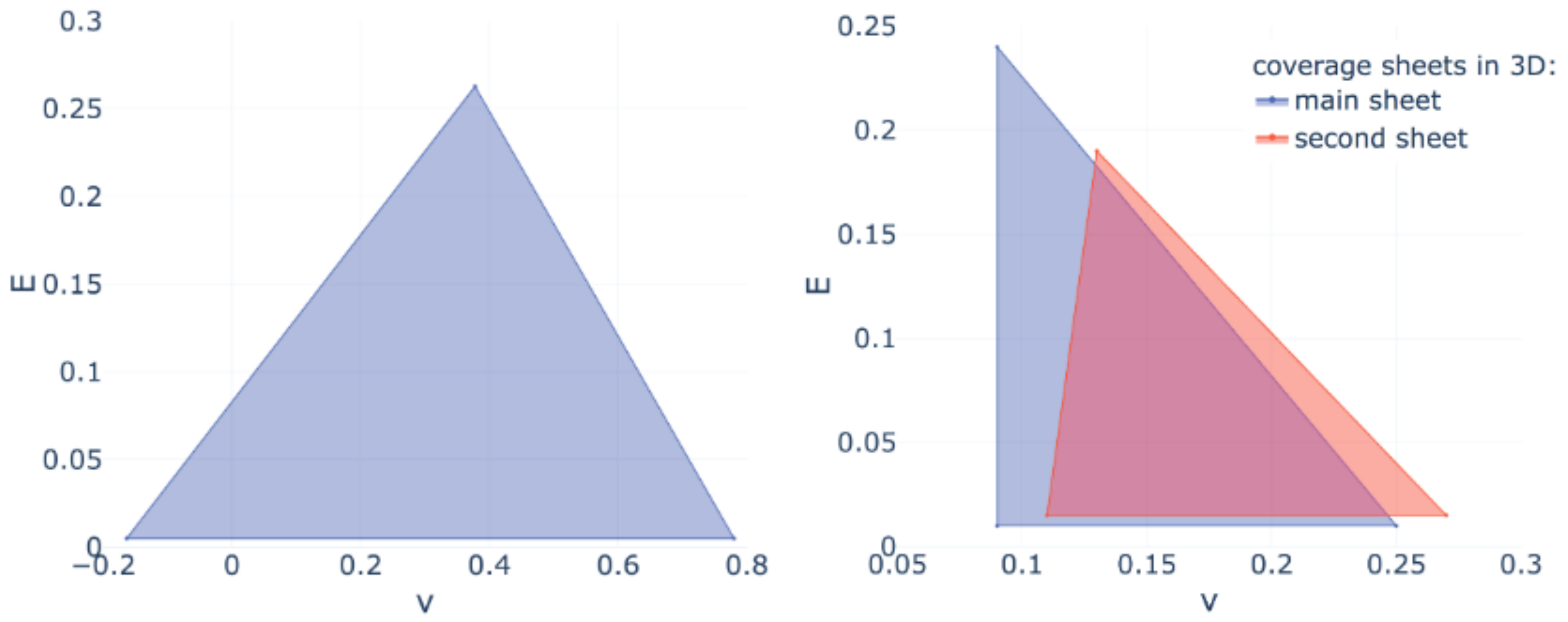




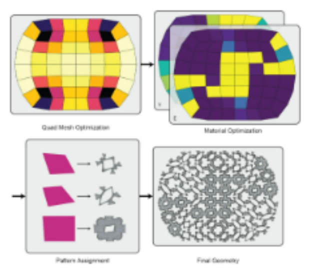




(a) 2D (b) 3D



(a) 2D (b) 3D

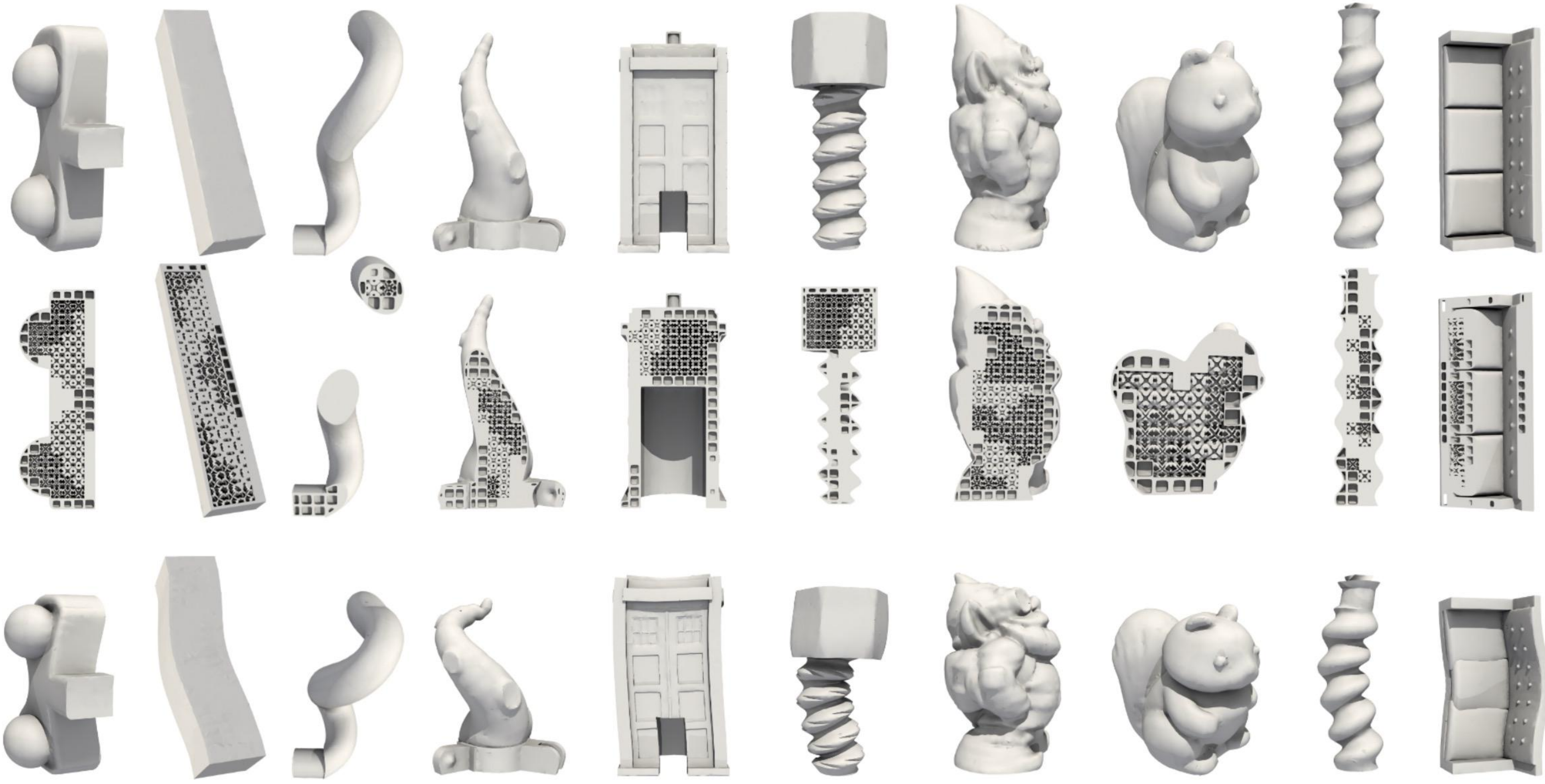


A Low-Parametric Rhombic Microstructure Family for Irregular Lattices

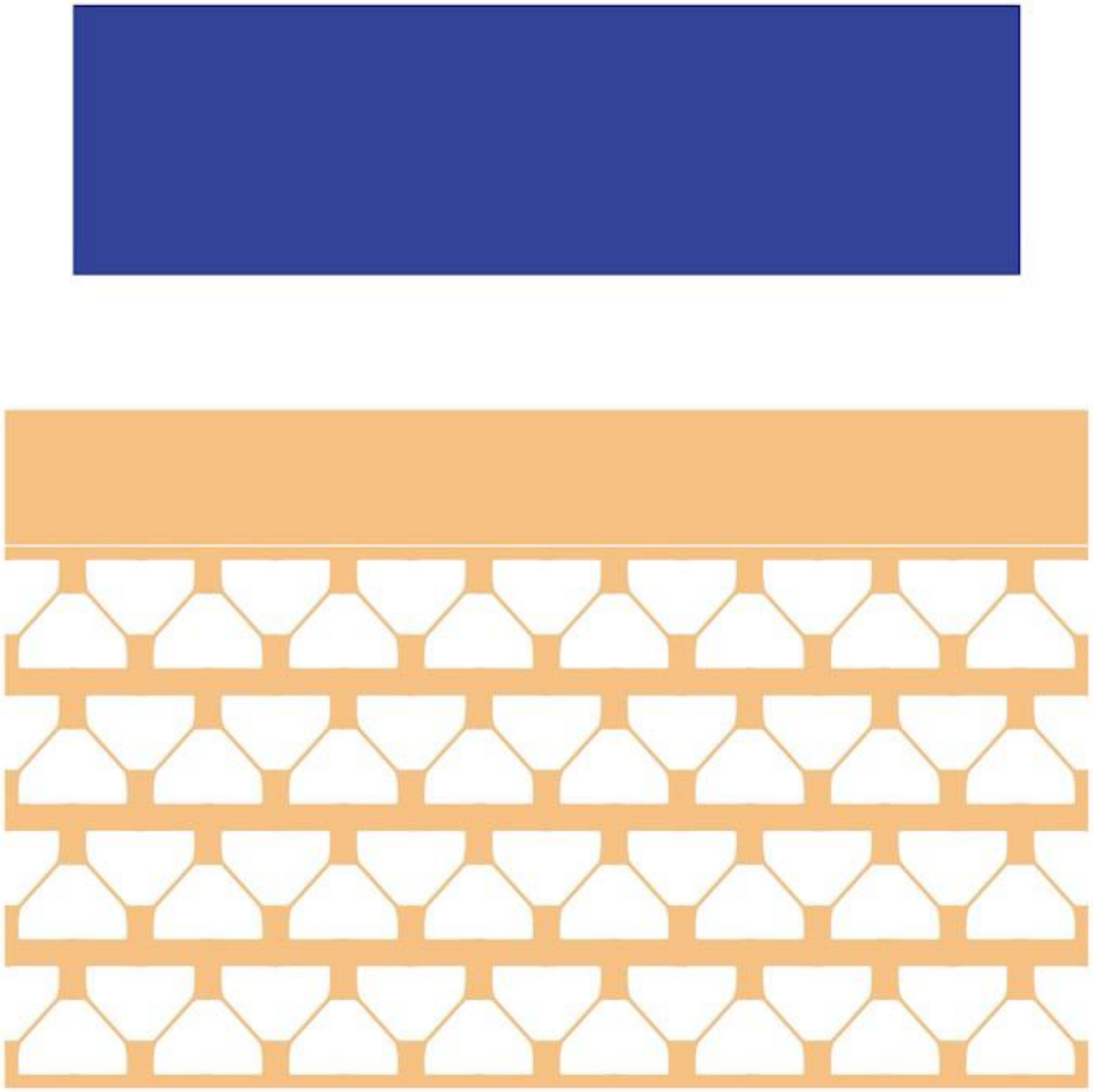
Davi Colli Tozoni, Jeremie Dumas, Zhongshi Jiang, Julian Panetta, Daniele Panozzo, Denis Zorin,

ACM Transaction on Graphics (SIGGRAPH), 2020

[Paper] [Code] [Supplemental] [Talk]

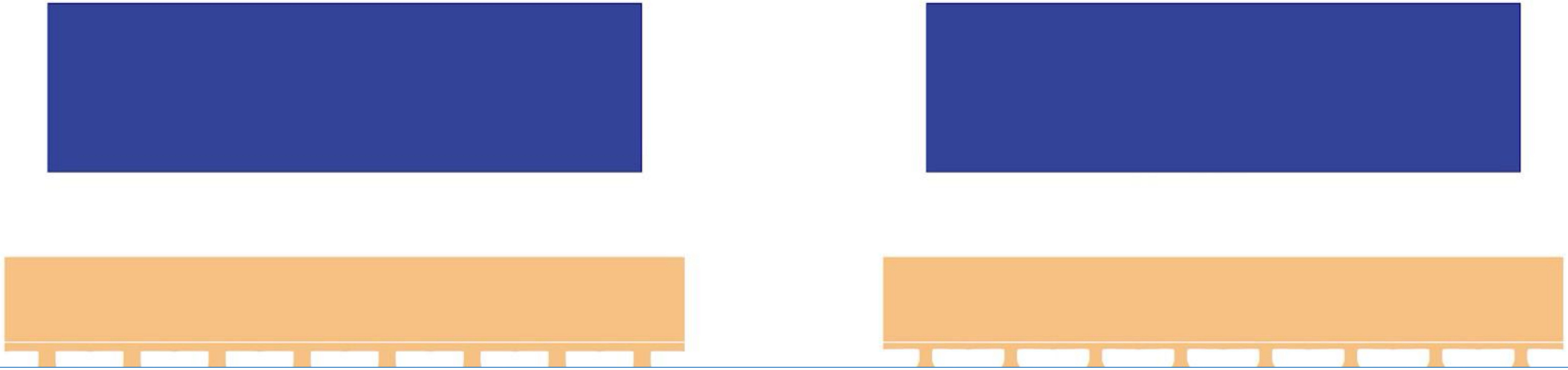


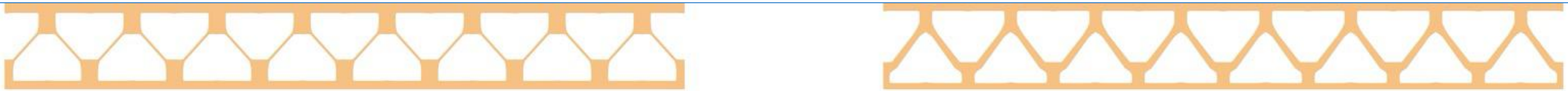
Elastodynamic and Contact



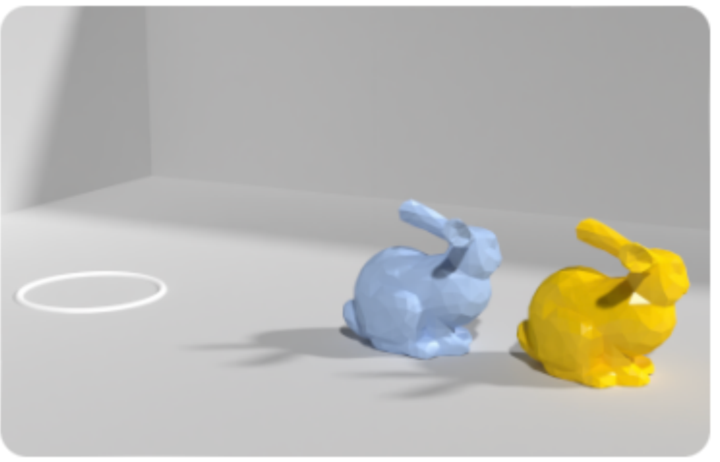
Initial

Elastodynamic and Contact





Initial Optimized



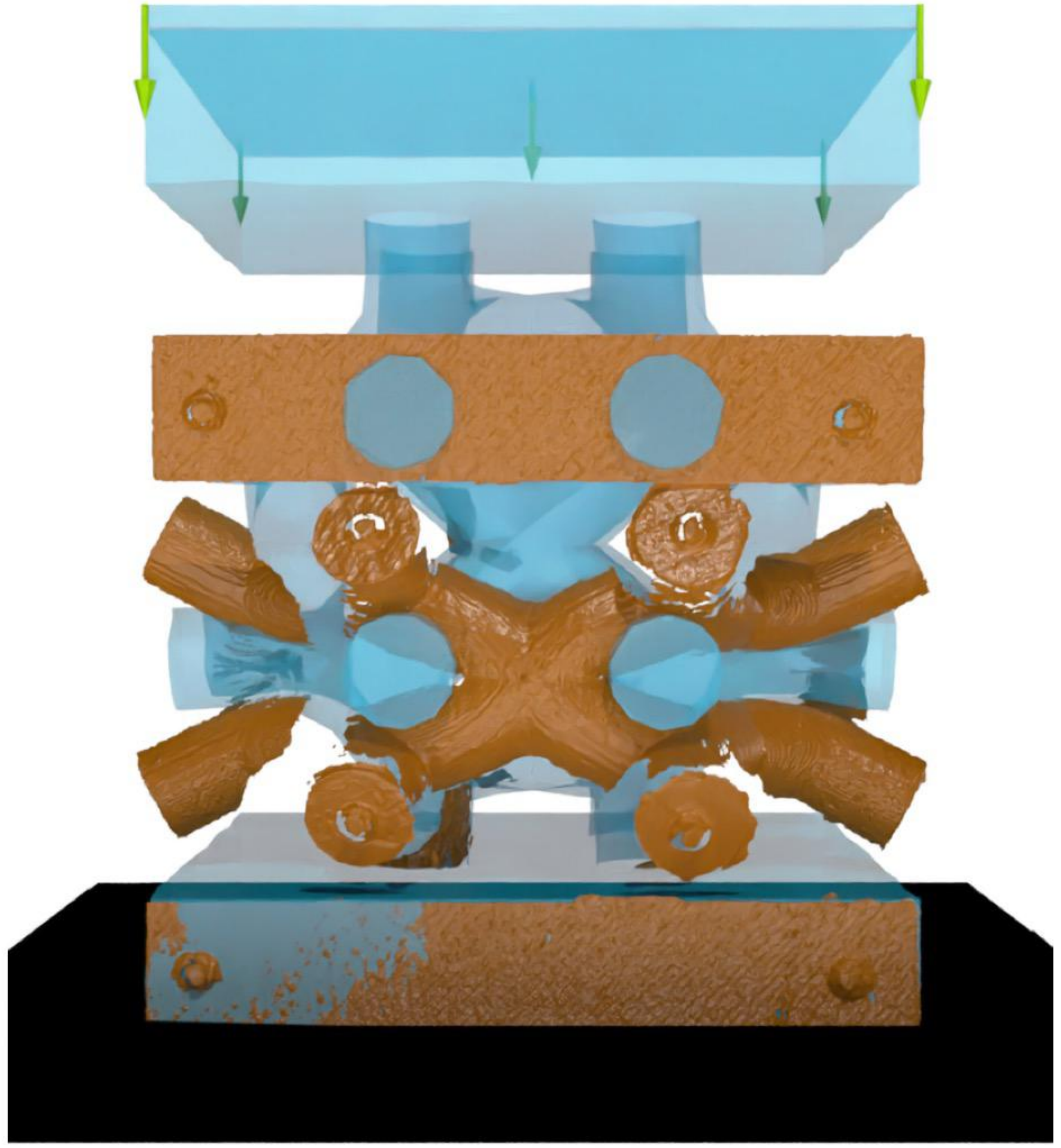
Differentiable solver for time-dependent deformation problems with contact

Zizhou Huang, Davi Colli Tozoni, Arvi Gjoka, Zachary Ferguson, Teseo Schneider, Daniele Panozzo, Denis Zorin,

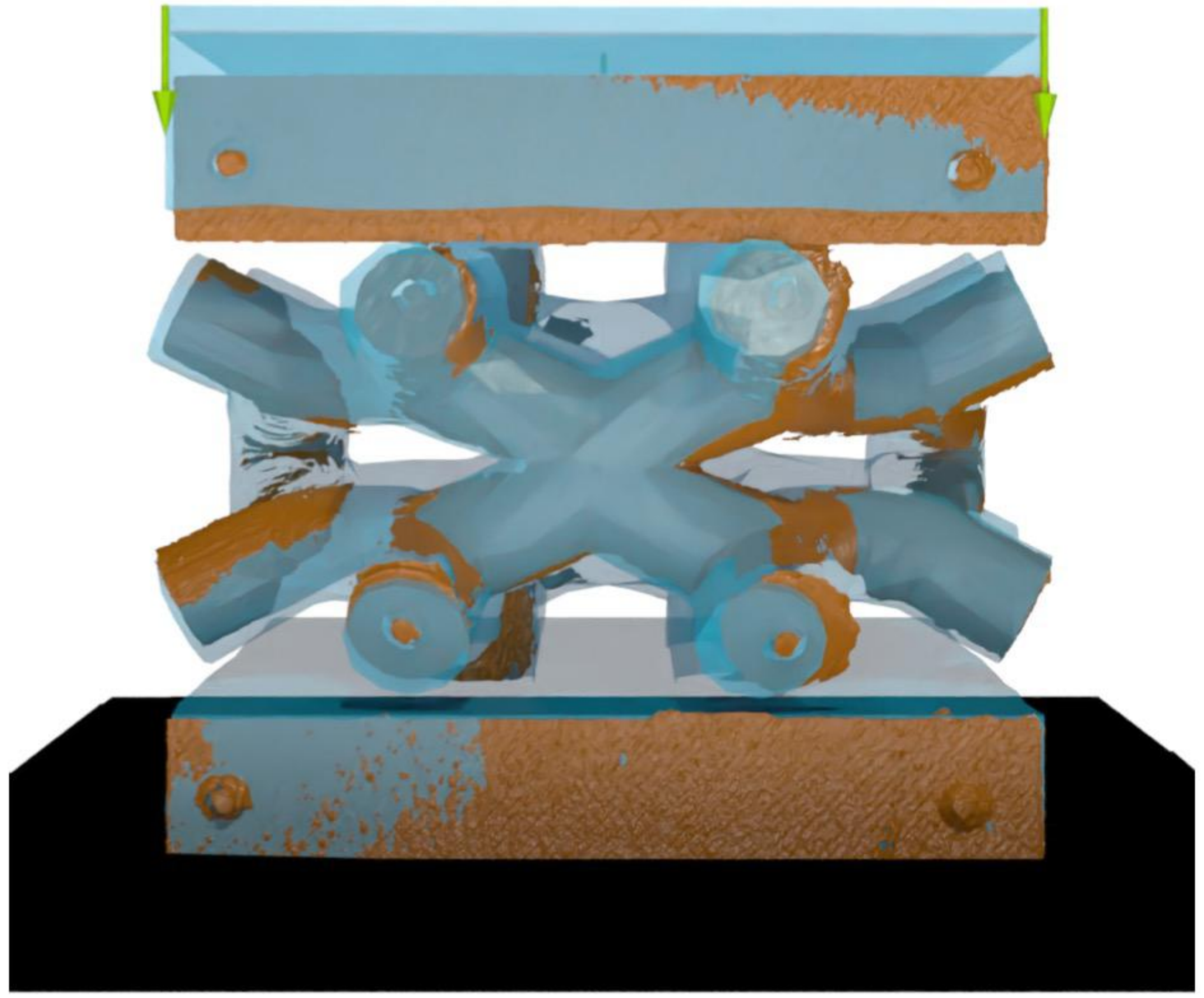
ACM Transaction on Graphics, 2024

[Paper] [Video] [Code]

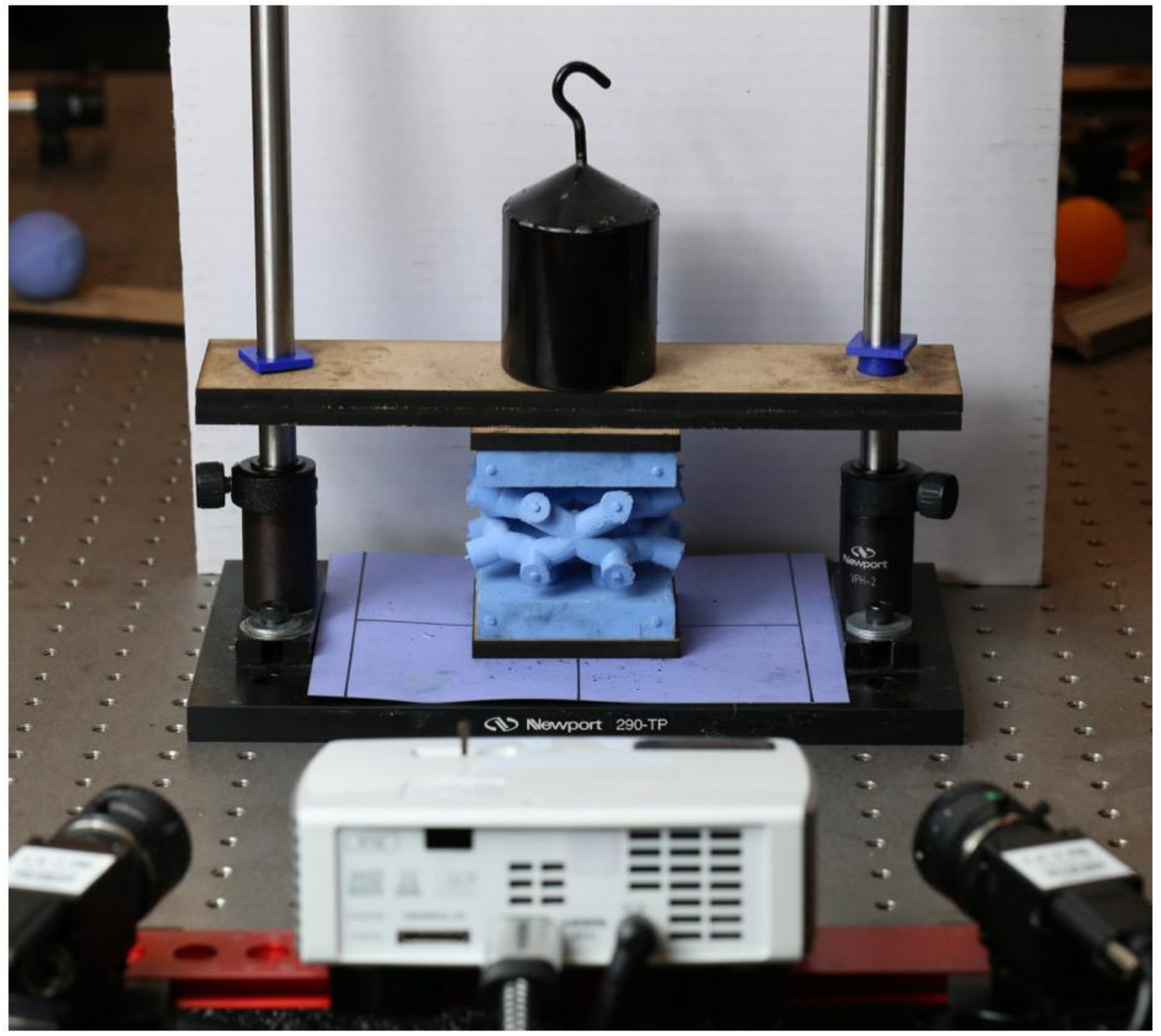
Indirect Material Testing



Initial.



Optimized.



Physical Experiment Setup.

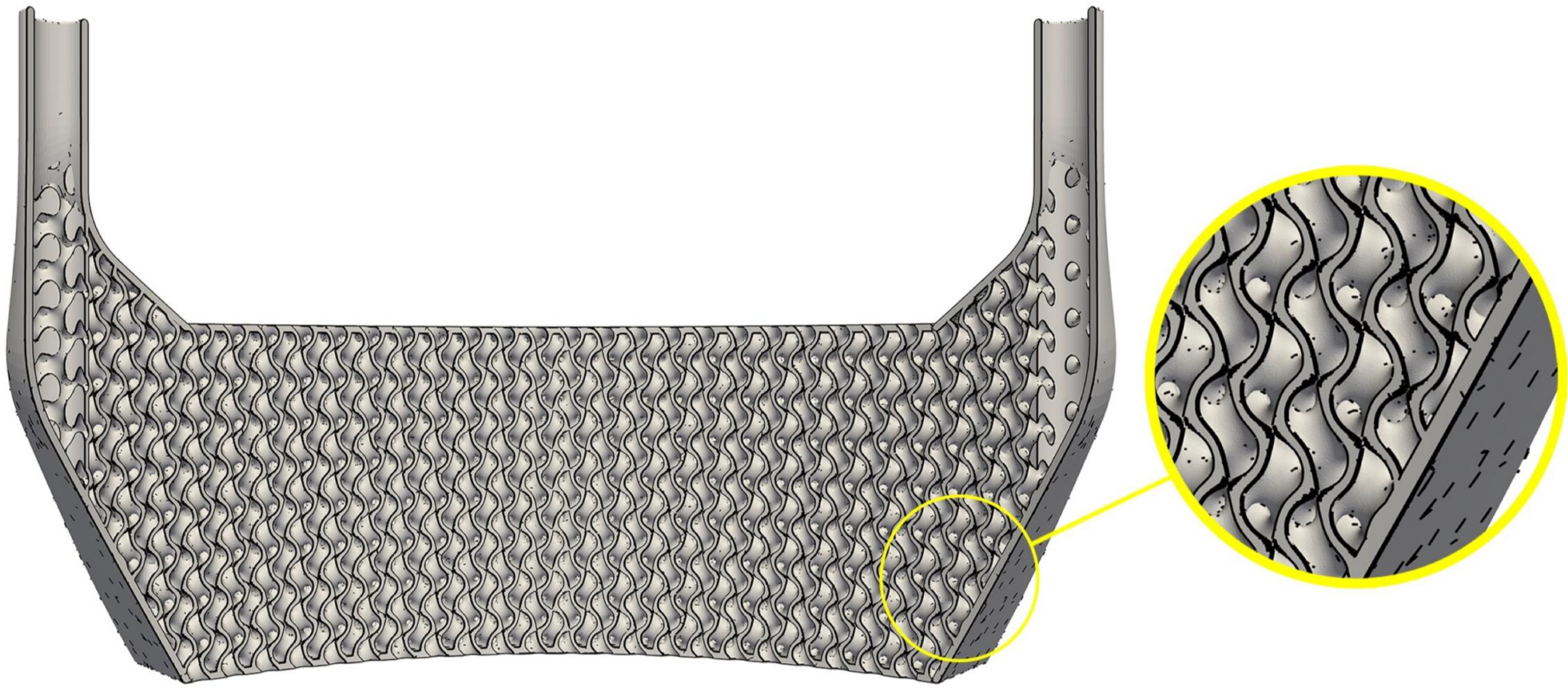
Automating Finite Element Analysis

Forward Simulation

Forward Simulation

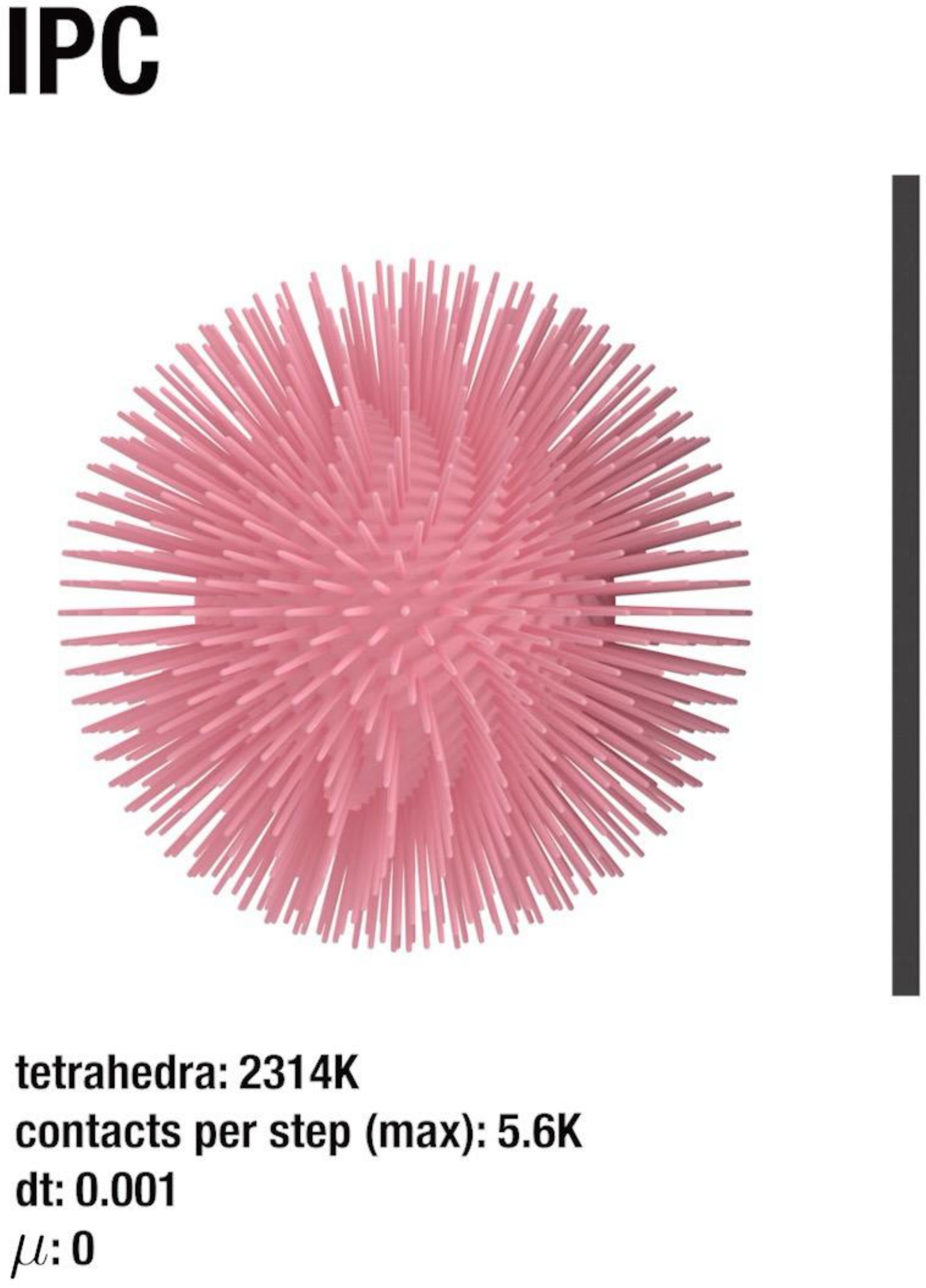
Complex Geometry

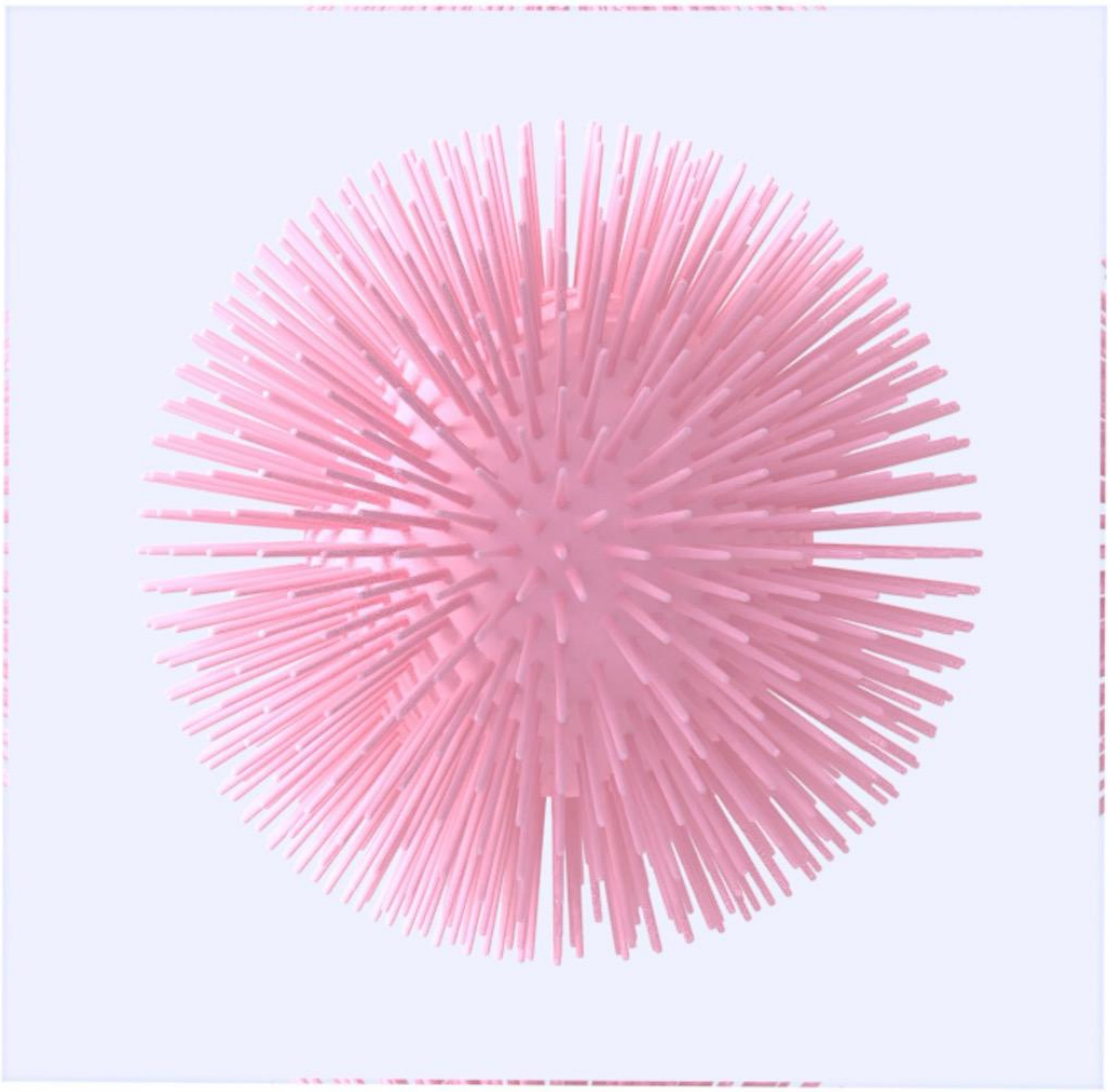
Volume Reduction



Heat Flux Increase

Complex Contacts





Goals

- Minimal/No requirements on input geometry, which can be implicit (SDF) or explicit (triangle meshes, CAD).
- Fully automatic, parameters should relate to accuracy of the solution or of the physical system (for example Young's modulus for elasticity), not algorithmic details.
- Minimize time, given an error bound.

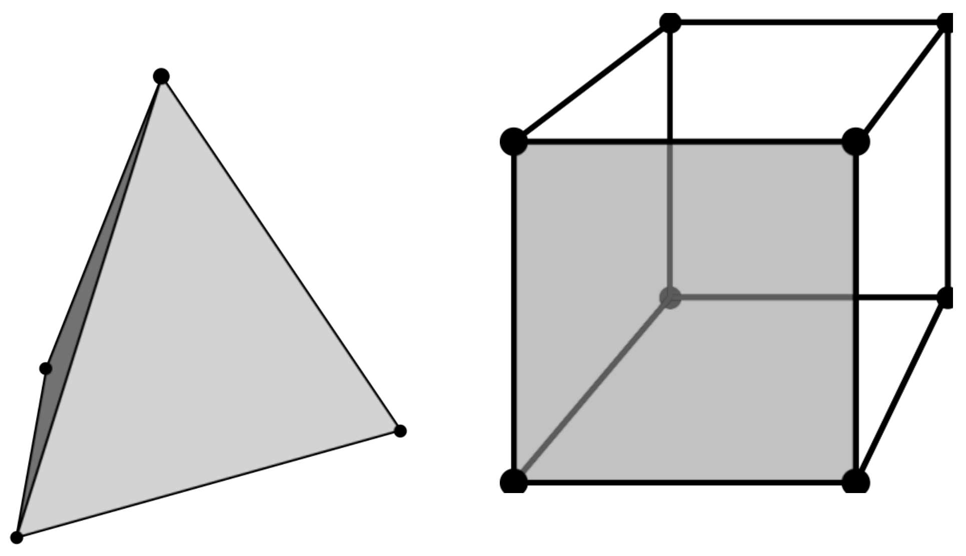
Observations

- It is a difficult problem, involving geometry, space discretization, solution of PDEs, contact detection, numerical methods, and physics. Evaluating individual components is insufficient and often misleading.

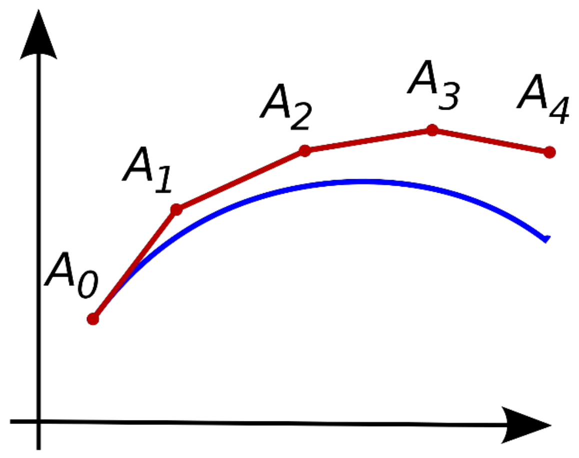
- I advocate for solving this problem holistically as a single challenge.

Research Overview

High Level Decisions

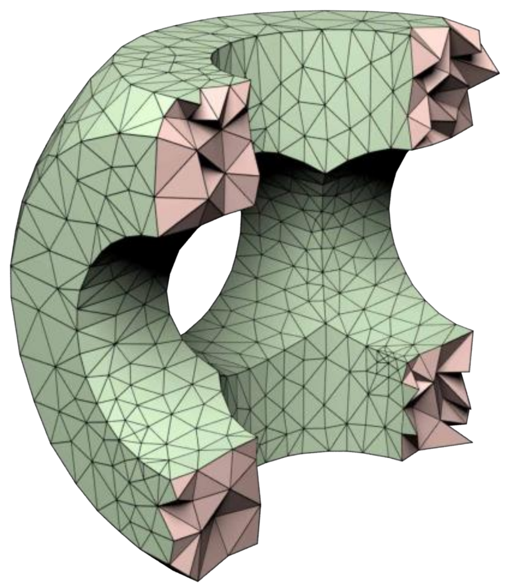


Which space discretization?

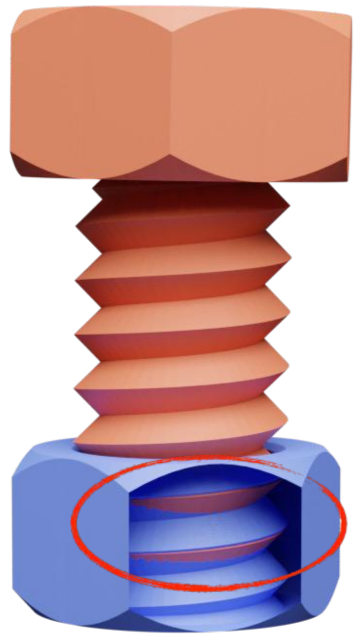


Which time integrator?

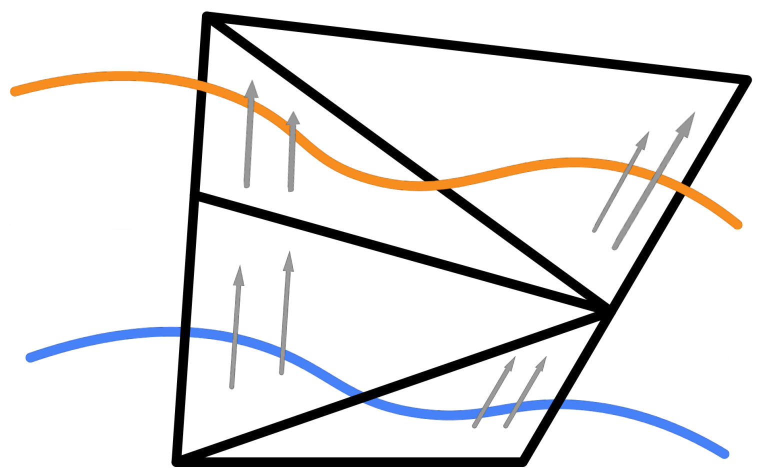
Realization



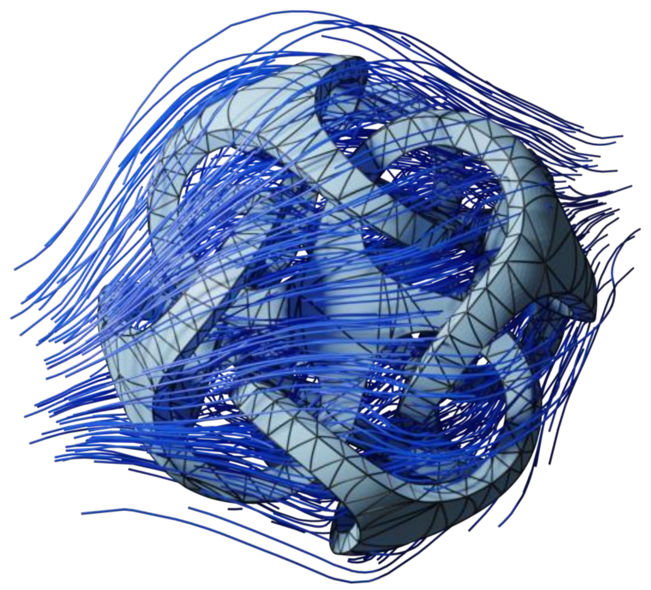
How to discretize space?



How to handle contacts?

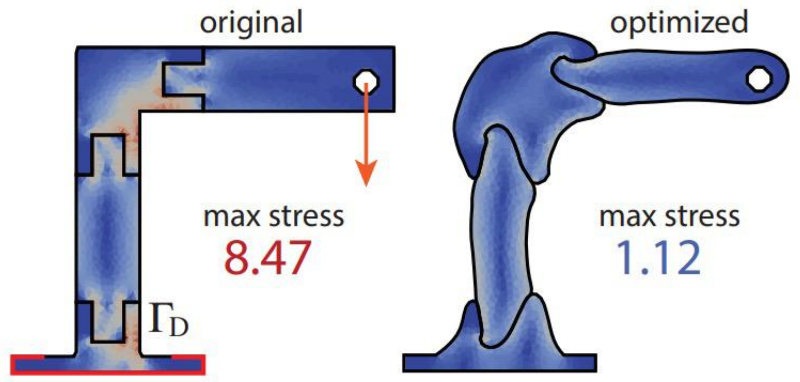


How to map input to discretization?

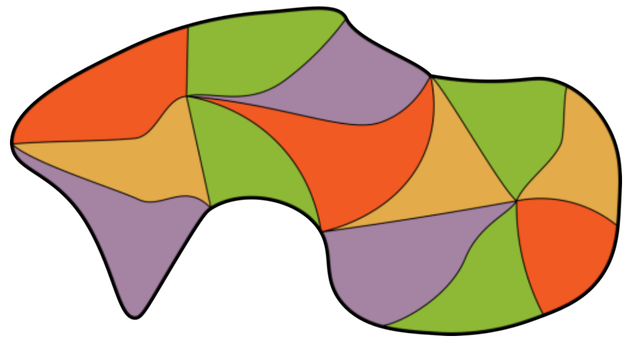


How to solve the PDE?

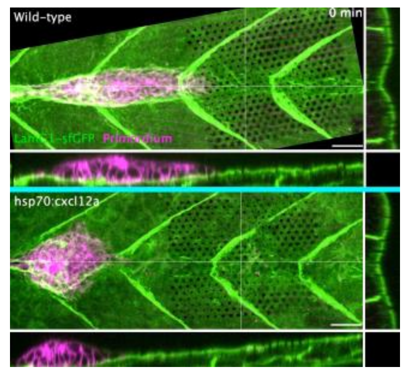
Applications/Improvement

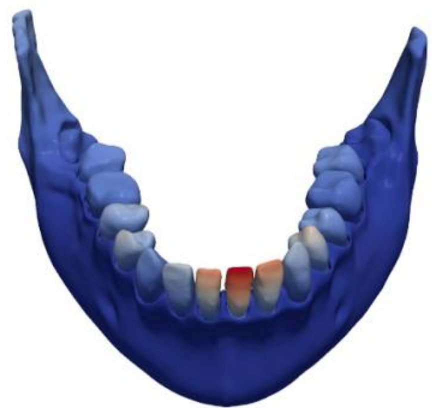


How to optimize functionals depending on the solution of a PDE?



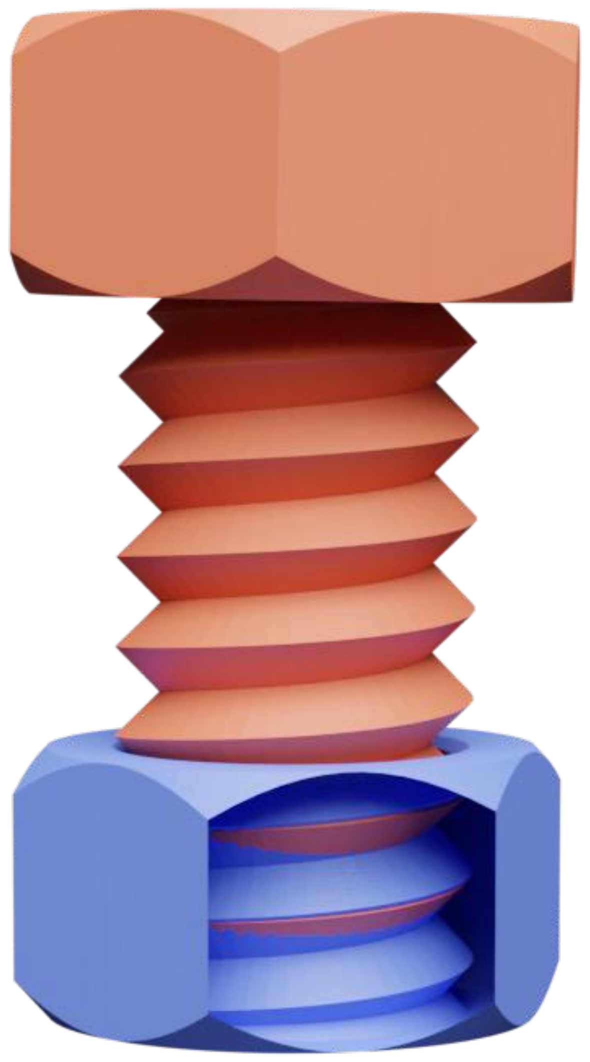
How to accelerate the system to make optimization possible?



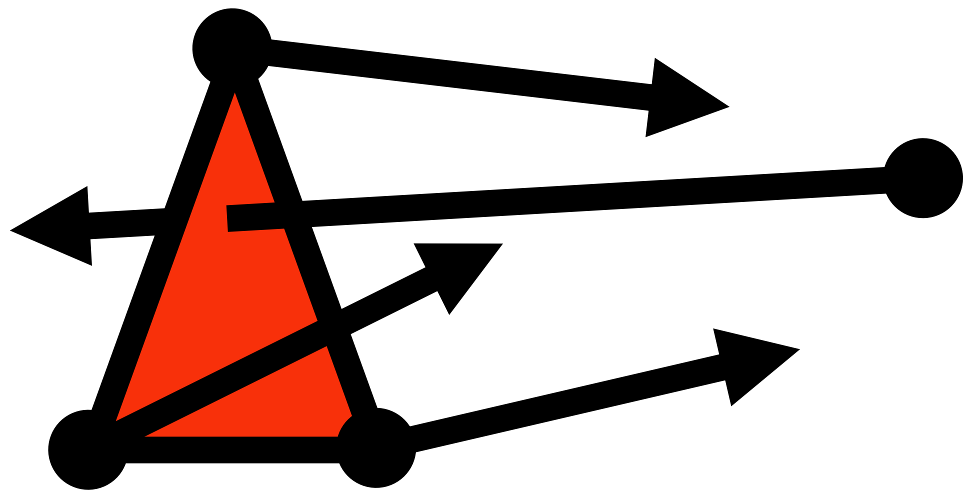


What can we do with it?

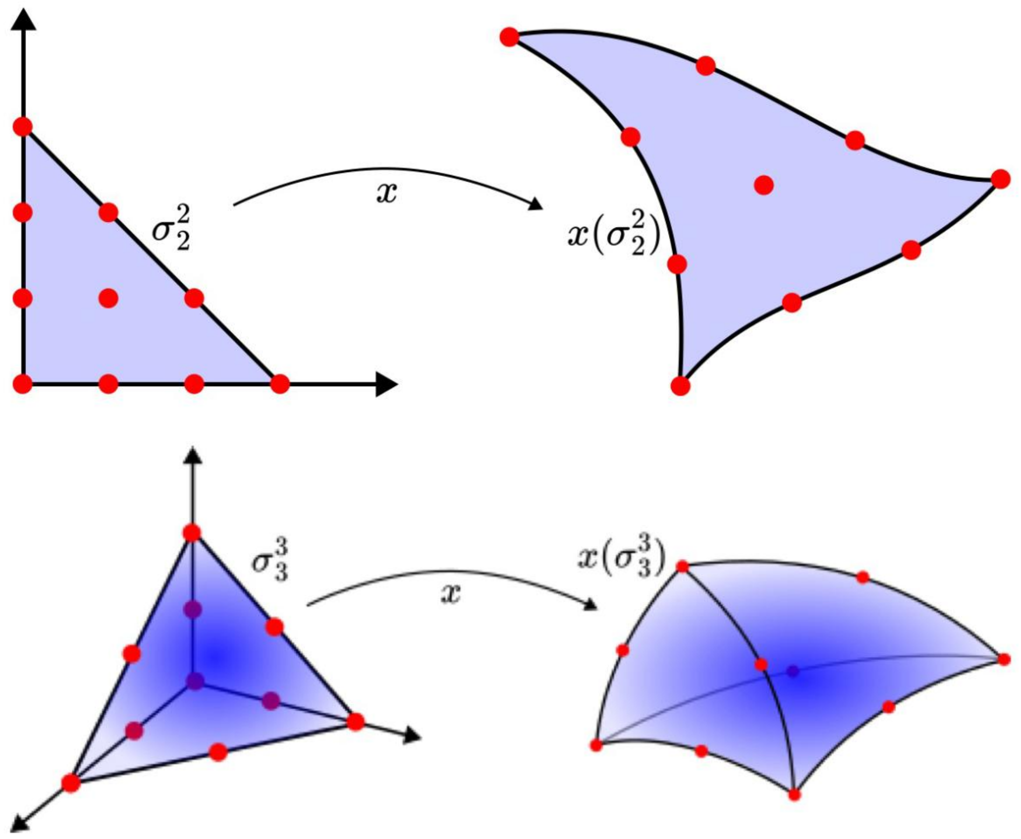
Talk Overview



Differentiable Contact Elastodynamics with an Incremental Potential Formulation

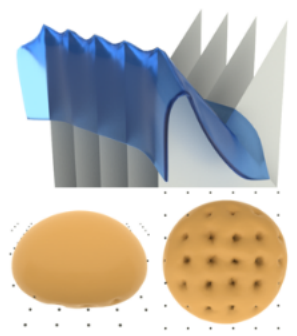


Provably Conservative Collision Avoidance



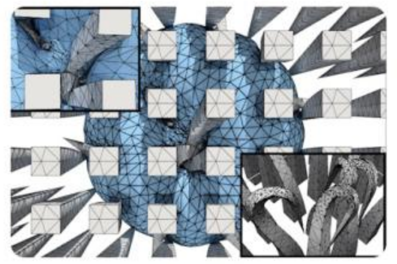
Provably Conservative Local Injectivity

Differentiable Contact Elastodynamics

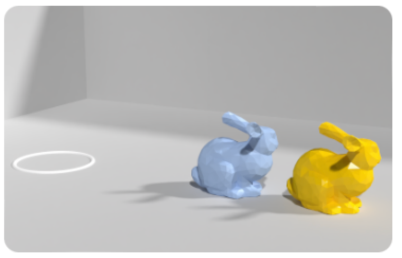


Incremental Potential Contact: Intersection- and Inversion-free, Large-Deformation Dynamics
Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, Danny M. Kaufman,
ACM Transaction on Graphics (SIGGRAPH), 2020
[Paper] [Video] [Supplemental Video] [Code] [S1: Technical] [S2: Comparisons] [S3: Statistics] [Website]

Intersection-free Rigid Body Dynamics
Zachary Ferguson, Minchen Li, Teseo Schneider, Francisca Gil Ureta, Timothy Langlois, Chenfanfu Jiang, Denis Zorin, Danny M. Kaufman, Daniele Panozzo,
ACM Transaction on Graphics (SIGGRAPH), 2021
[Paper] [Video 1] [Video 2] [Website]



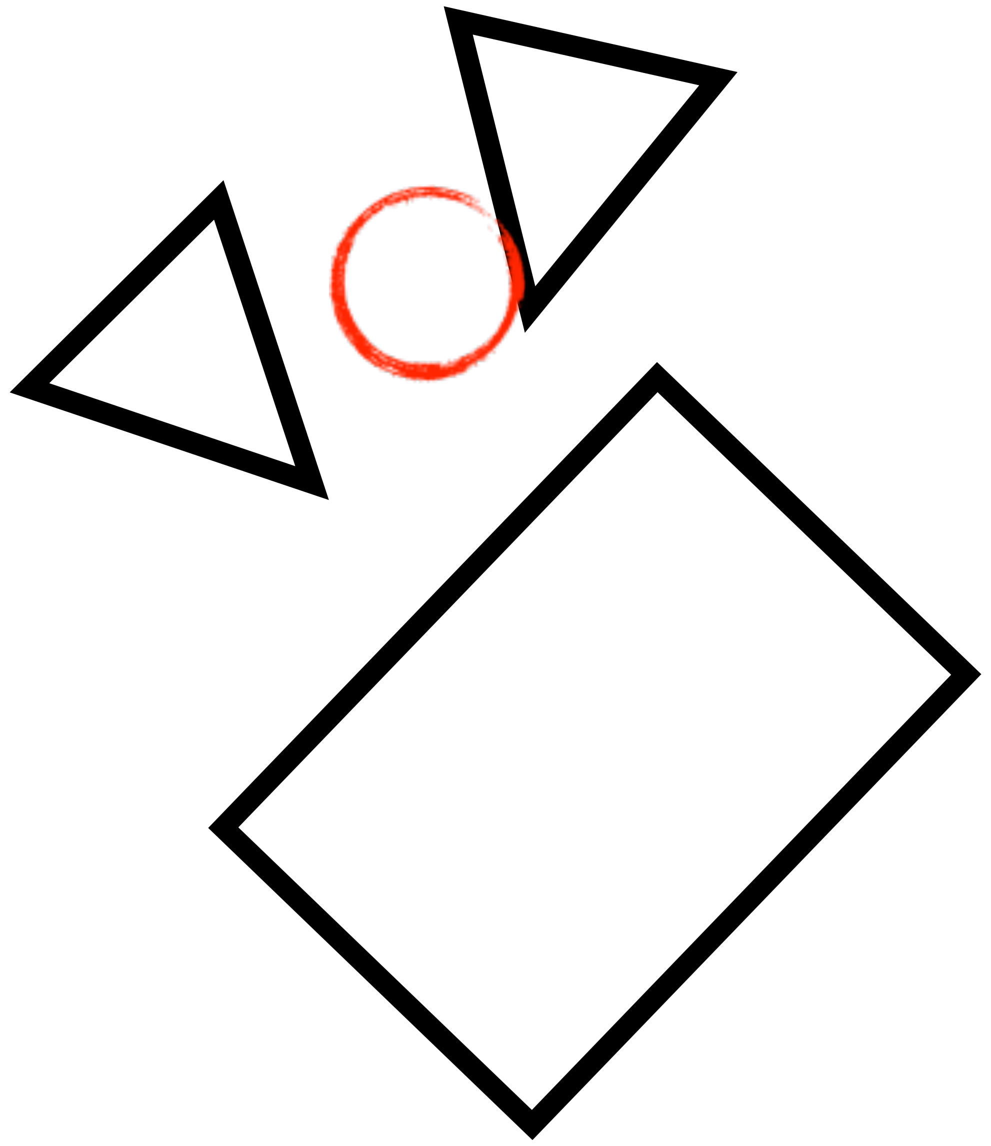
In-Timestep Remeshing for Contacting Elastodynamics
Zachary Ferguson, Teseo Schneider, Danny M. Kaufman, Daniele Panozzo,
ACM Transaction on Graphics (SIGGRAPH), (DK and DP are co-corresponding authors), 2023
[Paper] [Code (Coming Soon)]



Differentiable solver for time-dependent deformation problems with contact
Zizhou Huang, Davi Colli Tozoni, Arvi Gjoka, Zachary Ferguson, Teseo Schneider, Daniele Panozzo, Denis Zorin,
ACM Transaction on Graphics, 2024
[Paper] [Video] [Code]

Background

State of the Art

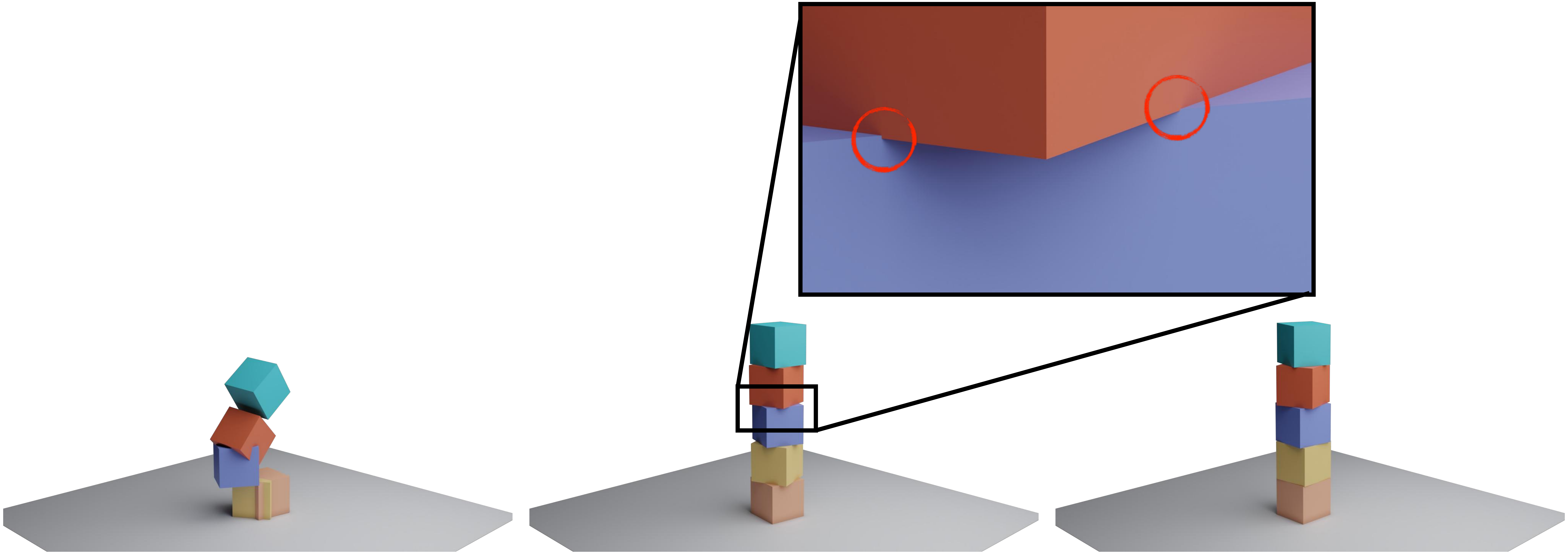


Large time steps

No guarantees!

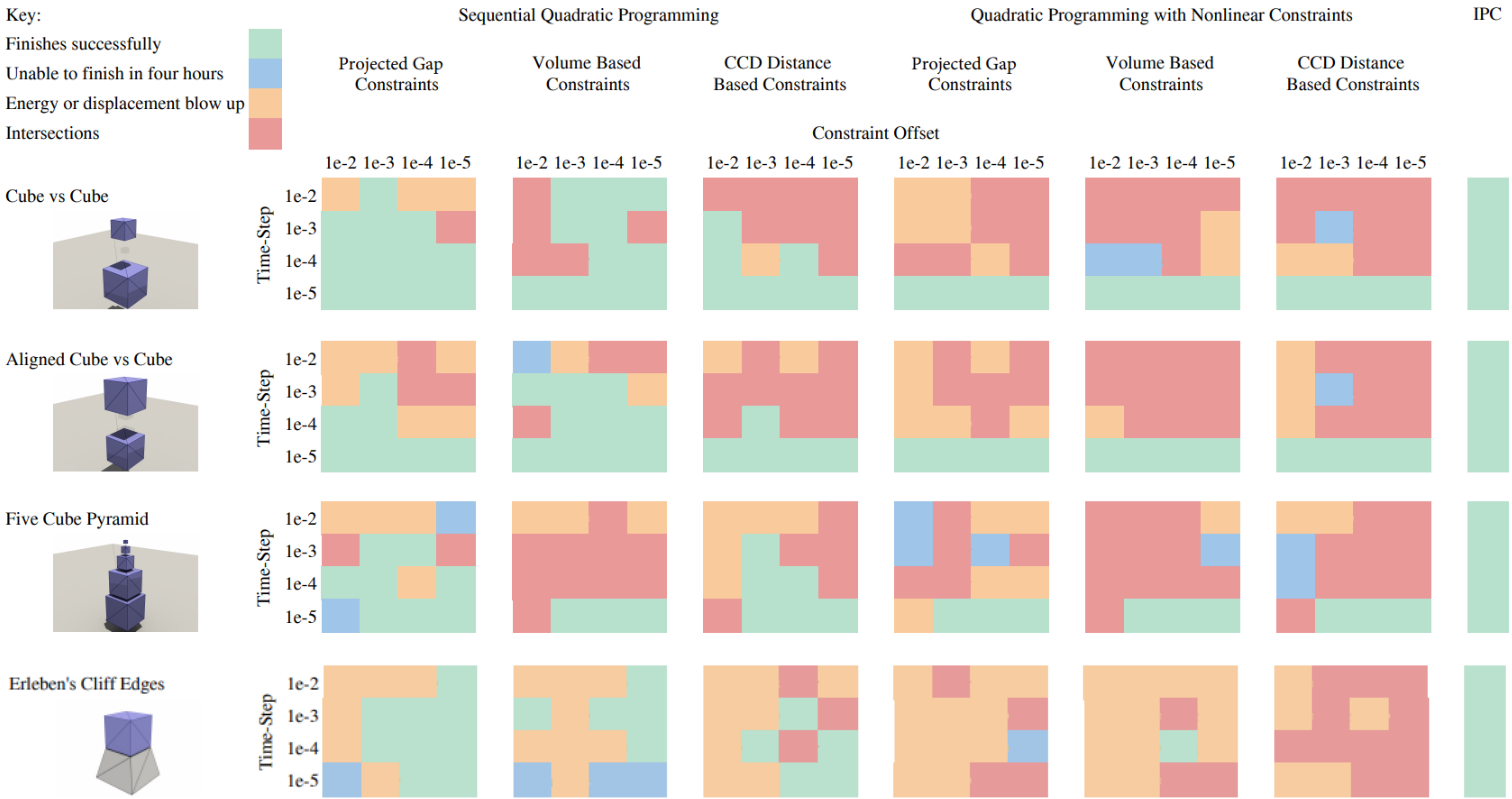
Daniele Panozzo

A solution might not exist!



$h = 0.01s$ $h = 0.001s$ $h = 0.0001s$

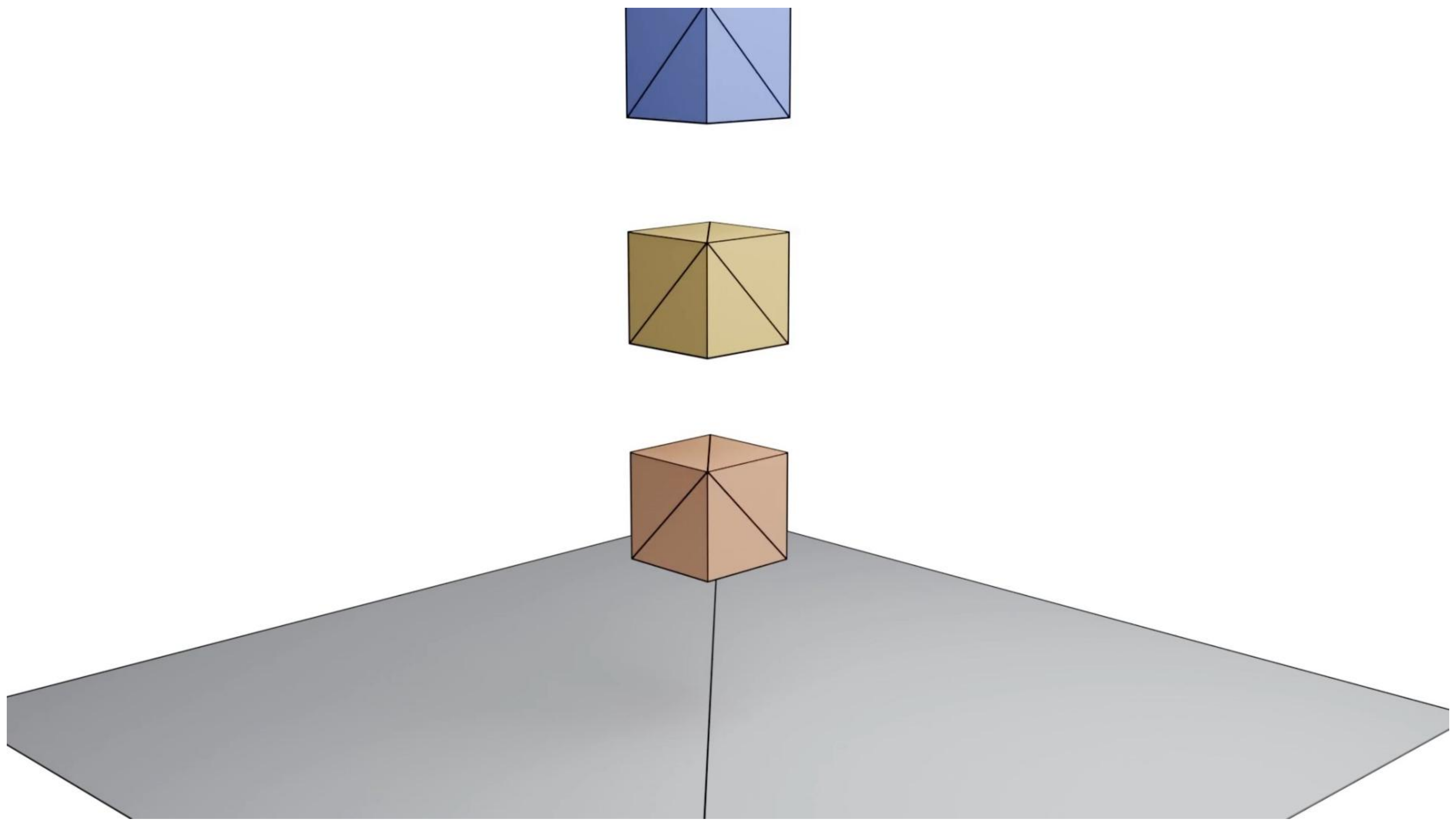
Simulated using Chrono [Tasora et al. 2016]

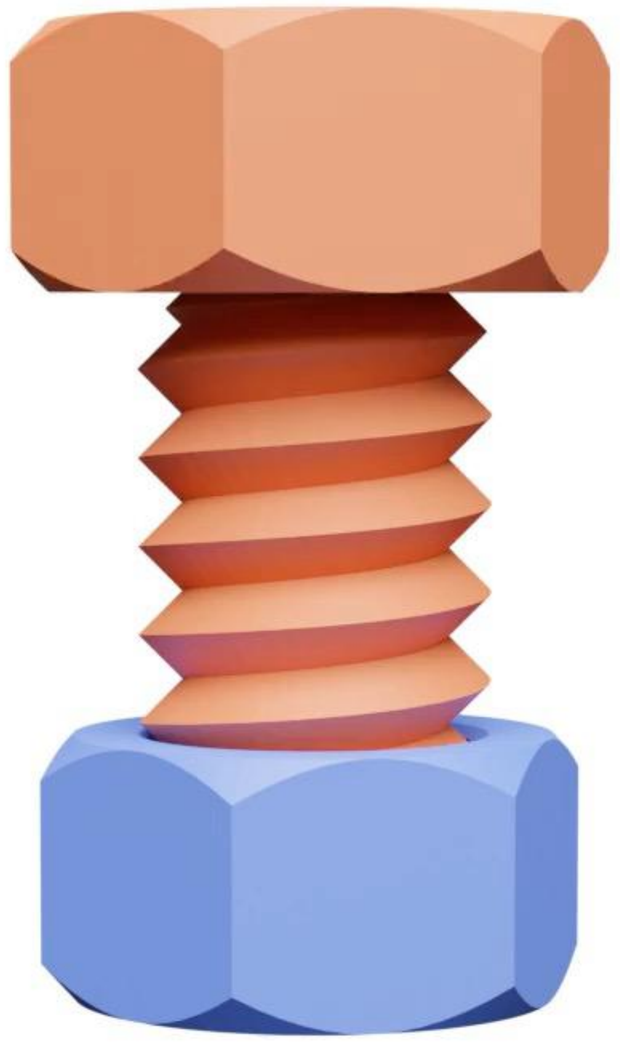


Daniele Panozzo

Goals

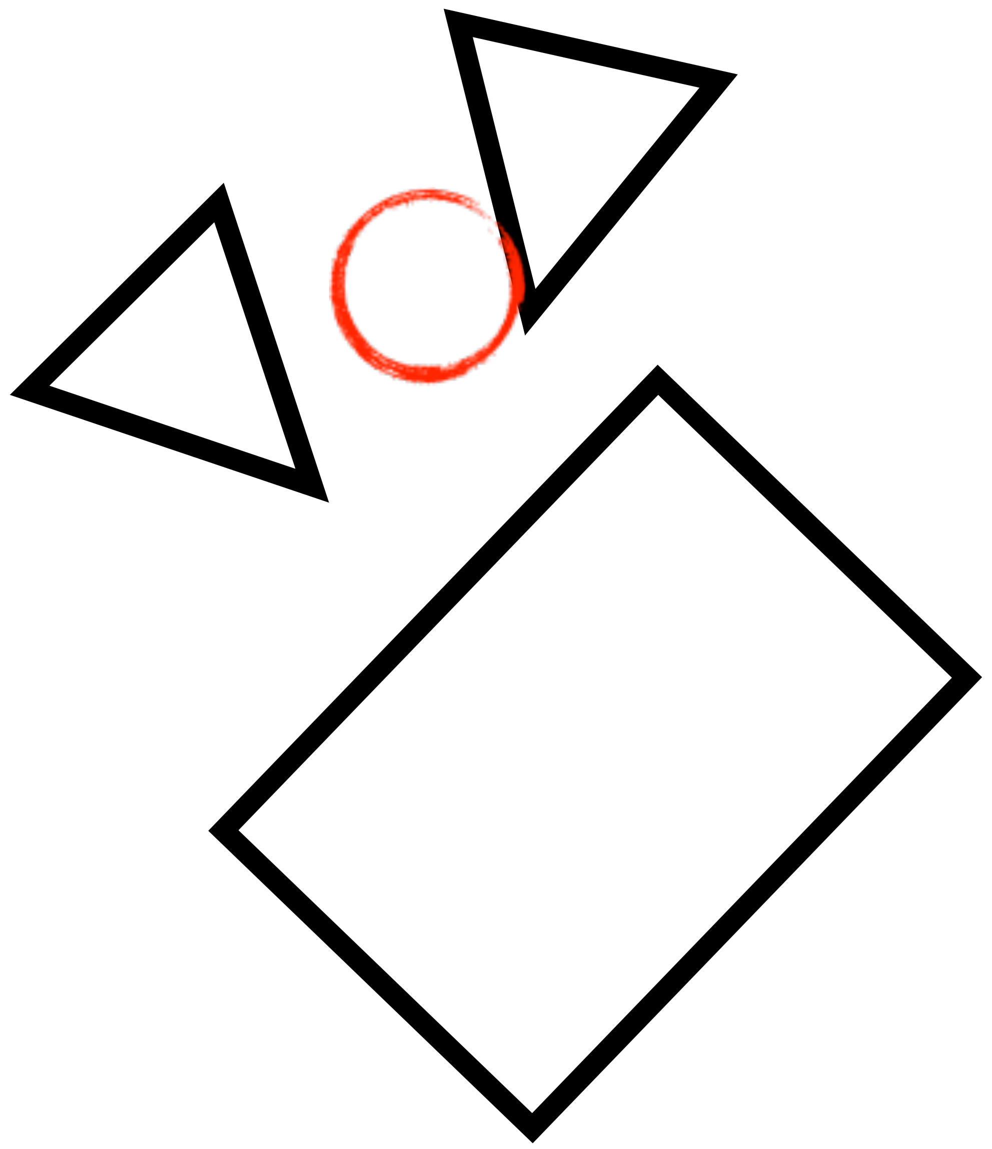
- Invariant 1: No intersection
- Invariant 2: Positive Volume
- Requirement: Differentiable





Solution

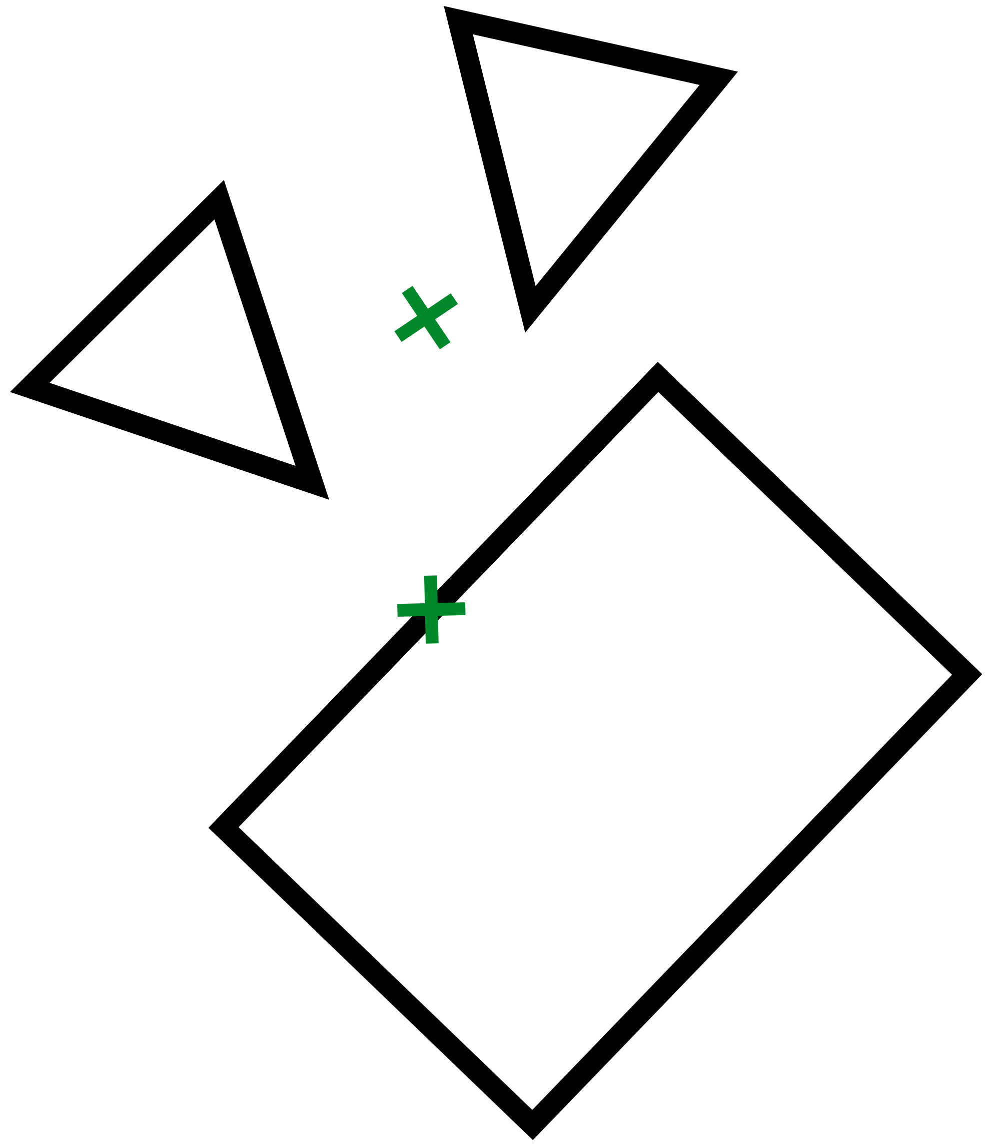
State of the Art



Large time steps

No guarantees!

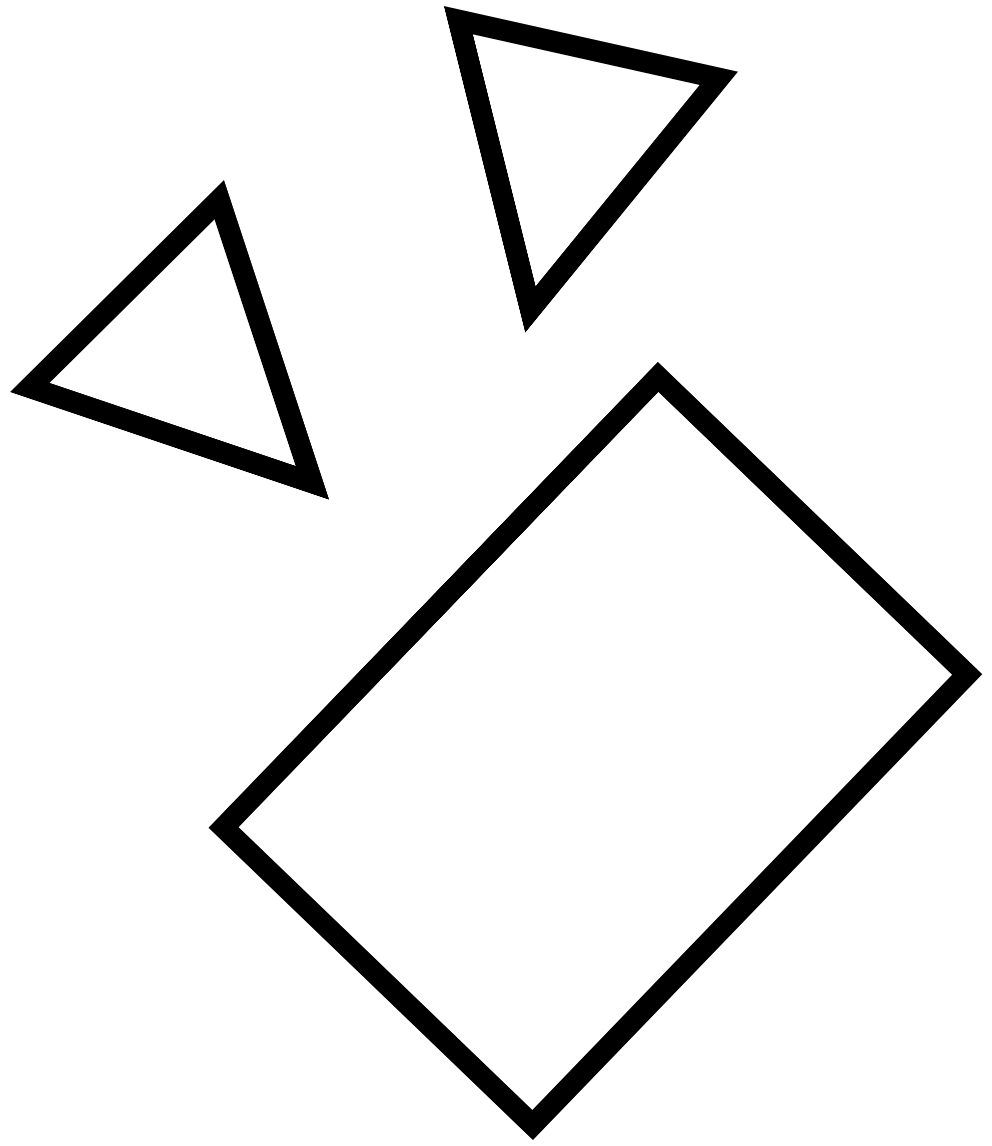
Explicit Method



Small time steps!

Guarantees

Our Approach/IPC



Large time steps

Guarantees

Incremental Potential

- Newton's equation of motion (x =vertex positions):

$$M\ddot{x} = f(x)$$

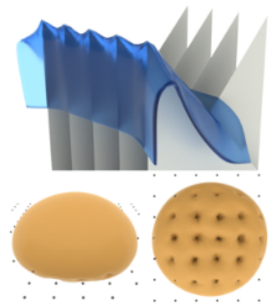
- Implicit time integration (e.g., implicit Euler):

$$x^{t+1} = x^t + h\dot{x}^t + h^2 M^{-1} f(x^{t+1})$$

- Optimization-based time integration [Kane et al. 2000]:

$$\hat{x} = x^t + h\dot{x}^t$$

$$x^{t+1} = \underset{x}{\operatorname{argmin}} \left(\underbrace{E(x) = \frac{1}{2}(x - \hat{x})^T M (x - \hat{x}) - h^2 x^T f(x)}_{\text{Incremental Potential (IP)}} \right)$$

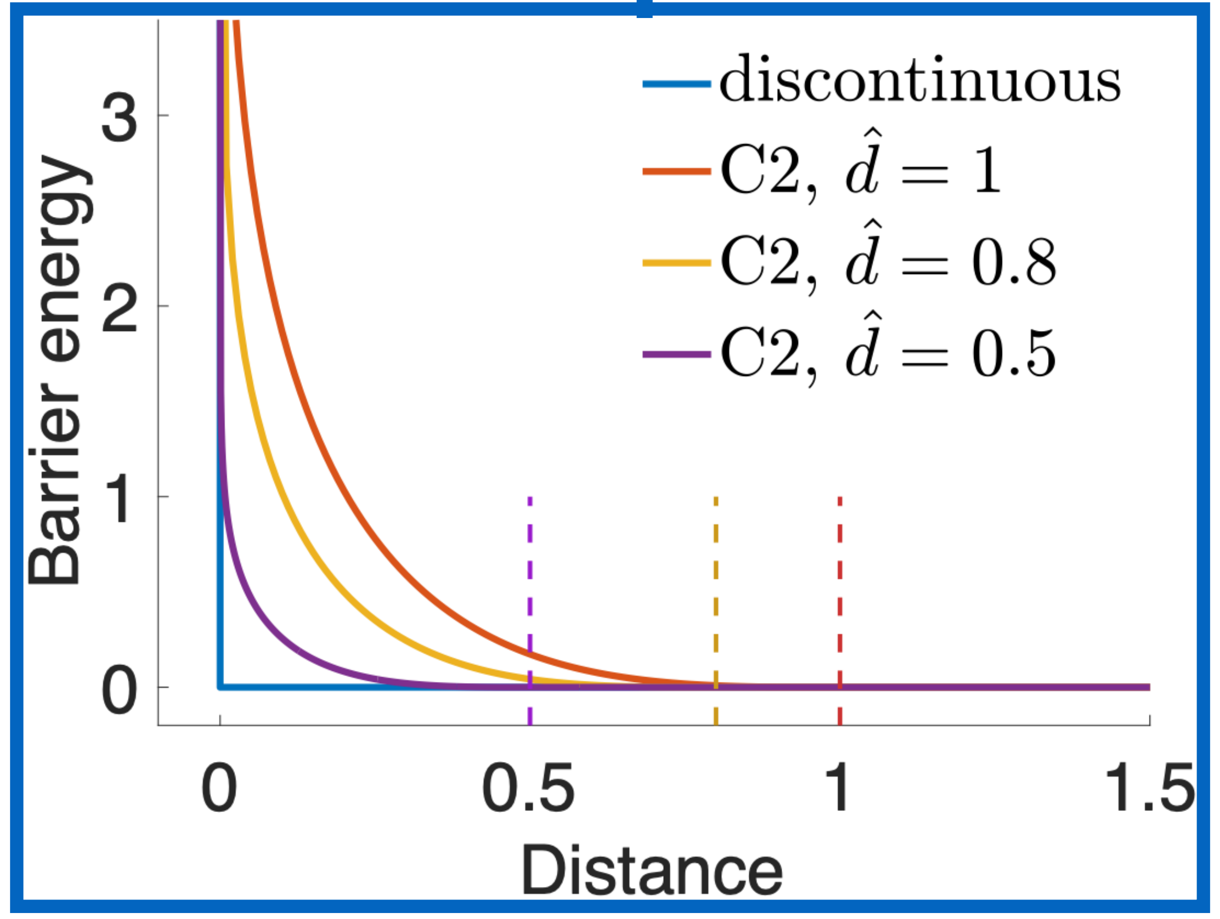


Incremental Potential Contact: Intersection- and Inversion-free, Large-Deformation Dynamics
Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, Danny M. Kaufman,
ACM Transaction on Graphics (SIGGRAPH), 2020
[Paper] [Video] [Supplemental Video] [Code] [S1: Technical] [S2: Comparisons] [S3: Statistics] [Website]

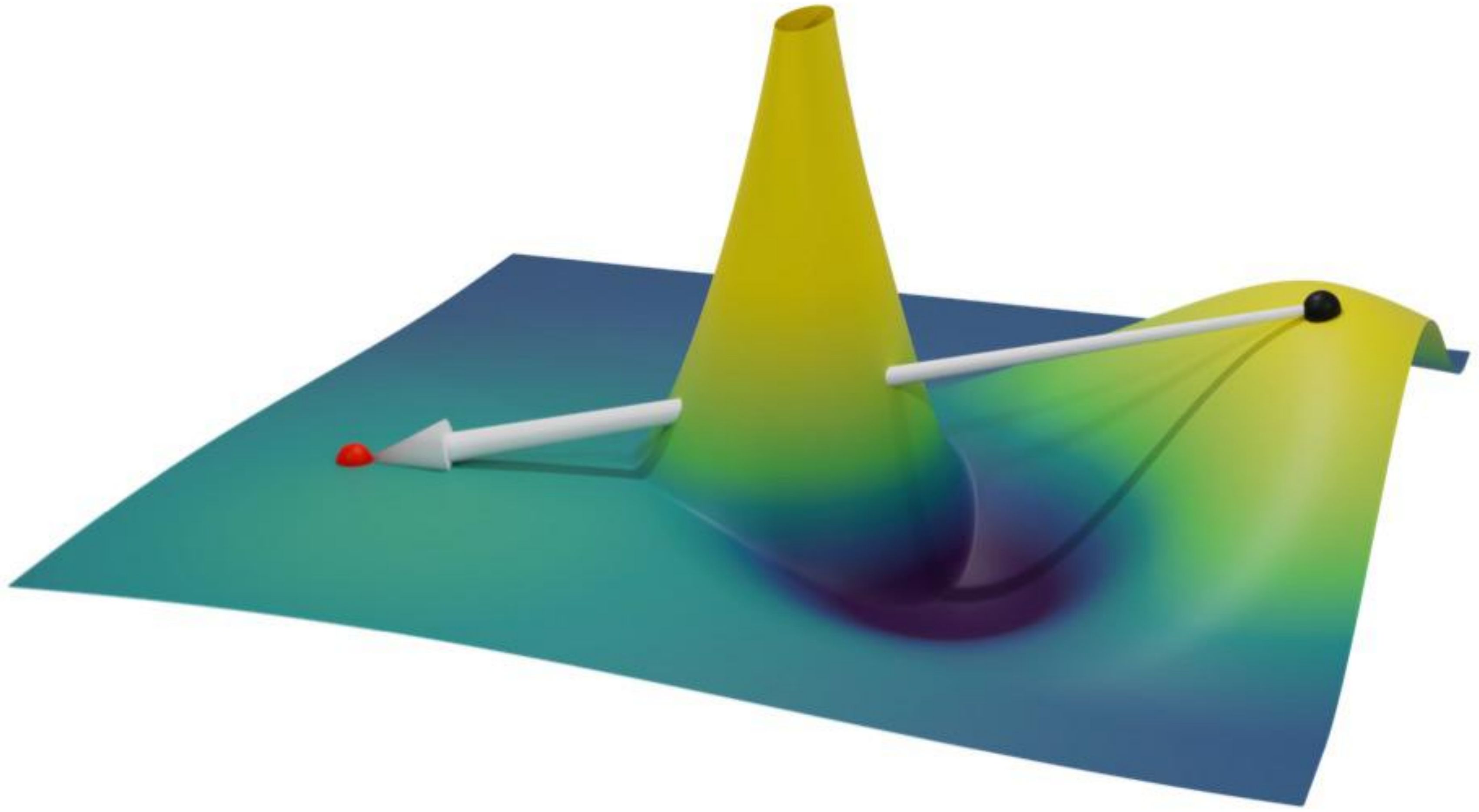
Contact and Friction* Potentials

$$x^{t+1} = \underset{x}{\operatorname{argmin}}\, E(x) + B(x, \hat{d}) + D(x)$$

Inertia & Elasticity



Friction*



Minimize using Newton's method, with line search to ensure lack of constraints violation

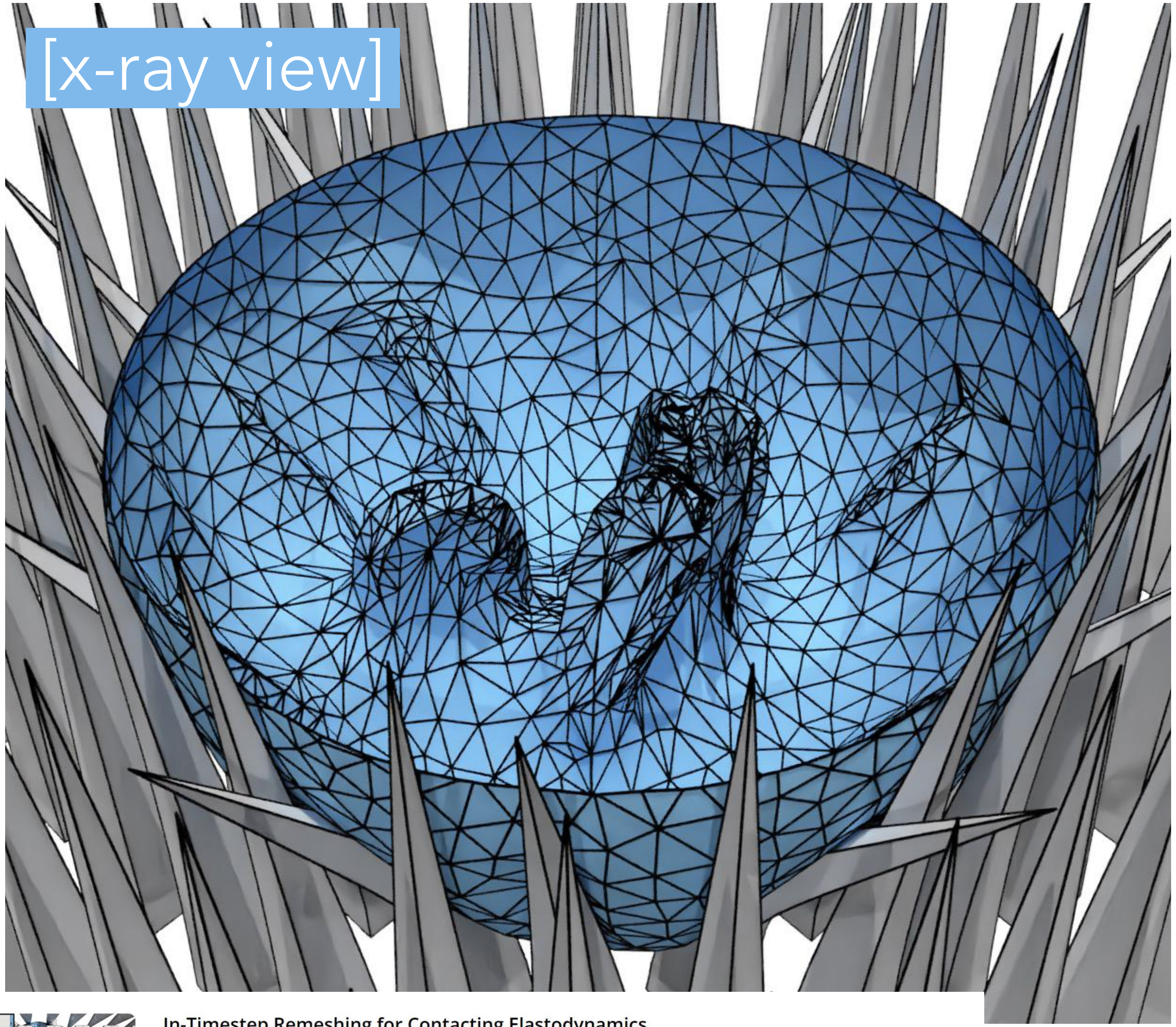
In-Timestep Remeshing for Contacting Elastodynamics

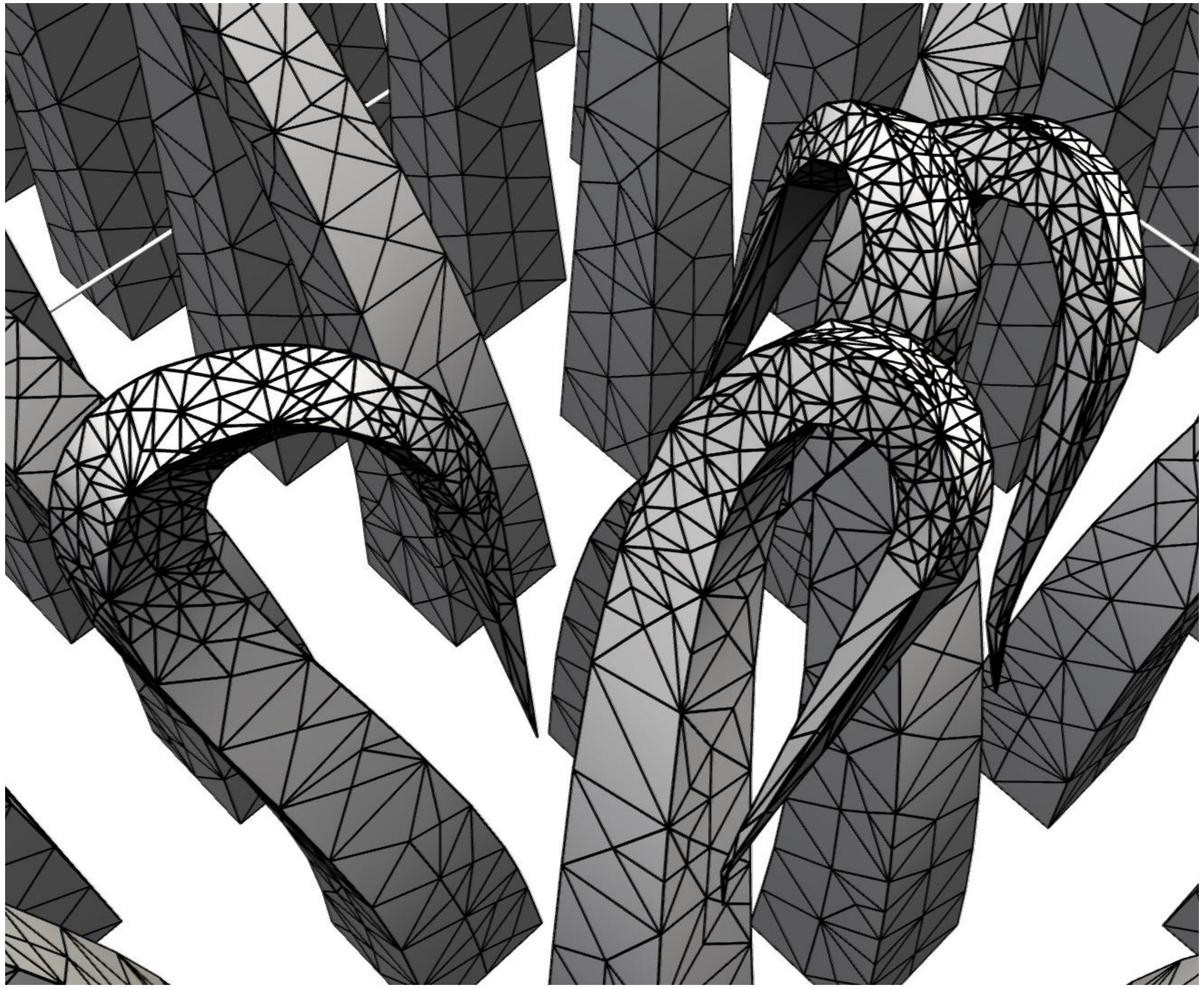
Zachary Ferguson, Teseo Schneider, Danny M. Kaufman, Daniele Panozzo,

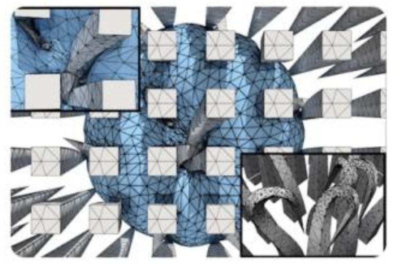
ACM Transaction on Graphics (SIGGRAPH), (DK and DP are co-corresponding authors), 2023

[Paper] [Code (Coming Soon)]

Our Adaptive Meshing

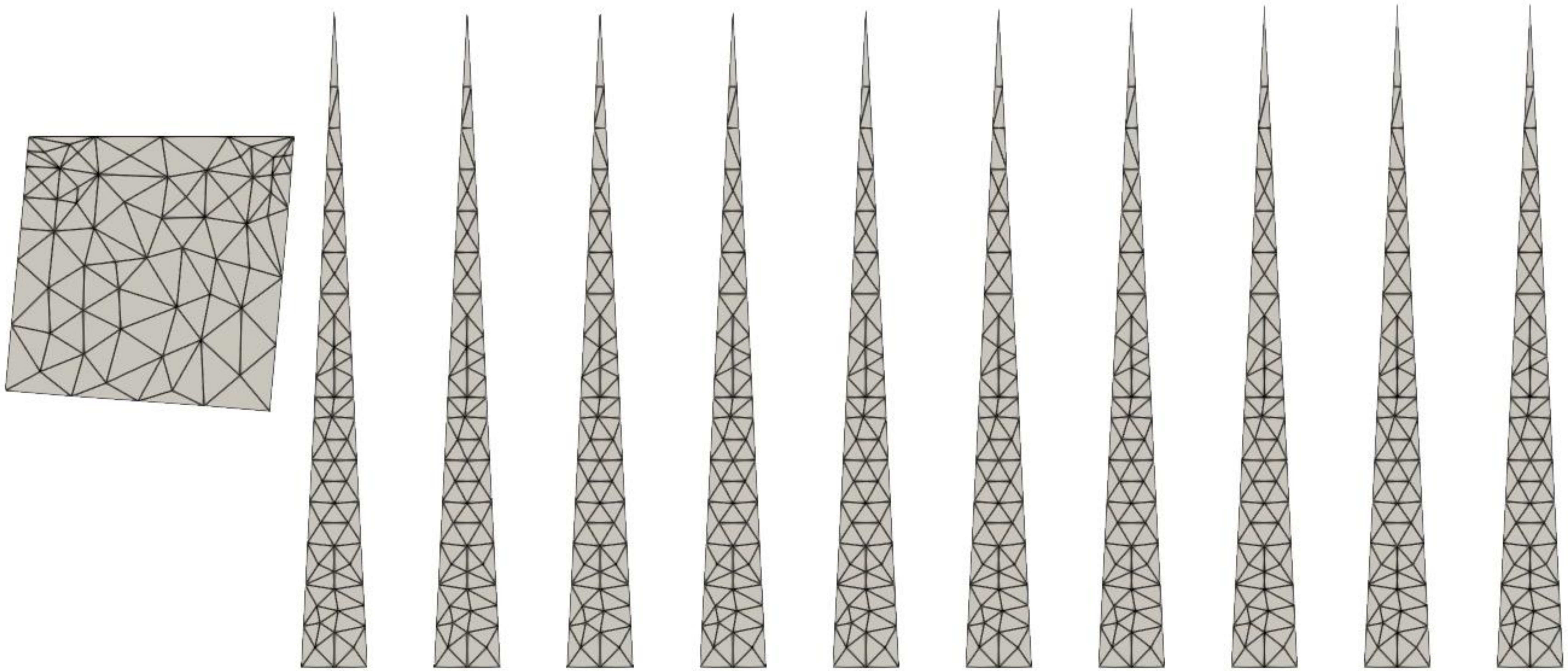


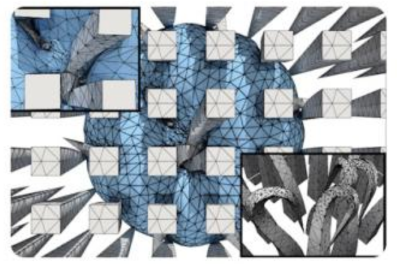




In-Timestep Remeshing for Contacting Elastodynamics
Zachary Ferguson, Teseo Schneider, Danny M. Kaufman, Daniele Panozzo,
ACM Transaction on Graphics (SIGGRAPH), (DK and DP are co-corresponding authors), 2023
[Paper] [Code (Coming Soon)]

dt: 0.01s





In-Timestep Remeshing for Contacting Elastodynamics

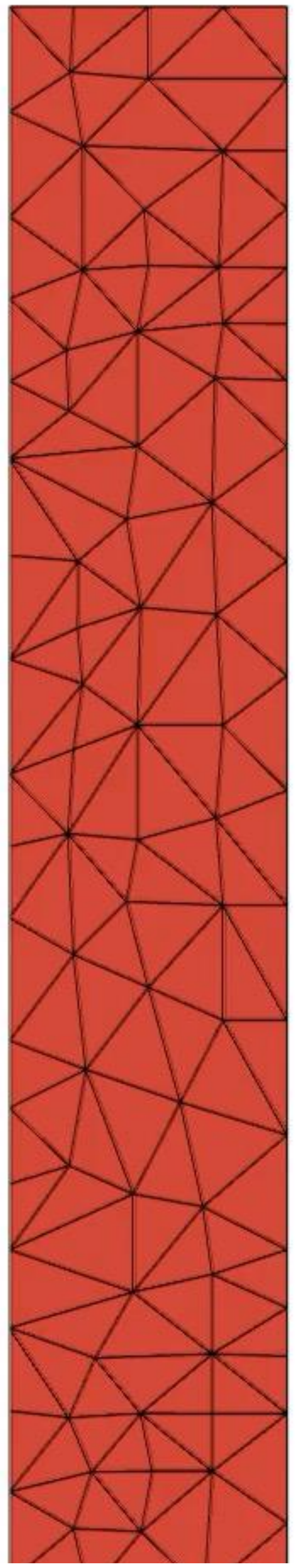
Zachary Ferguson, Teseo Schneider, Danny M. Kaufman, Daniele Panozzo,

ACM Transaction on Graphics (SIGGRAPH), (DK and DP are co-corresponding authors), 2023

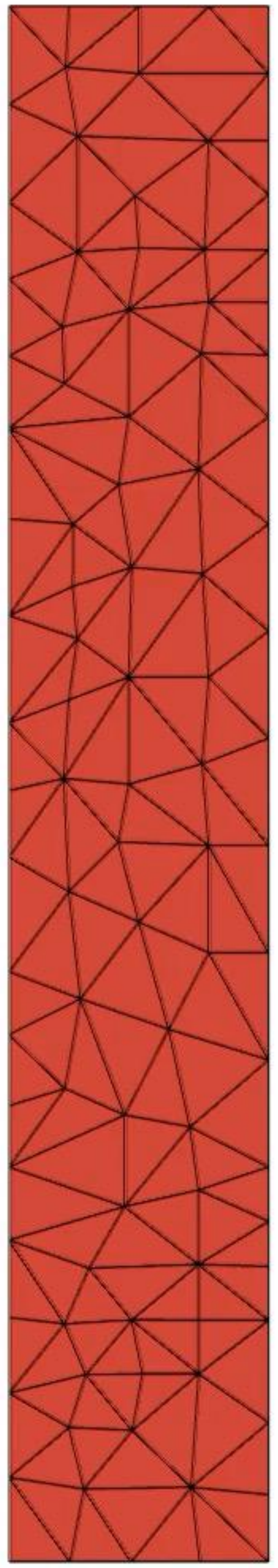
[Paper] [Code (Coming Soon)]

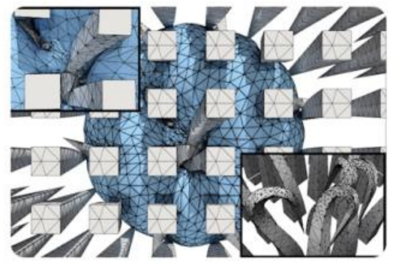
Bar twisting

No Remeshing



Ours





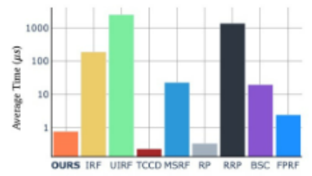
In-Timestep Remeshing for Contacting Elastodynamics

Zachary Ferguson, Teseo Schneider, Danny M. Kaufman, Daniele Panozzo,

ACM Transaction on Graphics (SIGGRAPH), (DK and DP are co-corresponding authors), 2023

[Paper] [Code (Coming Soon)]

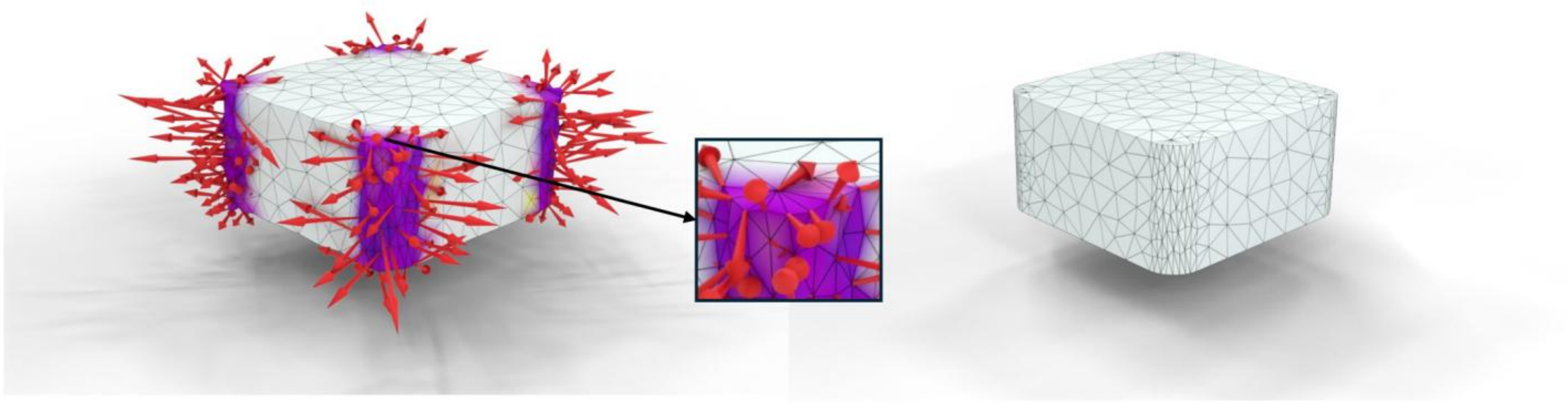
Collision Detection



A Large Scale Benchmark and an Inclusion-Based Algorithm for Continuous Collision Detection
Bolun Wang, Zachary Ferguson, Teseo Schneider, Xin Jiang, Marco Attene, Daniele Panozzo,
ACM Transaction on Graphics, 2021
[Paper] [Video] [CCD Code] [Benchmark Code]

Orientation-aware Incremental Potential Contact

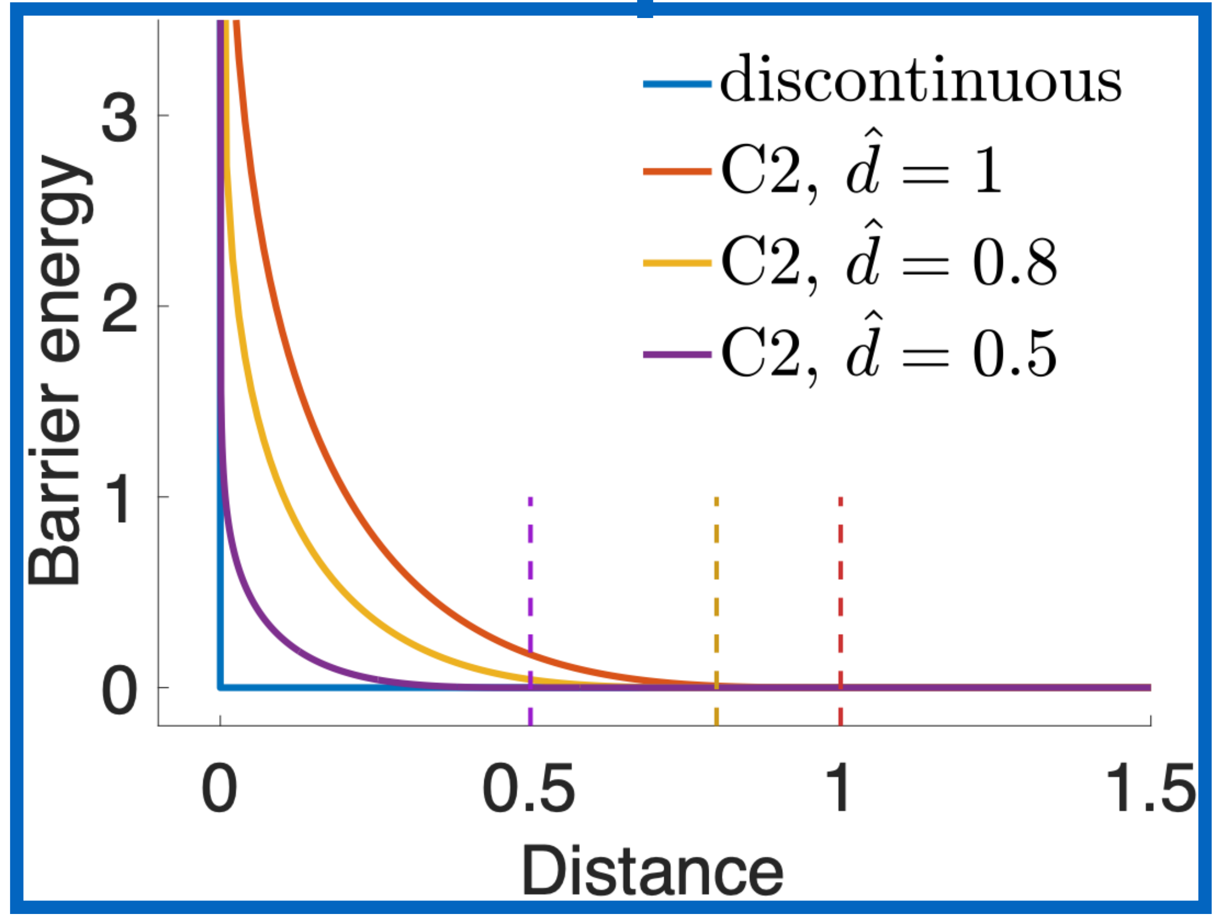
ZIZHOU HUANG, New York University, USA
MAX PAIK, New York University, USA
ZACHARY FERGUSON, Massachusetts Institute of Technology, USA
DANIELE PANOZZO, New York University, USA
DENIS ZORIN, New York University, USA



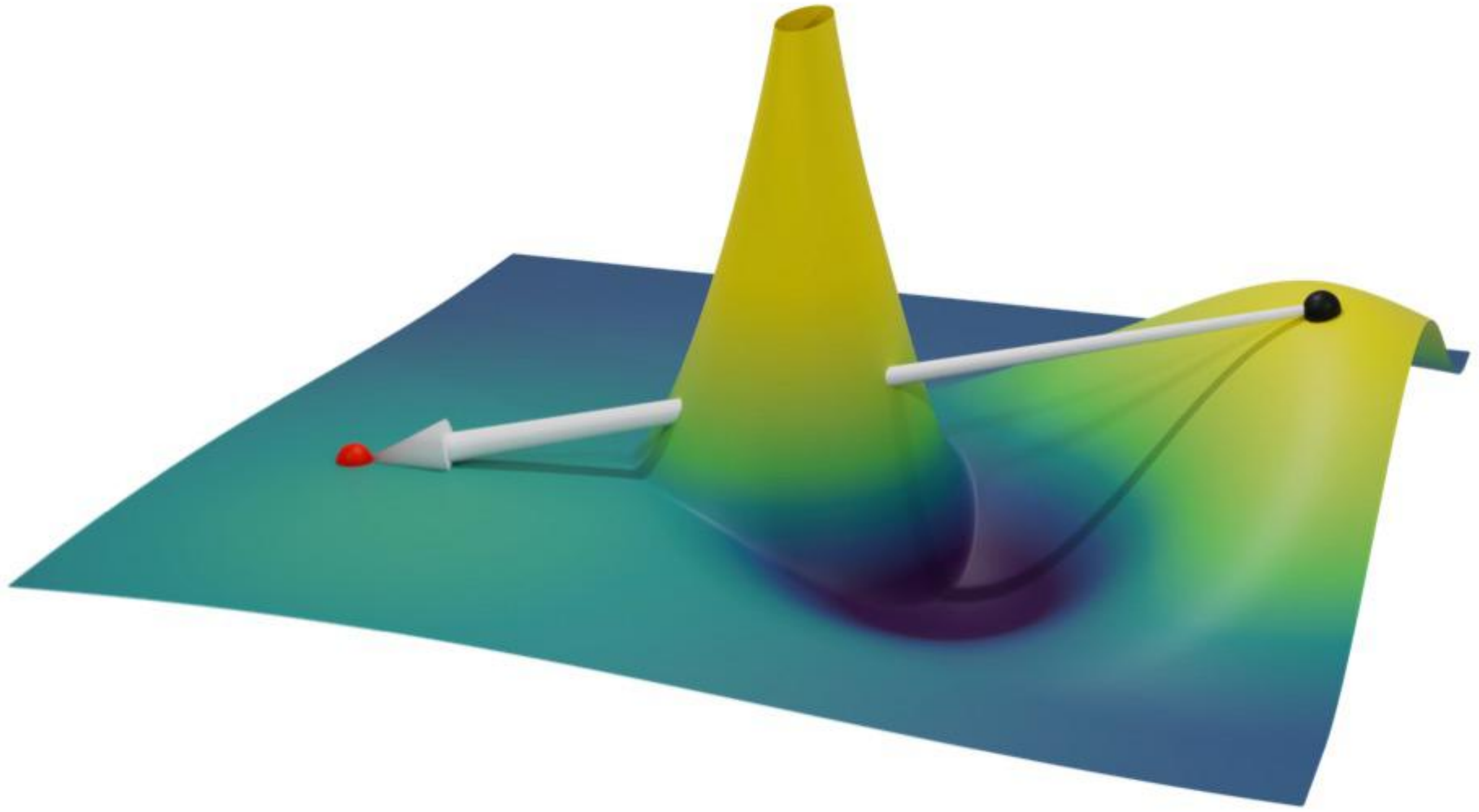
Contact and Friction* Potentials

$$x^{t+1} = \underset{x}{\operatorname{argmin}}\, E(x) + B(x, \hat{d}) + D(x)$$

Inertia & Elasticity



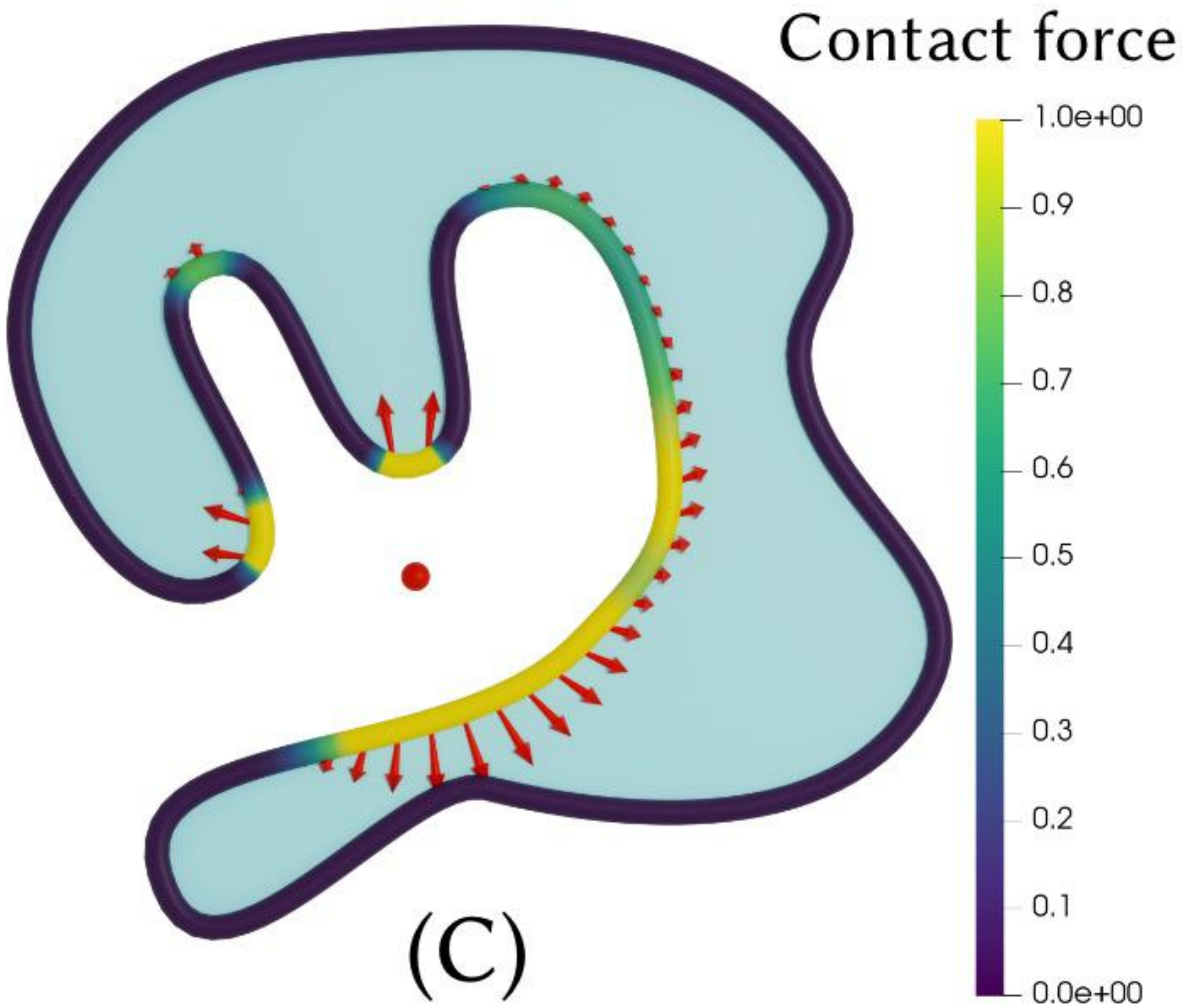
Friction*

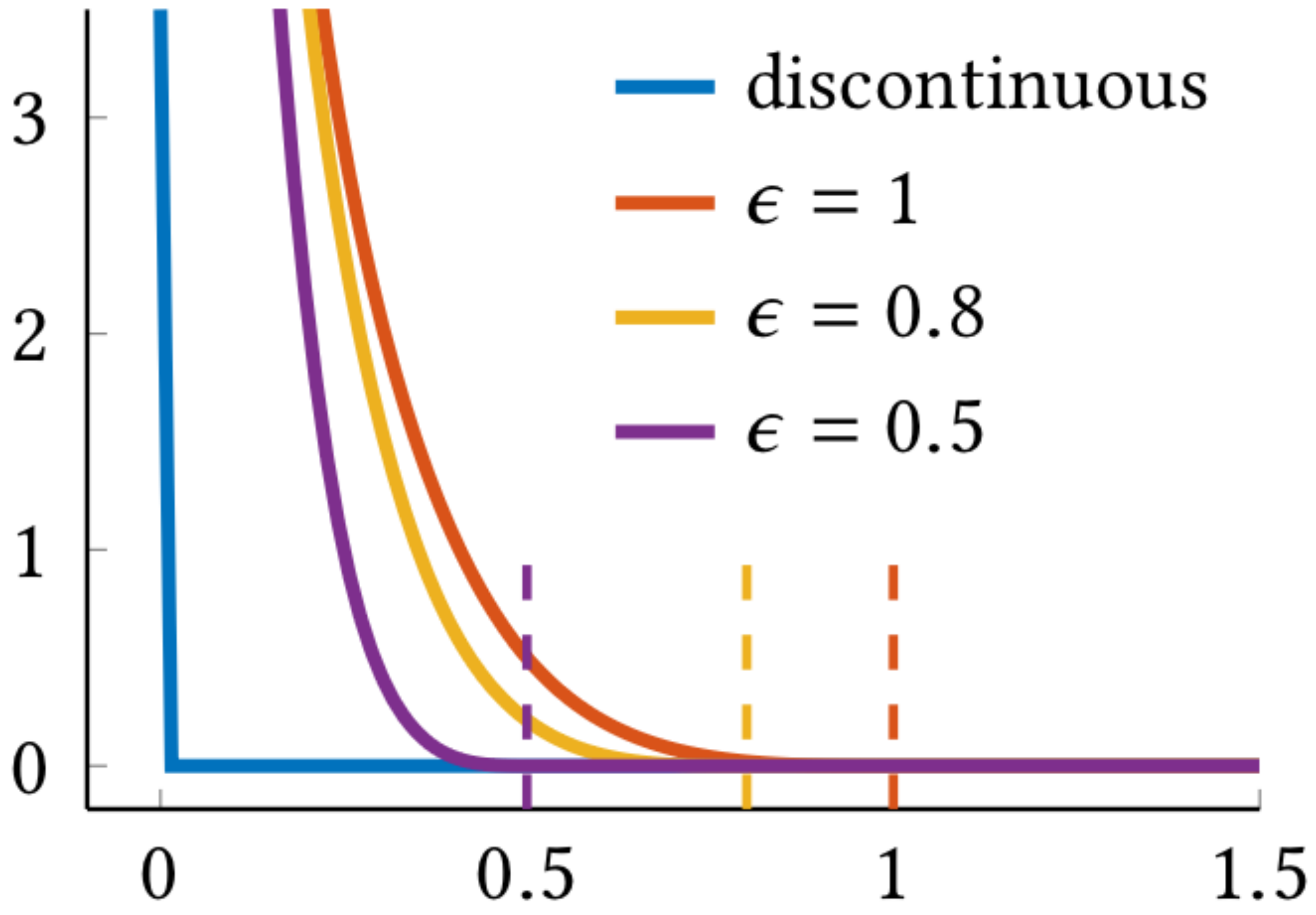


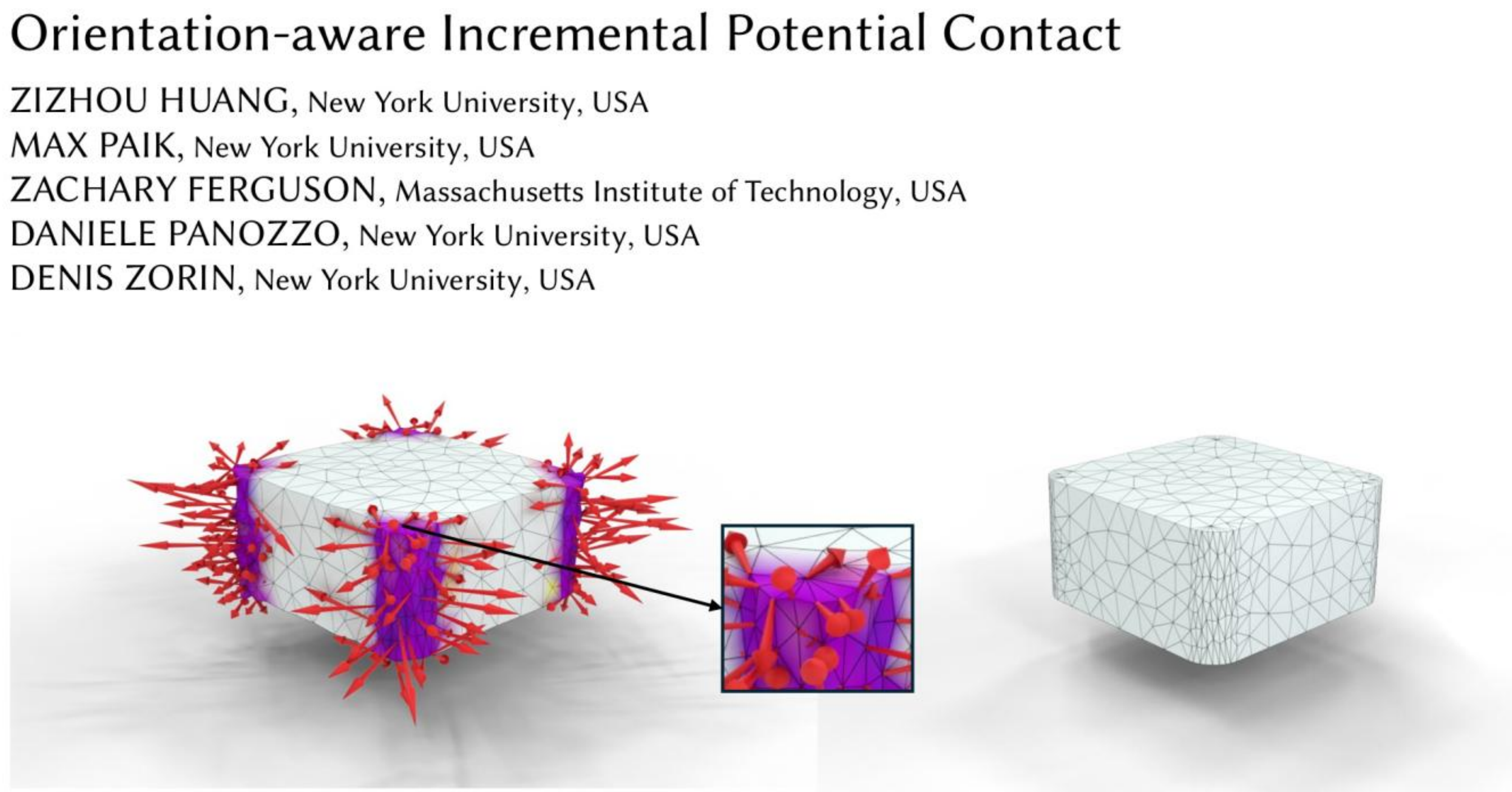
Minimize using Newton's method, with line search to ensure lack of constraints violation

Daniele Panozzo

Contact Potential



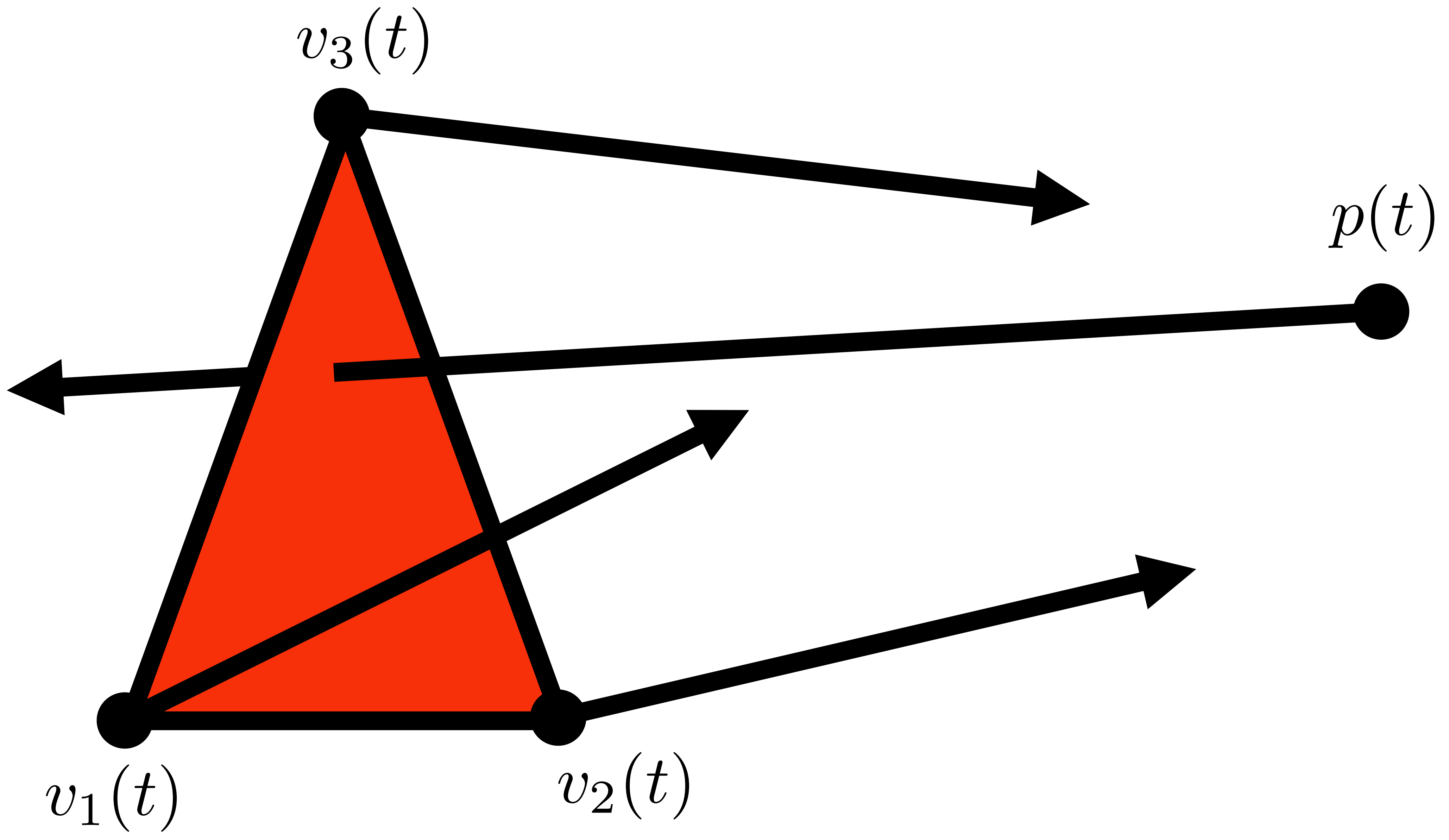




Point to Triangle

Multivariate (Snyder 1992)

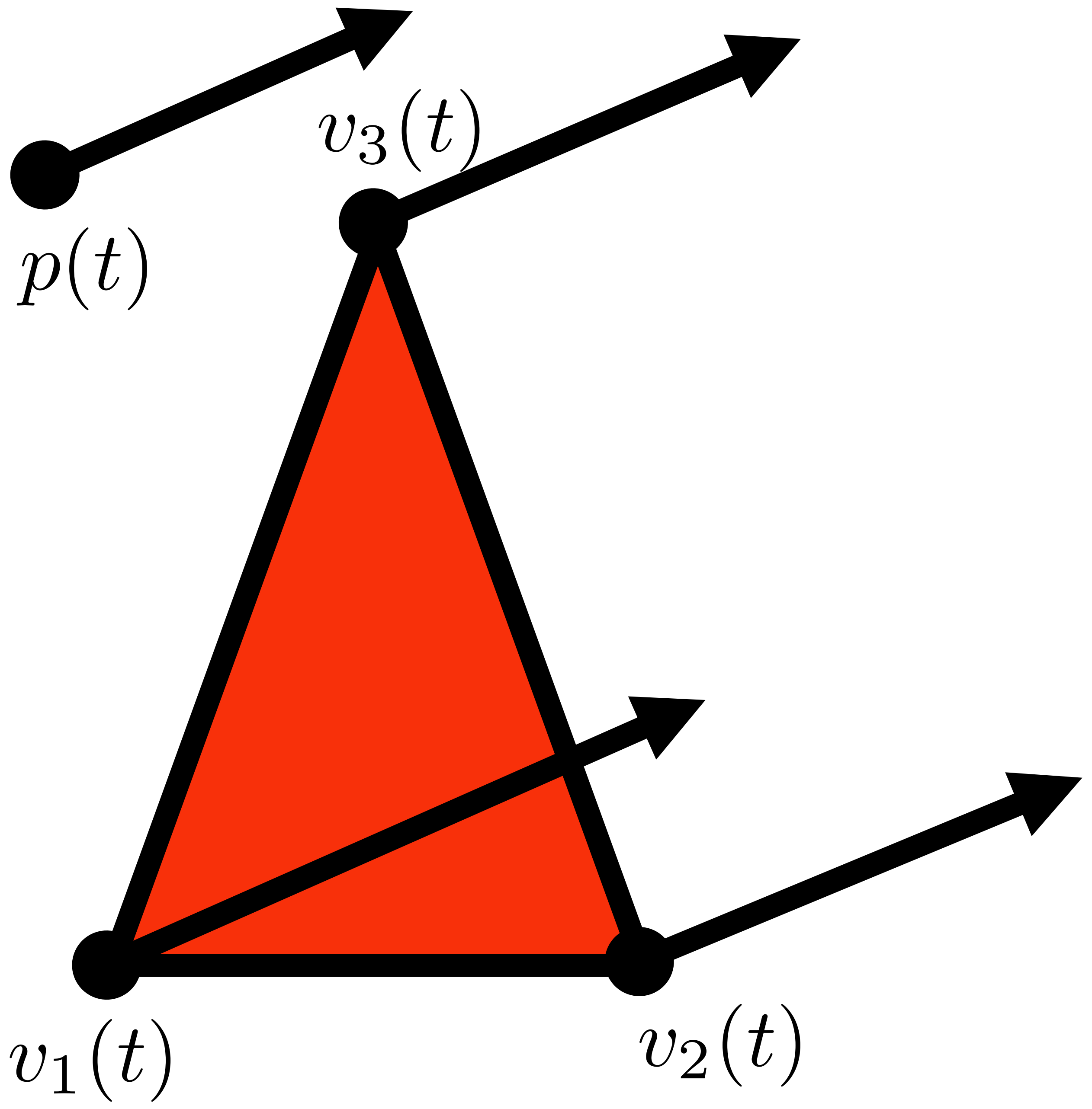
$$F_{\mathrm{vf}}(t,u,v) = p(t) - \big((1-u-v)v_1(t) + uv_2(t) + vv_3(t)\big),$$



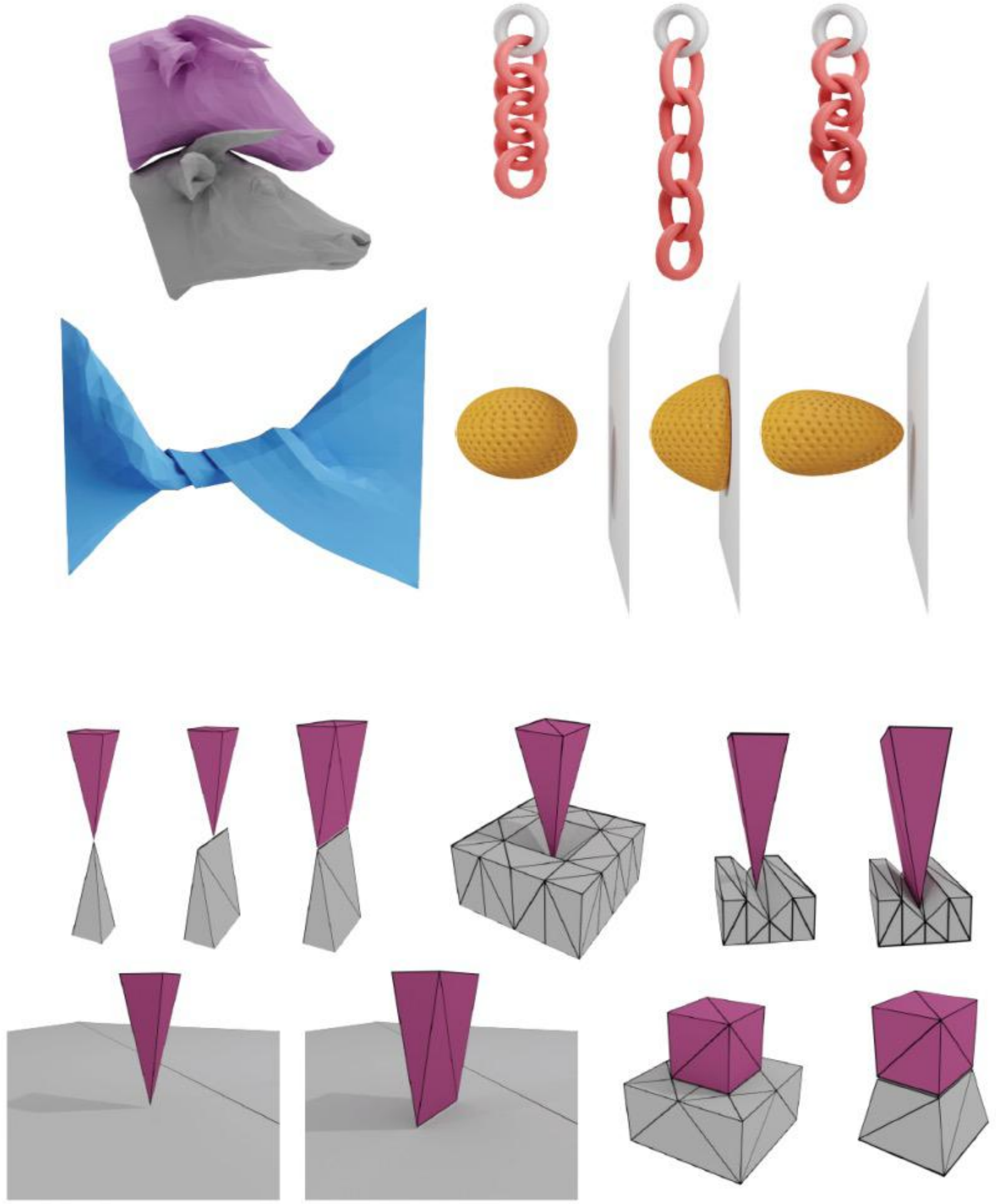
Univariate (modern approach)

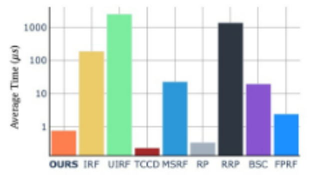
$$f(t) = \langle n(t), q(t) \rangle = 0,$$

$$n(t) = \big(v_2(t) - v_1(t)\big) \times \big(v_3(t) - v_1(t)\big) \text{ and } q(t) = p(t) - v_1(t)$$



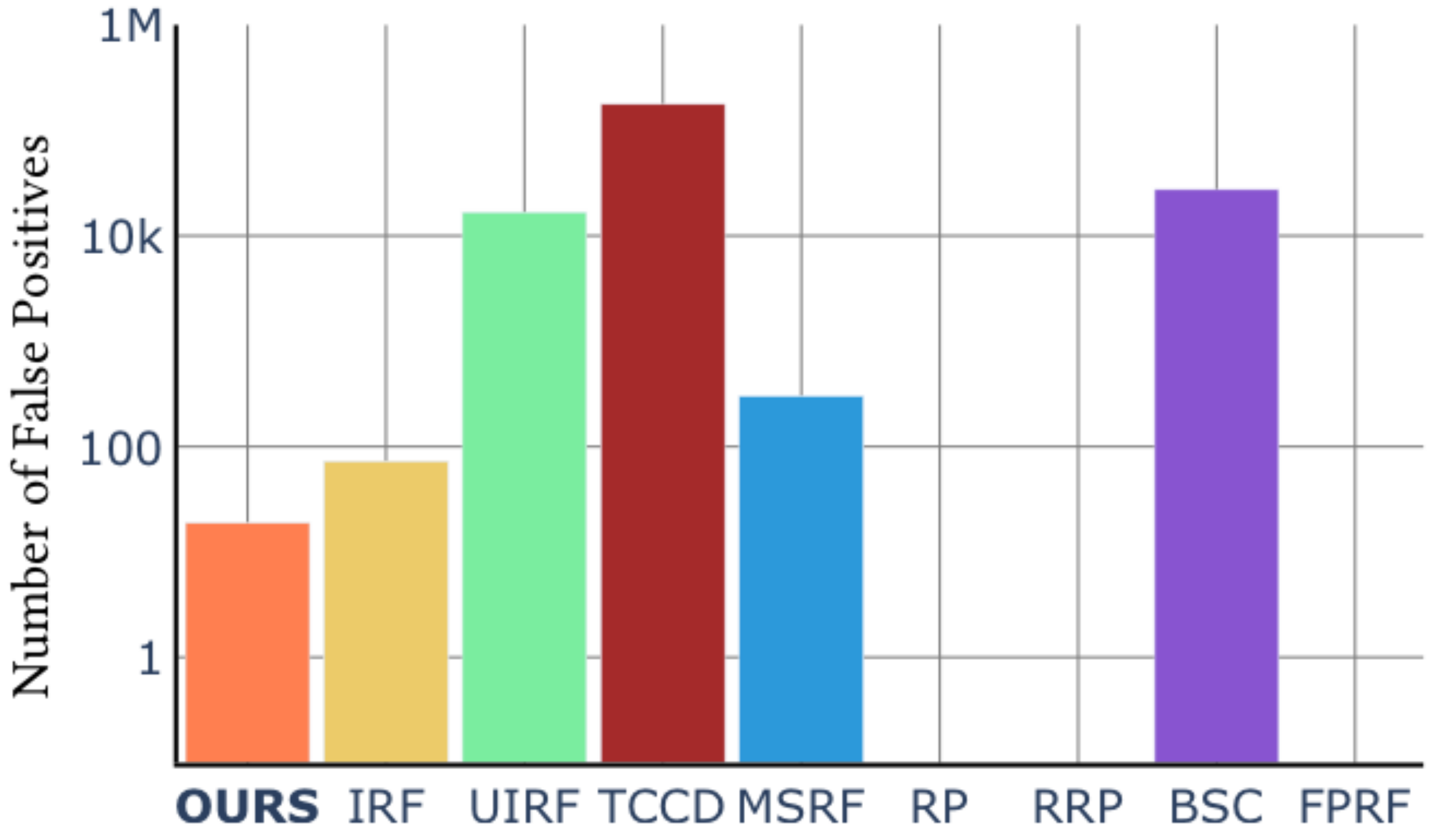
Continuous Collision Detection

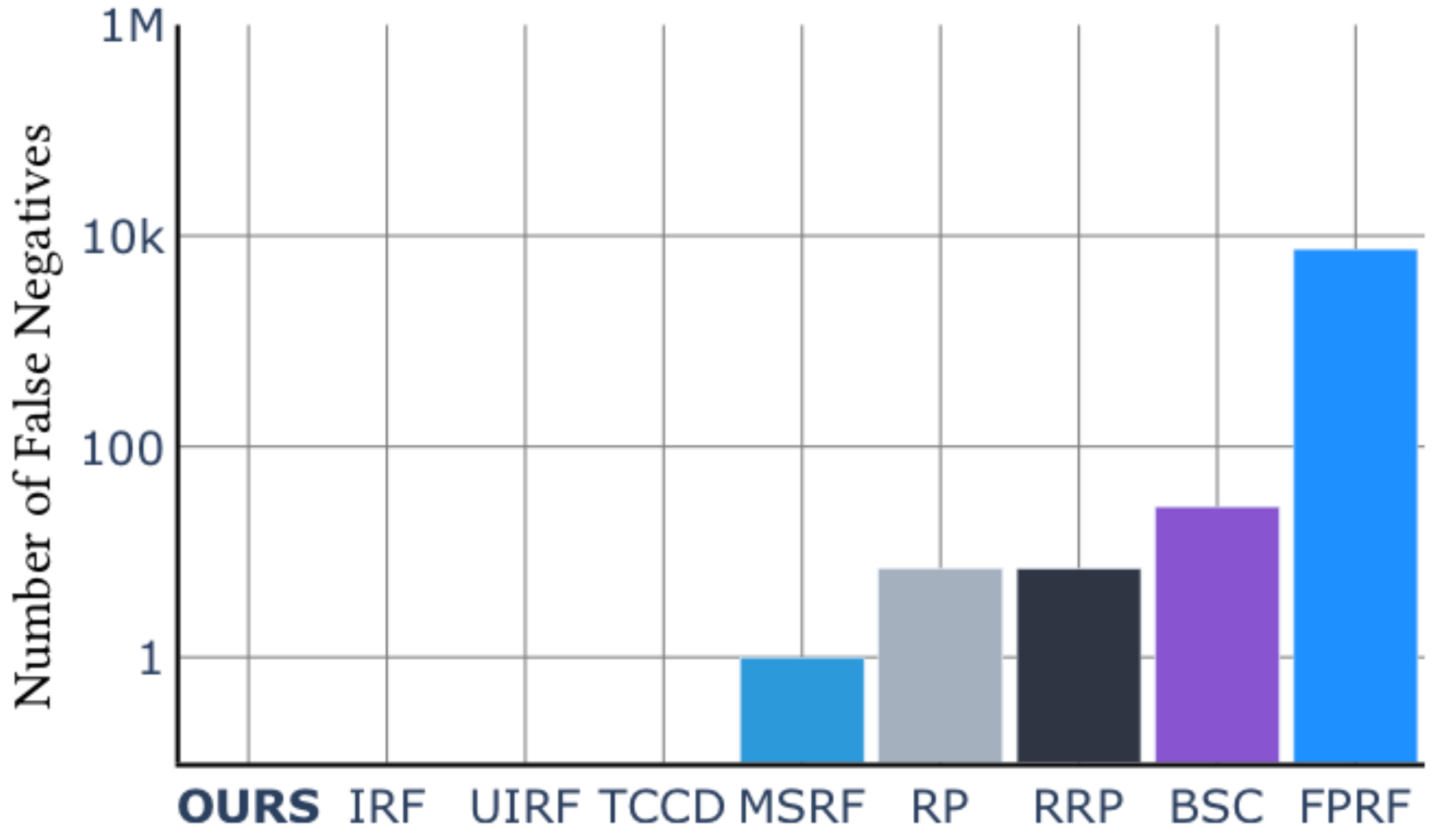


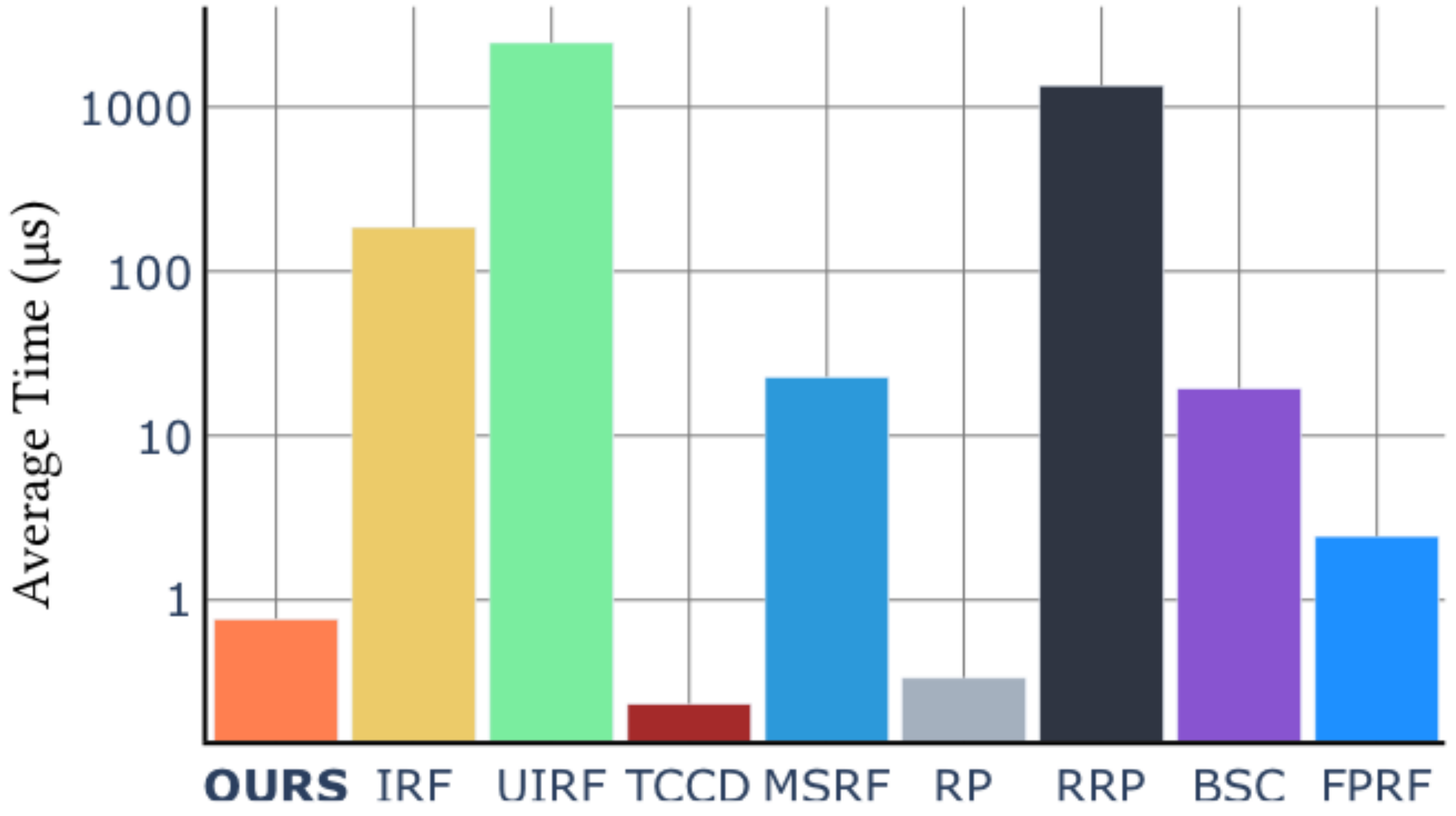


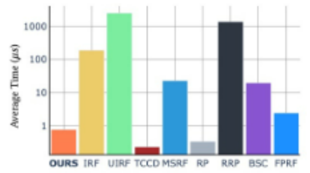
A Large Scale Benchmark and an Inclusion-Based Algorithm for Continuous Collision Detection
Bolun Wang, Zachary Ferguson, Teseo Schneider, Xin Jiang, Marco Attene, Daniele Panozzo,
ACM Transaction on Graphics, 2021
[Paper] [Video] [CCD Code] [Benchmark Code]

Benchmark



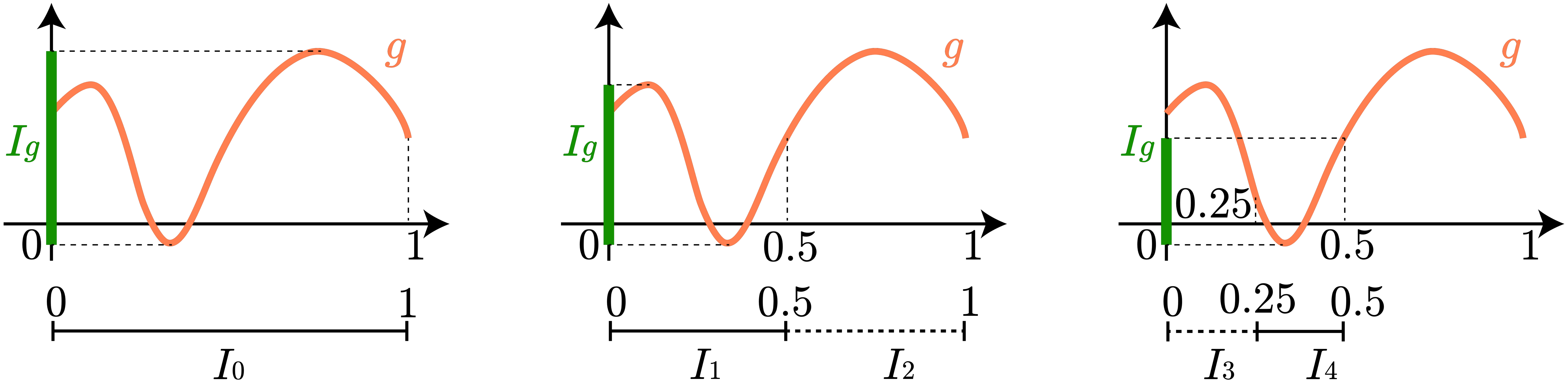






A Large Scale Benchmark and an Inclusion-Based Algorithm for Continuous Collision Detection
Bolun Wang, Zachary Ferguson, Teseo Schneider, Xin Jiang, Marco Attene, Daniele Panozzo,
ACM Transaction on Graphics, 2021
[Paper] [Video] [CCD Code] [Benchmark Code]

Bisection Root Finding



The inclusion function I_g is:

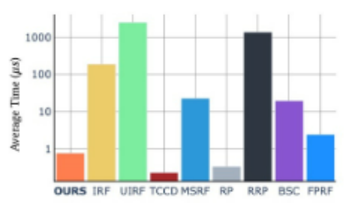
1. A conservative estimation (Inclusion) of the co-domain.
2. Convergent ($size(Ig) \to 0 \; when \; I_i \to 0$).

Algorithm

Algorithm 1 Inclusion-based root-finder

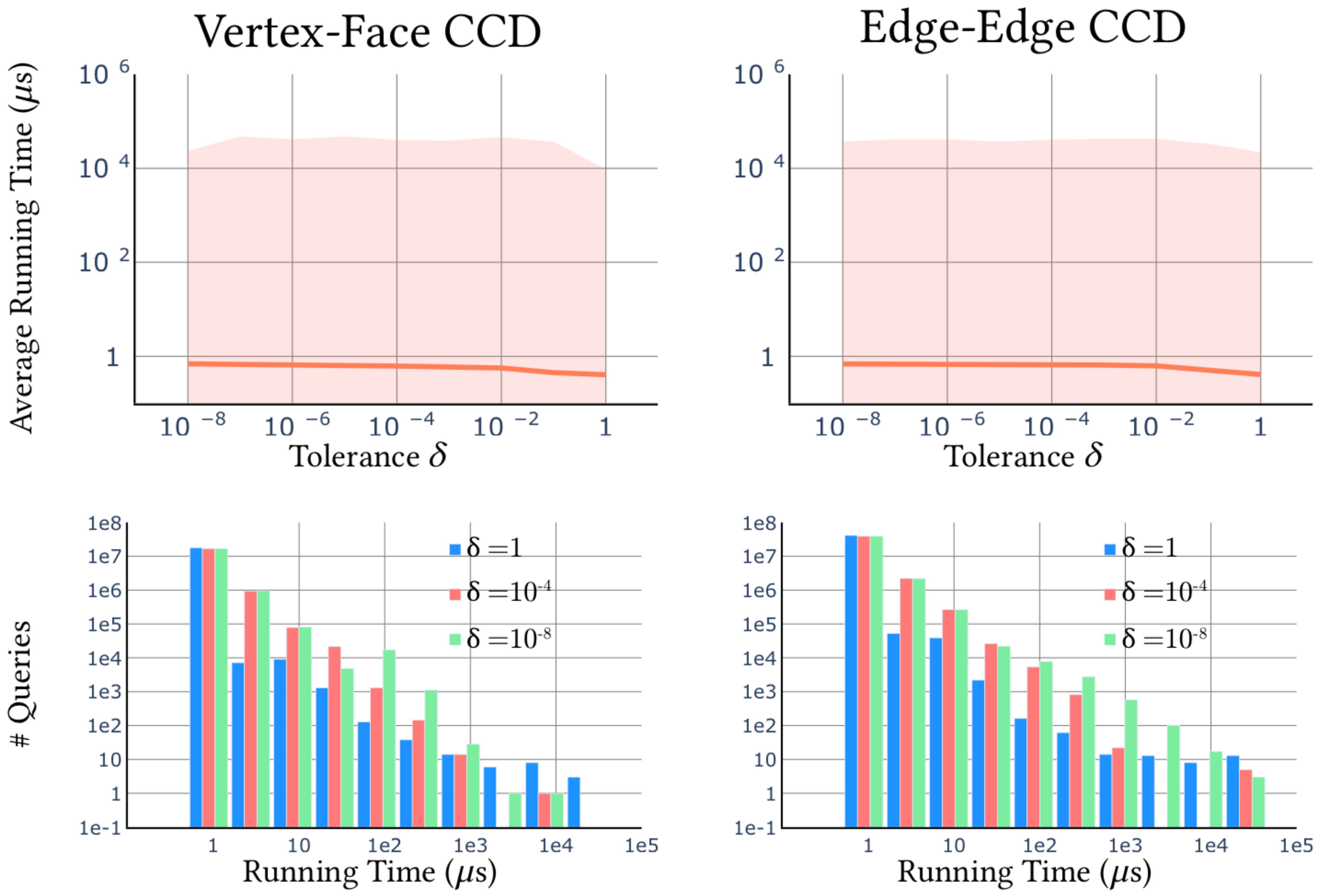
function SOLVE(I_0, g, δ)
    res $\leftarrow \emptyset$
    $S \leftarrow \{I_0\}$
    $\ell \leftarrow 0$
    while $L \neq \emptyset$ **do**
        $I \leftarrow$ POP(L)
        $I_g \leftarrow \square g(I)$ ▷ Compute the inclusion function
        if $0 \in I_g$ **then**
            if $w(I) < \delta$ **then** ▷ I is small enough
                res $\leftarrow R \cup \{I\}$
            else
                $I_1, I_2 \leftarrow$ SPLIT(I)
                $S \leftarrow S \cup \{I_1, I_2\}$
        $\ell \leftarrow \ell + 1$
     return res

- How to define the inclusion function?
- Implementation with intervals is slow
- We discovered (see the paper for the derivation) that a pure floating point inclusion function can be devised for this specific polynomial!



A Large Scale Benchmark and an Inclusion-Based Algorithm for Continuous Collision Detection
Bolun Wang, Zachary Ferguson, Teseo Schneider, Xin Jiang, Marco Attene, Daniele Panozzo,
ACM Transaction on Graphics, 2021
[Paper] [Video] [CCD Code] [Benchmark Code]

Performance

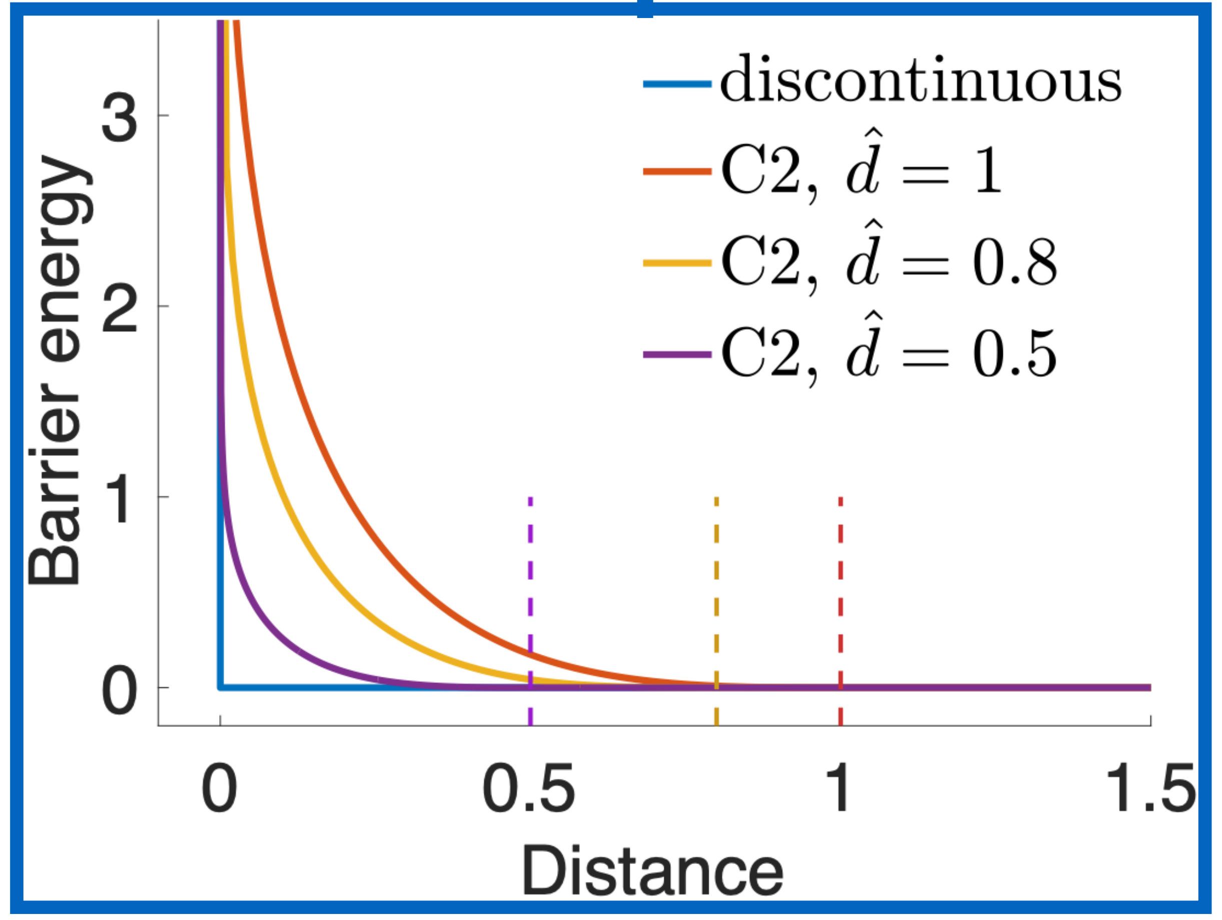


- Running time are suitable for large scale scenes, and fully parallelizable too!
- Additionally, since we only care about the first intersection, we can heavily prune.
- Implementation is challenging as the order of operations matters here!

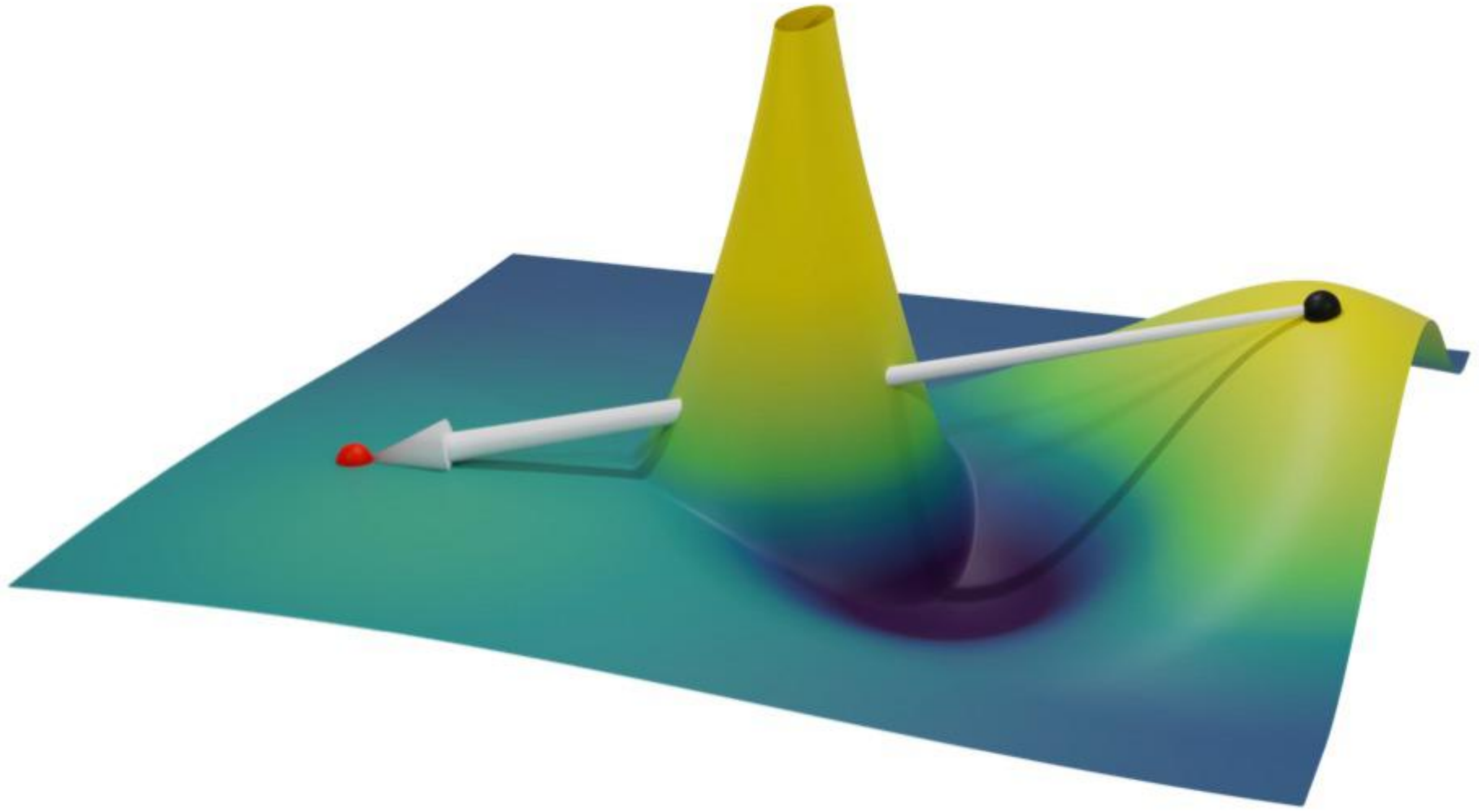
Contact and Friction* Potentials

$$x^{t+1} = \underset{x}{\operatorname{argmin}}\, E(x) + B(x, \hat{d}) + D(x)$$

Inertia & Elasticity

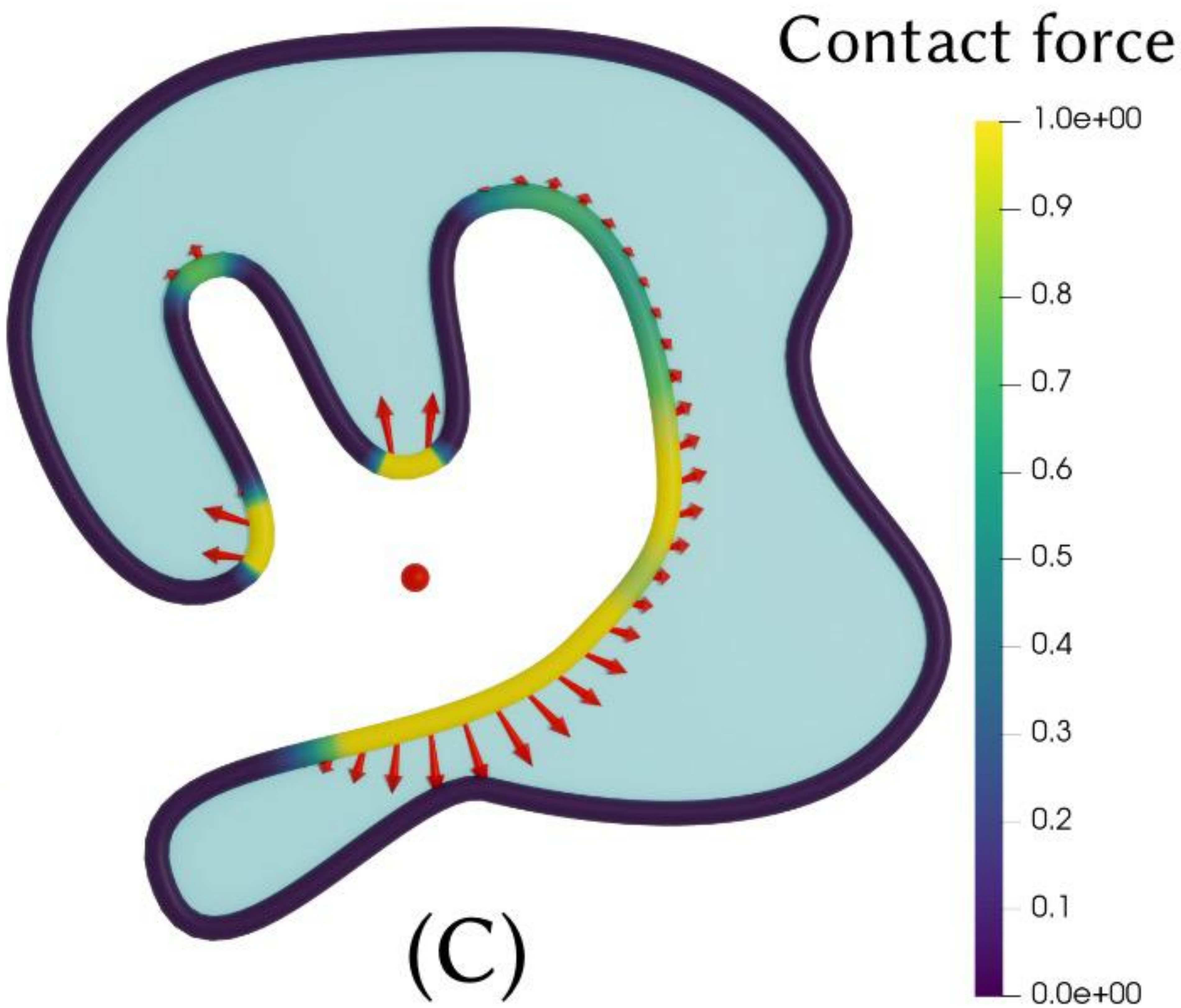


Friction*



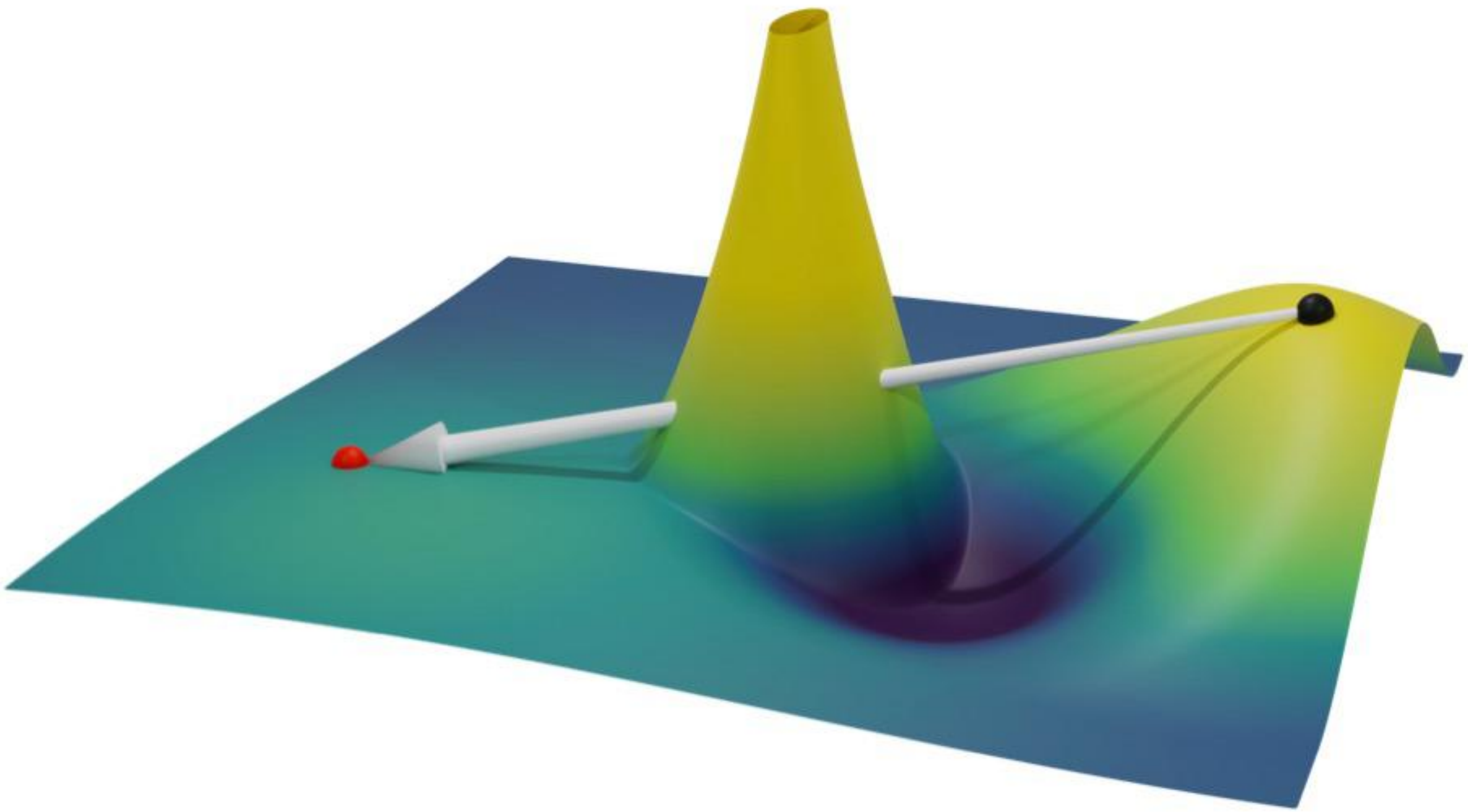
Minimize using Newton's method, with line search to ensure lack of constraints violation

Quadrature for Contact Potential



- The contact potential must be integrated with custom quadrature rules, so that the discrete quadrature diverges when the potential is infinite.
- We add a quadrature point on the closest contact point.
- This is crucial for convergence, otherwise the line search would stop convergence.

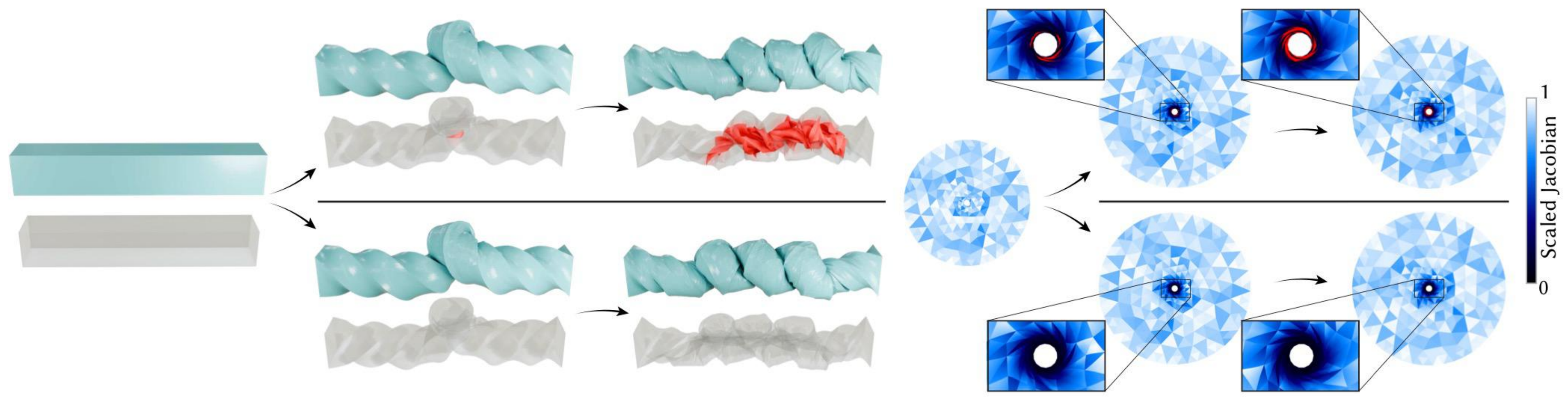
Quadrature for Contact Potential



High-Order Continuous Geometrical Validity

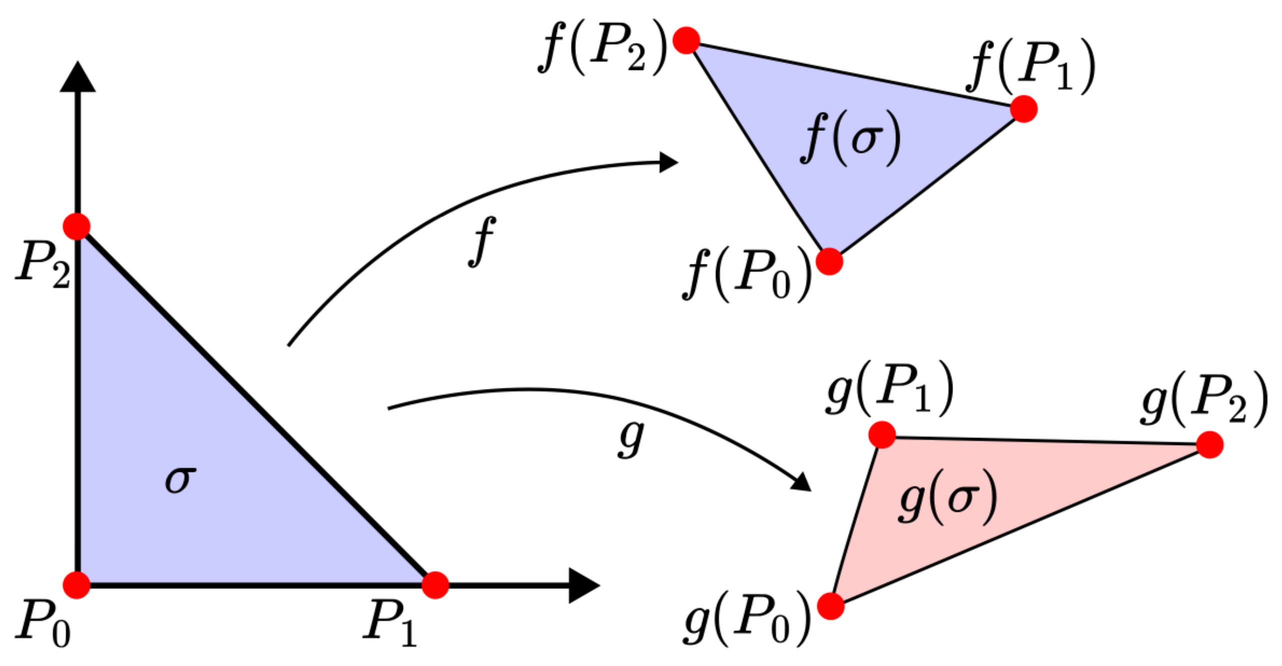
High-Order Continuous Geometrical Validity

FEDERICO SICHETTI, University of Genoa, Italy
ZIZHOU HUANG, New York University, United States
MARCO ATTENE, CNR-IMATI: GENOVA, Italy
DENIS ZORIN, New York University, United States
ENRICO PUPPO, University of Genoa, Italy
DANIELE PANOZZO, New York University, United States

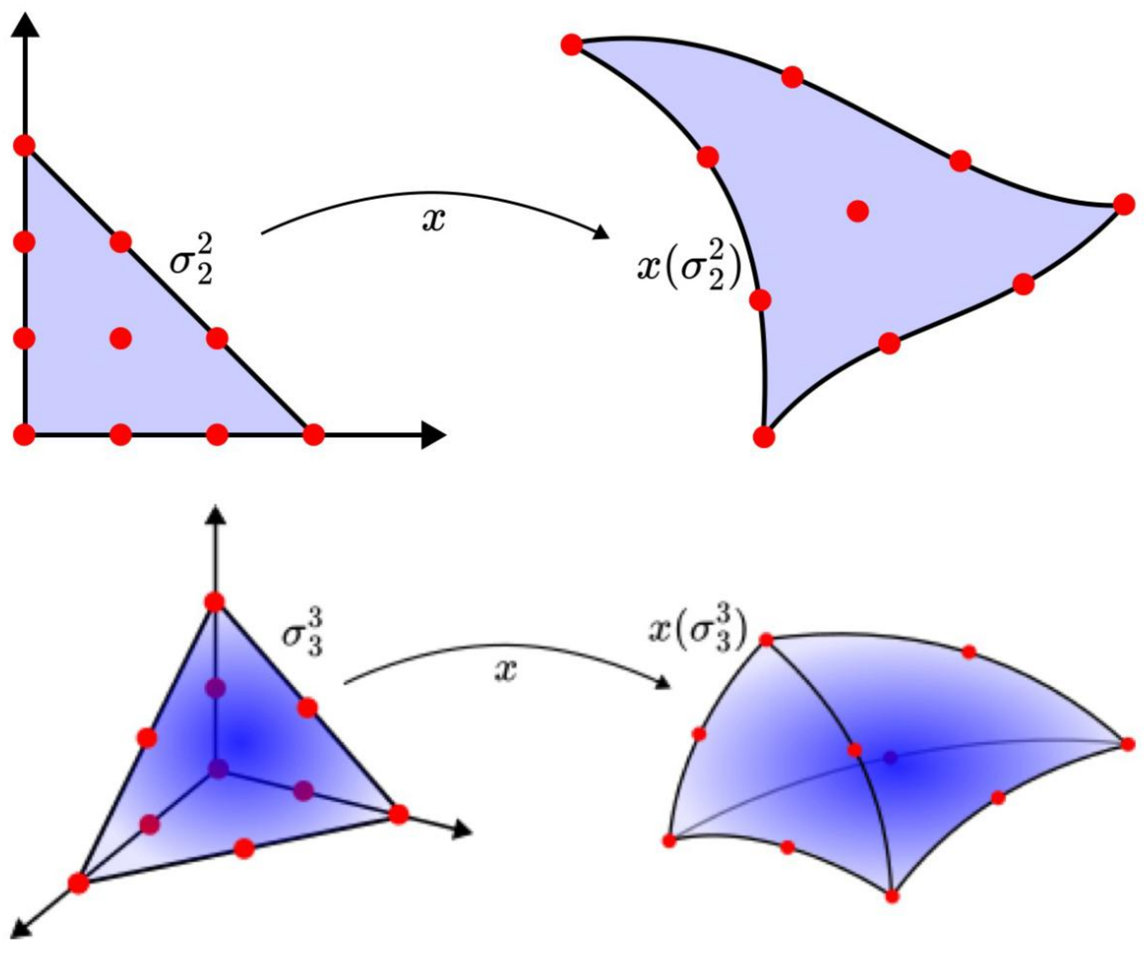


Geometric Map and Basis

Geometric Map

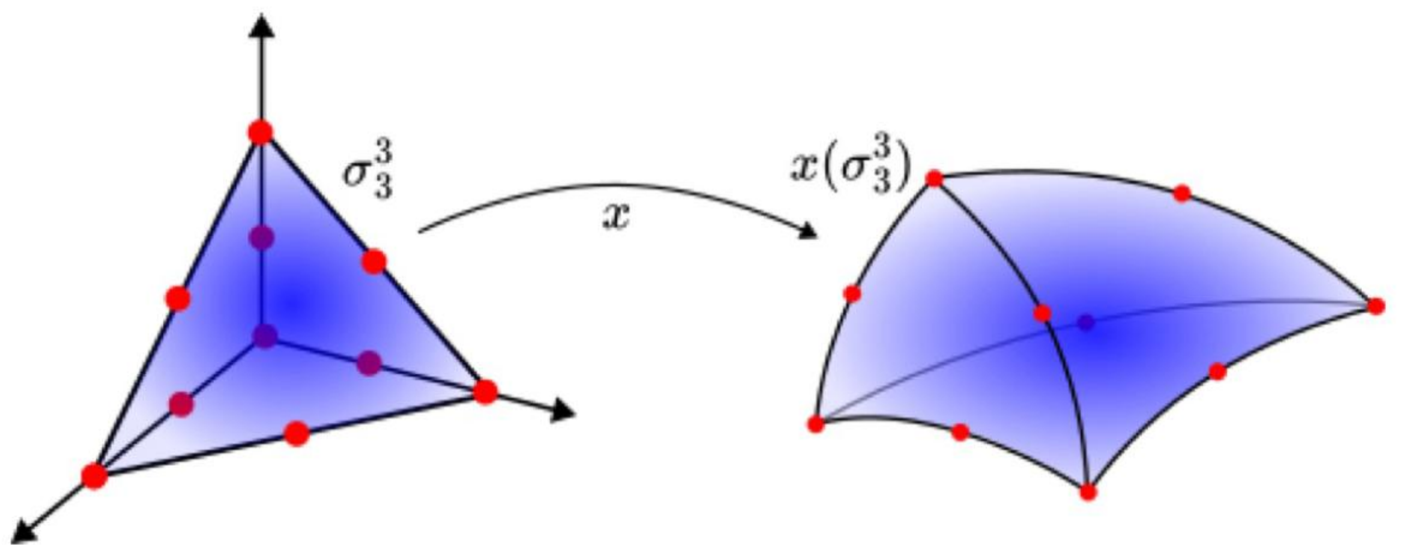


FEM Basis



- Both can be either linear or higher order.
- The geometric map must be always injective (usually an issue for meshing).
- If we solve for displacement, then the composition of geometric map and displacement must be injective.

Positivity of Jacobian



- Let's call it geometric map for simplicity:

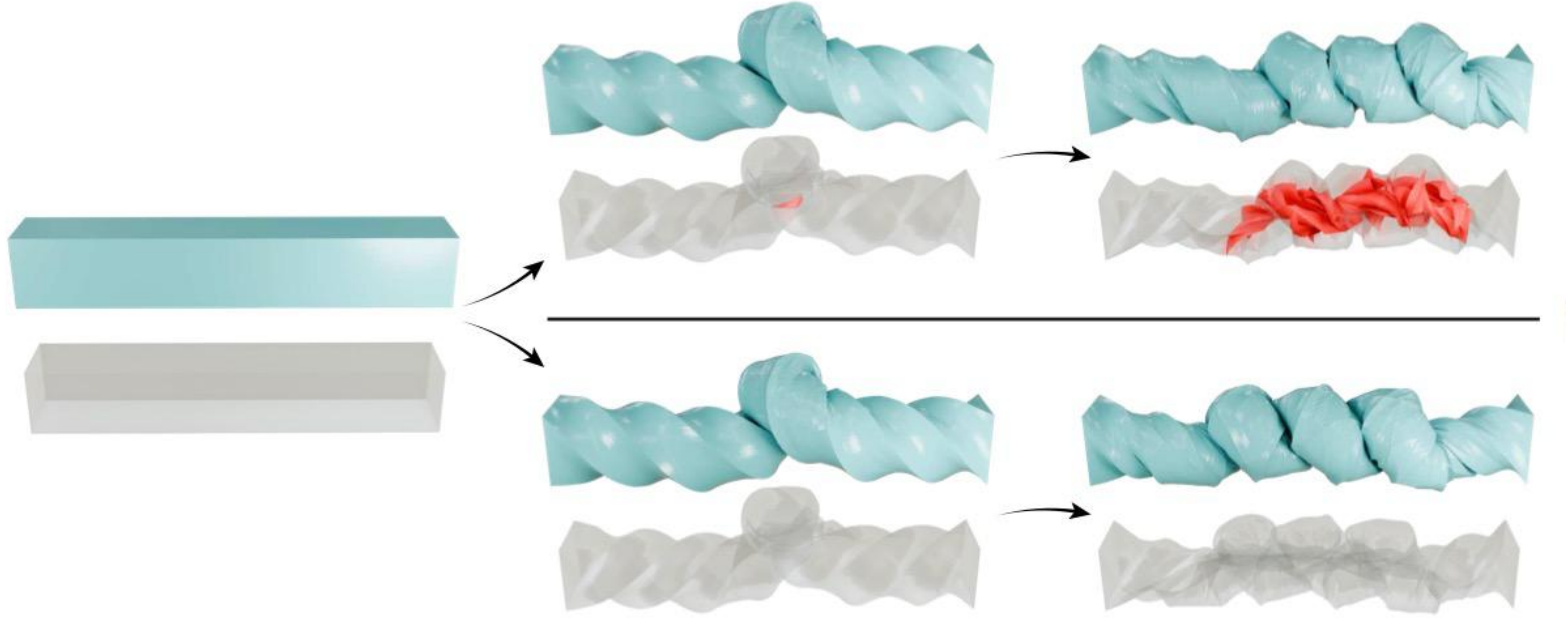
$$x : \sigma_s^n \longrightarrow \mathbb{R}^n$$

- The injectivity condition is usually expressed as:

$$|J_x| > 0$$

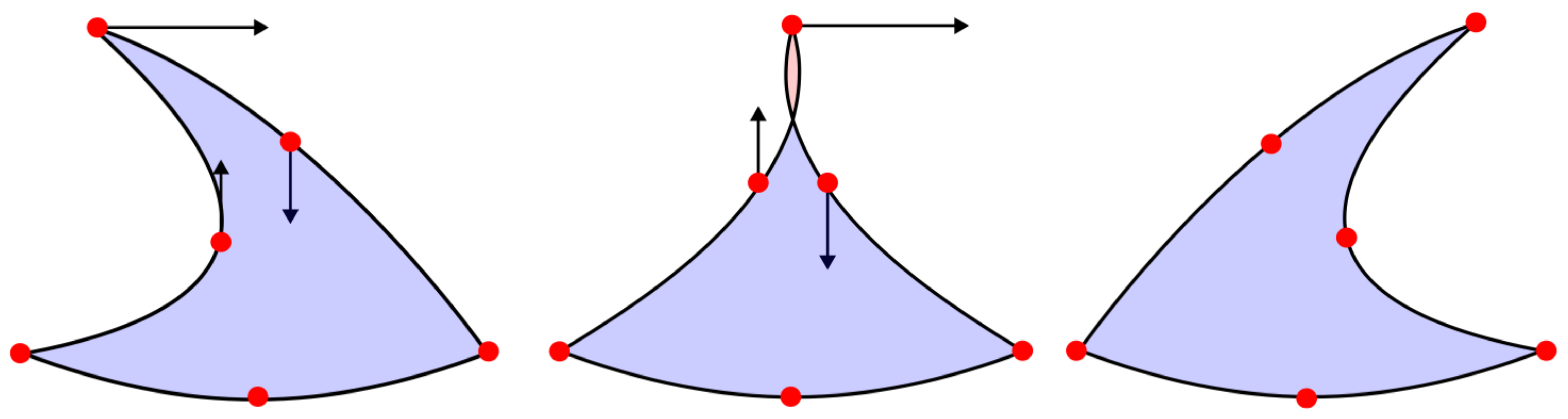
- And tested on quadrature points.

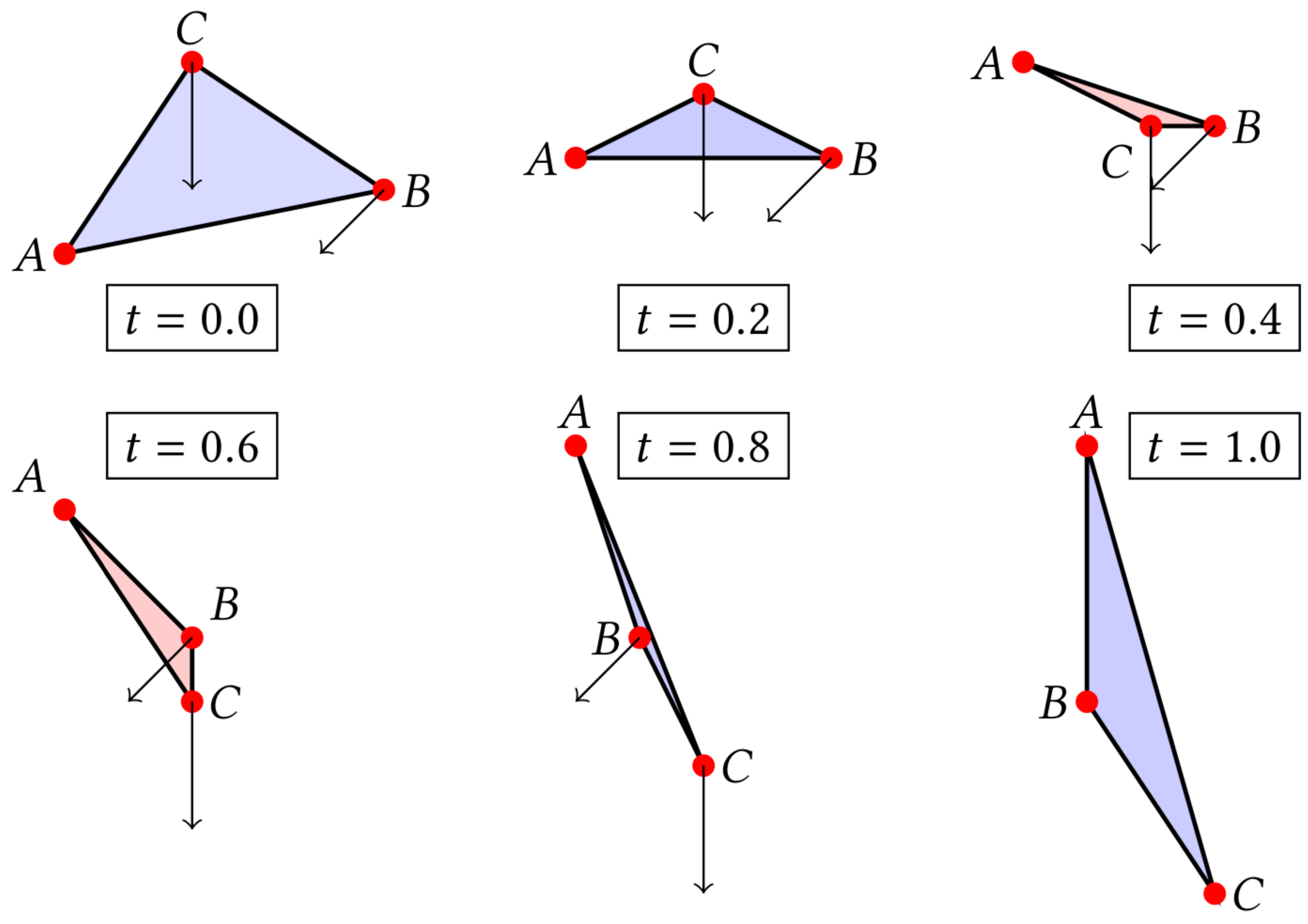
This is insufficient!



- Testing it on quadrature points misses violations very often: we discovered that all PolyFEM high-order displacement simulations suffer from this issue… [Johnen at al 2014], GMSH has a solution, not robust to rounding but way better than sampling.

This is still insufficient!





- It ignores what is happening over the time dimension. Testing at discrete time step is insufficient.

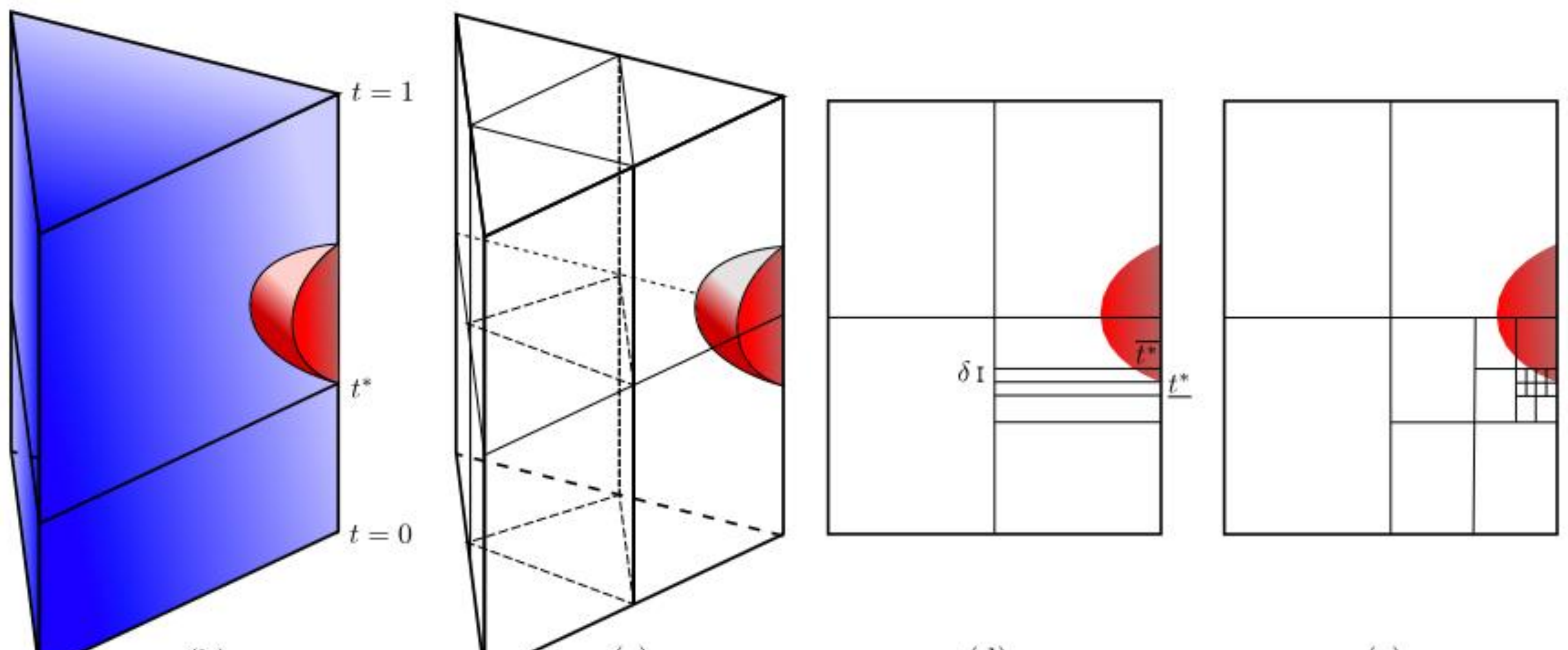
Time-Dependent Positivity Check

Definition 3 (Dynamic reference element). *Let σ_s^n be a reference element. The dynamic element $\bar{\sigma}_s^n$ of σ_s^n is the $(n+1)$-dimensional reference element $\sigma_s^{n+1} = \sigma_s^n \times [0, 1]$.*

Assuming linear trajectories, the *dynamic geometric map* $\bar{x} : \mathbb{R}^{n+1} \to \mathbb{R}^n$ of order p for $\bar{\sigma}_s^n$ is expressed by linear interpolation of the n-dimensional geometric maps $x^0(\xi)$ and $x^1(\xi)$ of the static element at the two consecutive time steps:

$$\bar{x}(\xi, t) = x^0(\xi) + t(x^1(\xi) - x^0(\xi)). \tag{2}$$

This map is of order p in the ξ variables and linear in t.



(b) (c) (d) (e)

- The new condition to check is:

$$|J_{\bar{x}}| > 0$$

- Similarly to before, we use bisection, but it is more nuanced here

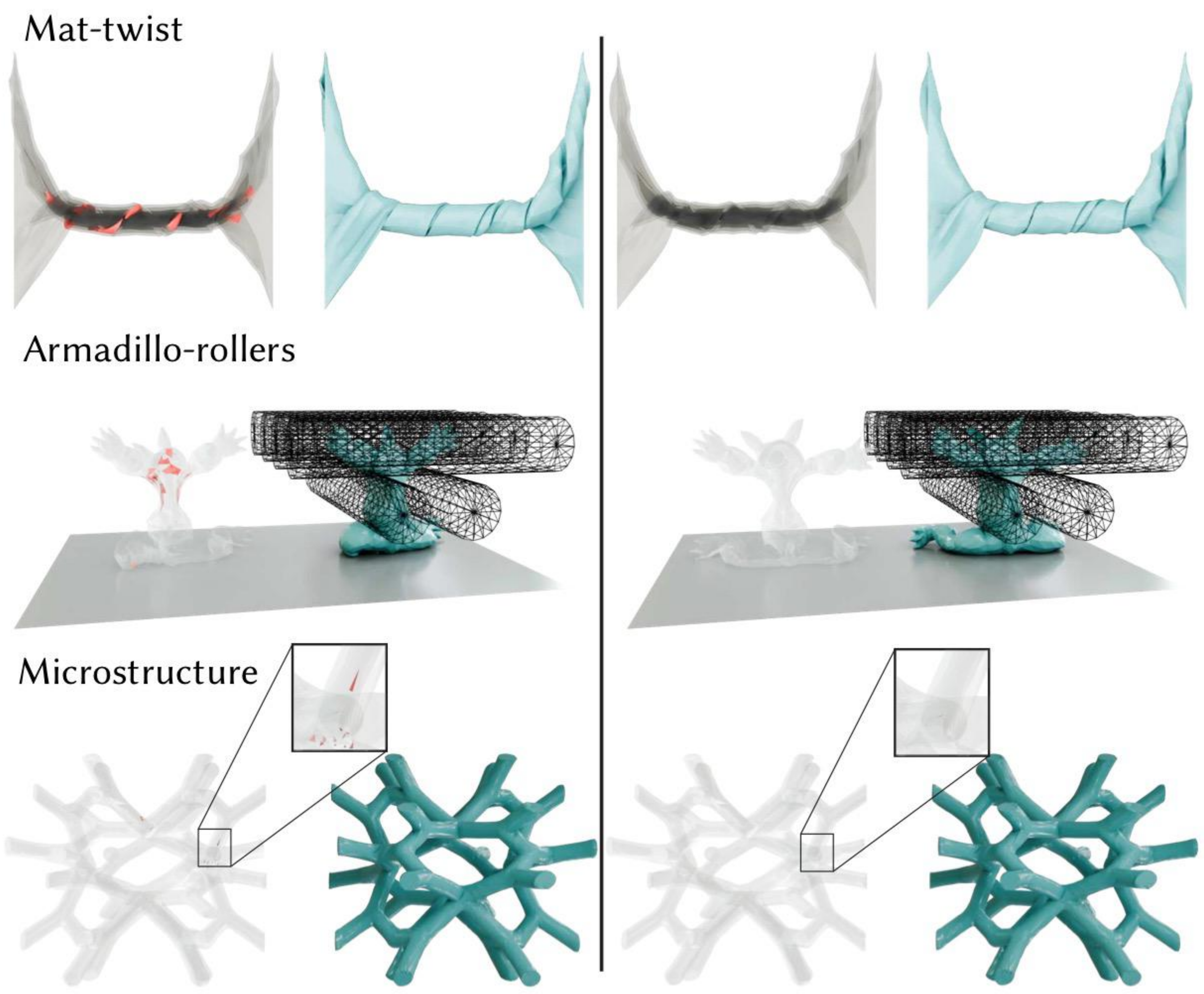
Algorithm

Algorithm 1 Maximum valid time step with inclusion functions

function $\text{MaxValidStep}(J, \delta, l_{\max})$
  $P \leftarrow \text{PriorityQueue}(\prec)$ ▷ *priority queue for subdomains*
  $\overline{t^*} \leftarrow 1$ ▷ *initialize upper bound of t^**
  $\underline{t^*} \leftarrow 0$ ▷ *initialize lower bound of t^**
  $\text{Push}(P, \sigma)$
  $F \leftarrow$ FALSE ▷ *flag of whether an invalidity has been found*
  $l \leftarrow 0$ ▷ *maximum subdivision depth reached so far*
  while TRUE **do**
    if $F \wedge (\overline{t^*} - \underline{t^*} \leq \delta) \wedge (\underline{t^*} > 0)$ **then** ▷ *reached accuracy*
      return $\underline{t^*}$ ▷ *conservative estimate of t^**
    if $\text{IsEmpty}(P)$ **then**
      return 1
    $S \leftarrow \text{Pop}(P)$ ▷ *get the next subdomain from P*
    $l \leftarrow \max\{l, \text{Depth}(S)\}$ ▷ *update maximum depth*
    if $l > l_{\max}$ **then** ▷ *maximum level reached: give up*
      return $\underline{t^*}$ ▷ *conservative estimate of t^**
    $\underline{t^*} \leftarrow \text{StartTime}(S)$ ▷ *everything before this time is valid*
    $I \leftarrow \square_{\min} J(S)$ ▷ *check minimum inclusion*
    if $\text{High}(I) \leq 0$ **then** ▷ *there is an invalidity in S*
      if $\text{EndTime}(S) < \overline{t^*}$ **then**
        $F \leftarrow$ TRUE
        $\overline{t^*} \leftarrow \text{EndTime}(S)$
      $\text{Push}(P, \psi^-(S))$ ▷ *bisect on the t axis only*
      $\text{Push}(P, \psi^+(S))$ ▷ *bisect on the t axis only*
    else if $\neg(\text{Low}(I) > 0)$ **then**
      for $q \in \{1, \ldots, Q\}$ **do**
        $\text{Push}(P, \psi^q(S))$ ▷ *subdivide on ξ and bisect on t*
function $\prec(S_0, S_1)$ ▷ *priority function*
  if $\text{StartTime}(S_0) \neq \text{StartTime}(S_1)$ **then** ▷ *lower time first*
    return $\text{StartTime}(S_0) < \text{StartTime}(S_1)$
  else ▷ *for ties, prioritize boxes most likely to be invalid*
    return $\text{High}(\square_{\min} J(S_0)) < \text{High}(\square_{\min} J(S_1))$

- Following [Johnen et al. 2014], we build the inclusion function over the Bezier form of the polynomial (massive reduction in depth).
- We precompute in rational both base changes and subdivision rules. At runtime, we use intervals to account for rounding errors.
- We use different split rules for space and time, see paper for details.

Running Times



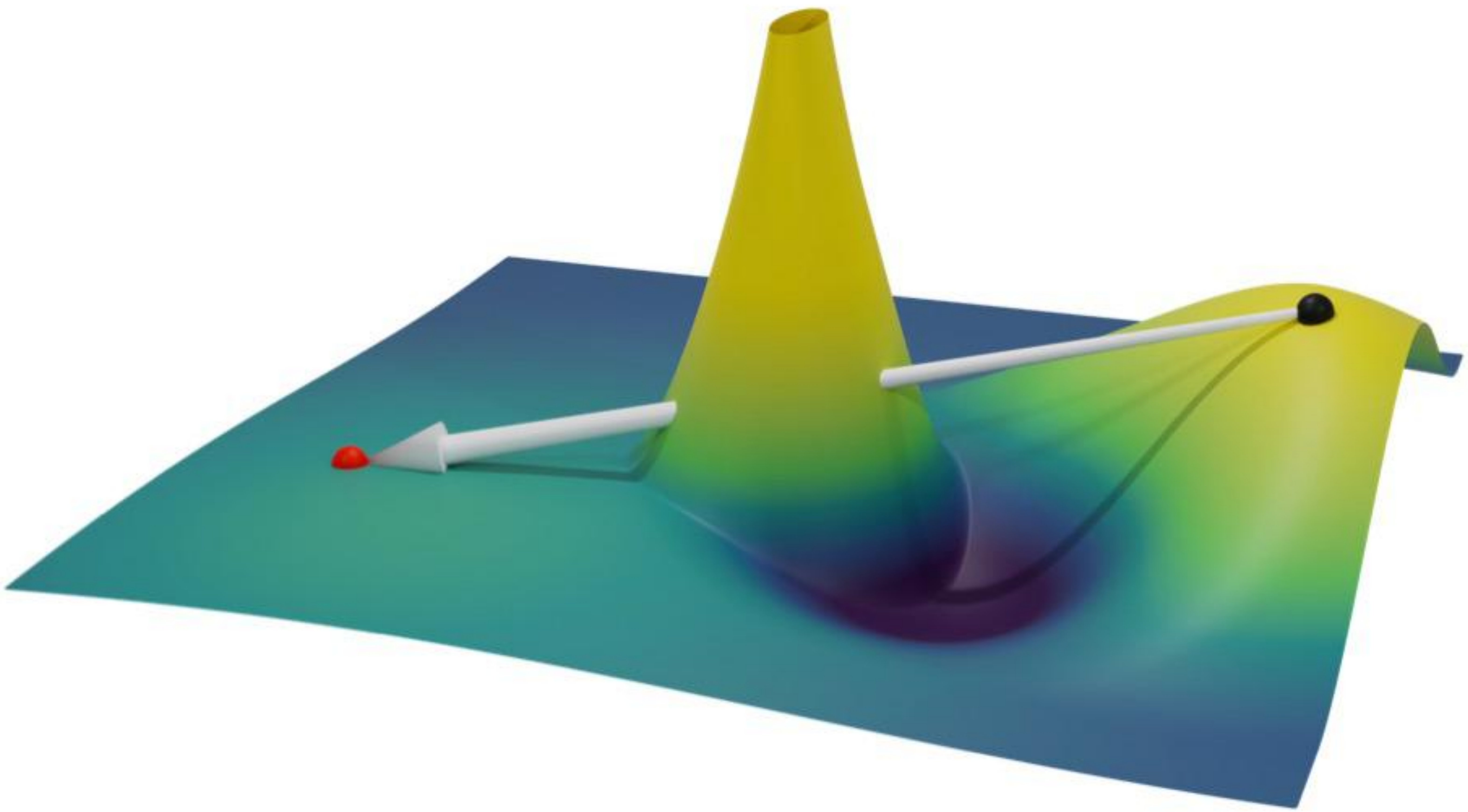
Running Times

Dataset	Element	n	s	p	# elements			time per element (μs)					
				tot	valid	invalid	pre	avg	valid	invalid	max	std	
Kangaroo	Triangles	2	2	1	172800	146248	26552	9	3	1	21	29	4
				2	172800	145602	27198	12	12	9	28	189	7
				3	172800	145623	27177	33	52	47	80	295	12
				4	172800	118436	27187	89	187	177	240	1035	24
Bar	Quadrangles	2	1	1	130291	130244	47	16	8	8	21	73	1
				2	31879	31808	71	115	125	125	348	2021	22
Armadillo	Tetrahedra	3	3	1	54985	51605	3380	21	2	1	9	19	2
				2	54985	48599	6386	229	280	268	373	24197	142
				3	54985	47988	6997	1936	4720	4216	8173	176461	3357
Bunny	Tetrahedra	3	3	1	19800	19561	239	24	1	1	10	69	1
				2	19800	19058	742	235	277	270	451	7132	98
				3	19800	18954	846	1964	4290	4133	7811	84298	1731

Running Times

Static algorithm	#val	#inv	#und	avg μs	med μs
Quadrature Points	48214	6771	-	16	16
Interval Bisection	46281	6561	2143	1400	23
FP Bézier (no optim.)	48050	6935	0	71	78
Ours (no optim.)	48050	6935	0	86	95
FP Bézier	48050	6935	0	5	5
Ours	48050	6935	0	8	8

Elastic Potential Quadrature



Elastic Potential Quadrature

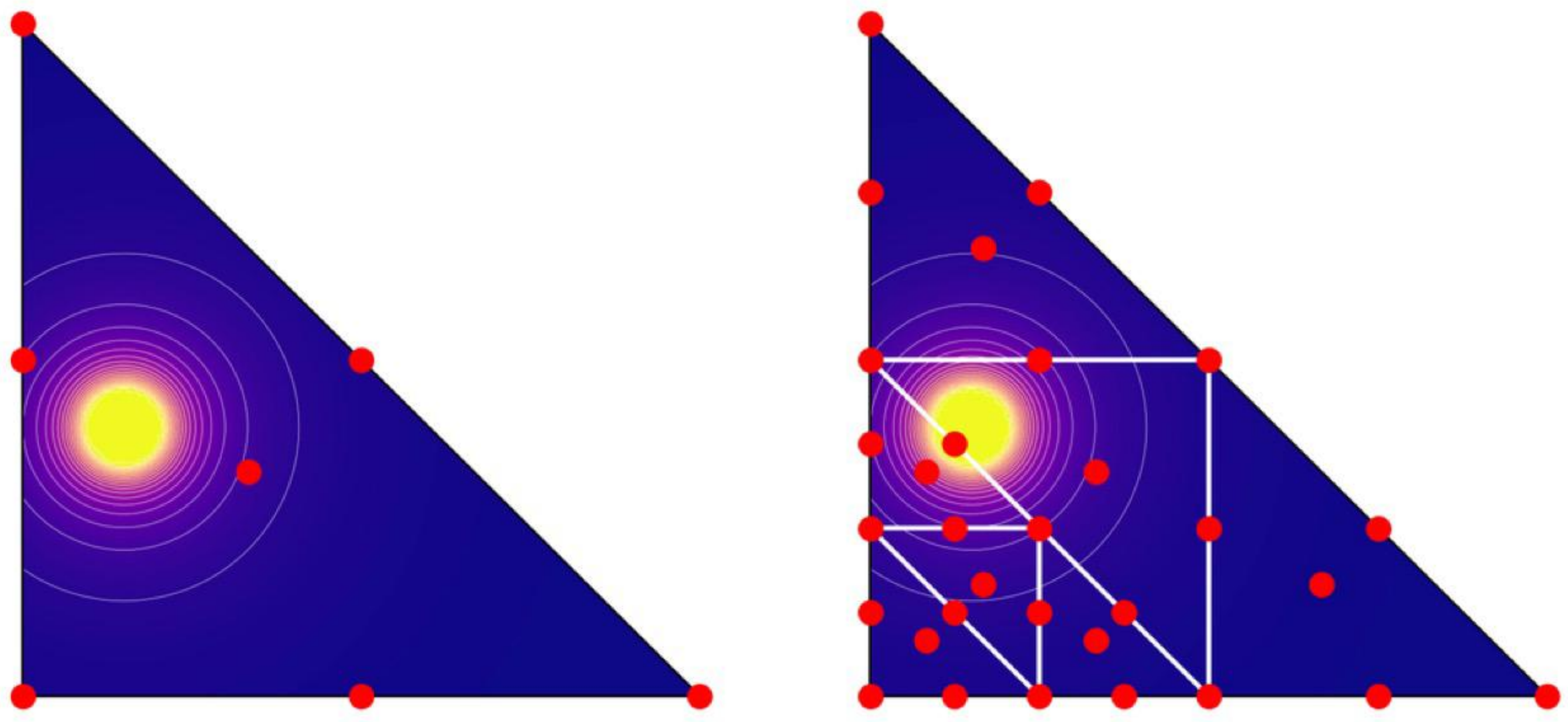
- Integrating the elastic potential requires care: if the potential diverges, the discrete quadrature should also diverge.

- Consider for example Neo-Hookean Elasticity:

$$\sigma[u] = \mu(F[u] - F[u]^{-T}) + \lambda \ln(\det F[u])F[u]^{-T}$$

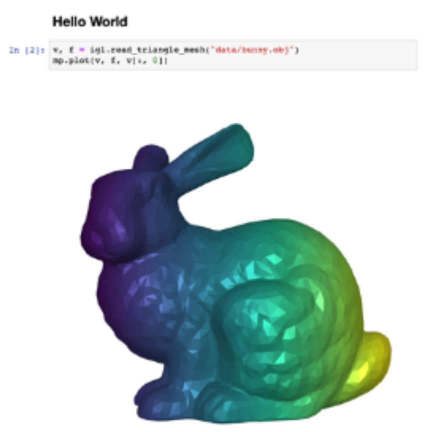
- Or Mooney-Rivlin

$$\Psi[u] = c_1(\widetilde{I}_1 - d) + c_2(\widetilde{I}_2 - d) + \frac{k}{2}\ln^2(J)$$



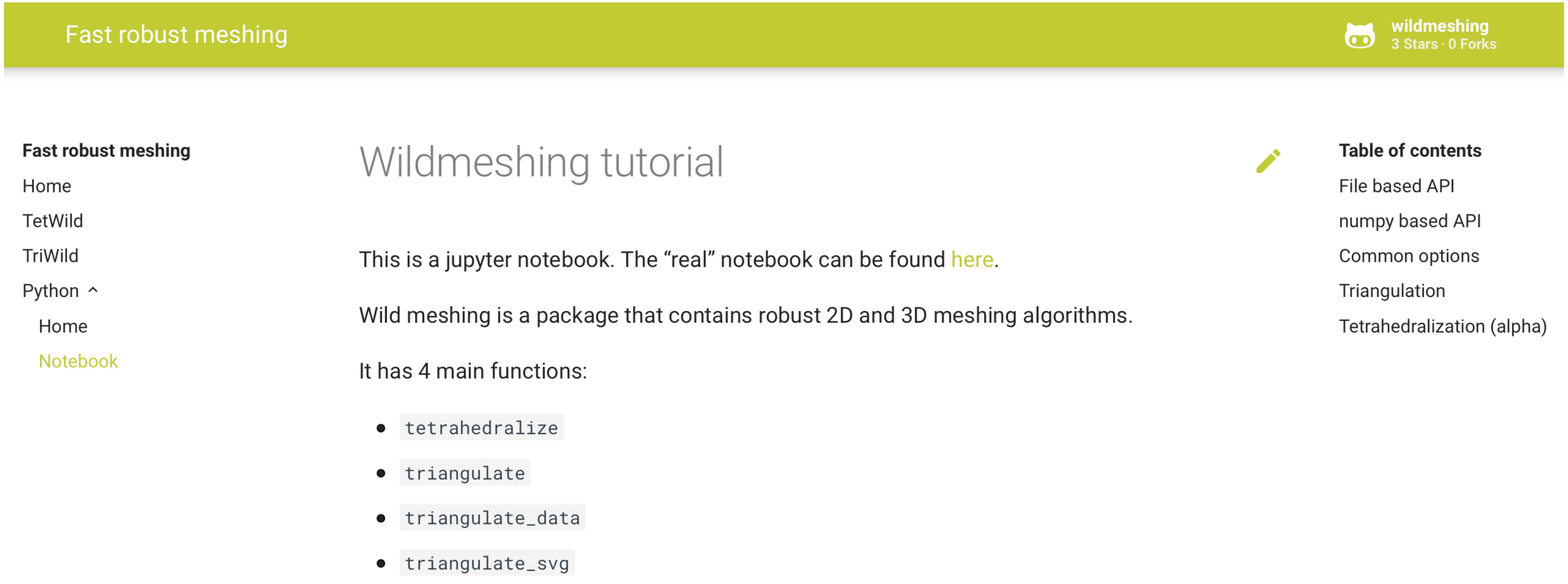
We use the bisection hierarchy to define adaptive quadrature rules for each element.

Software, Datasets, Benchmarks



Geometric Computing with Python
Sebastian Koch, Teseo Schneider, Francis Williams, Chencheng Li, Daniele Panozzo,
SIGGRAPH Course, 2019
[Website]

Wild Meshing (TetWild and TriWild)



Fast robust meshing

wildmeshing
3 Stars · 0 Forks

Fast robust meshing

- Home
- TetWild
- TriWild
- Python
  - Home
  - Notebook

Wildmeshing tutorial

This is a jupyter notebook. The “real” notebook can be found here.

Wild meshing is a package that contains robust 2D and 3D meshing algorithms.

It has 4 main functions:

- `tetrahedralize`
- `triangulate`
- `triangulate_data`
- `triangulate_svg`

Table of contents

- File based API
- numpy based API
- Common options
- Triangulation
- Tetrahedralization (alpha)

https://wildmeshing.github.io

Collisions

README.md

IPC Toolkit

Build passing | Python passing | Docs passing | license MIT

A set of reusable functions to integrate IPC into an existing simulation.

For a complete list of changes, please see `CHANGELOG.md`, and for a definitive reference for these functions, please see the IPC source code.

Integrating the IPC Toolkit into your project

1. Add it to CMake

The easiest way to add the toolkit to an existing CMake project is to download it through CMake. CMake provides functionality for doing this called FetchContent (requires CMake ≥ 3.14). We use this same process to download all external dependencies. For example,

```
include(FetchContent)
FetchContent_Declare(
    ipc_toolkit
    GIT_REPOSITORY https://github.com/ipc-sim/ipc-toolkit.git
    GIT_TAG ${IPC_TOOLKIT_GIT_TAG}
    GIT_SHALLOW TRUE
)
FetchContent_MakeAvailable(ipc_toolkit)
```

where `IPC_TOOLKIT_GIT_TAG` is set to the version of the toolkit you want to use. This will download and add the toolkit to CMake. The toolkit can then be linked against using

README.md

Tight-Inclusion Continuous Collision Detection

Build passing

Root-Parity
Tight-Inclusion

A conservative continuous collision detection (CCD) method with support for minimum separation.

You can read more about this work in our ACM Transactions on Graphics paper:
"A Large Scale Benchmark and an Inclusion-Based Algorithm for Continuous Collision Detection"

Compiling Instruction

To compile the code, first make sure CMake is installed.

To build the library on Linux or macOS:

```
mkdir build
cd build
cmake ../ -DCMAKE_BUILD_TYPE=Release
make
```

Then you can run a CCD example:

```
./Tight_Inclusion_bin
```

https://github.com/ipc-sim/ipc-toolkit

https://github.com/Continuous-Collision-Detection/Tight-Inclusion

PolyFEM

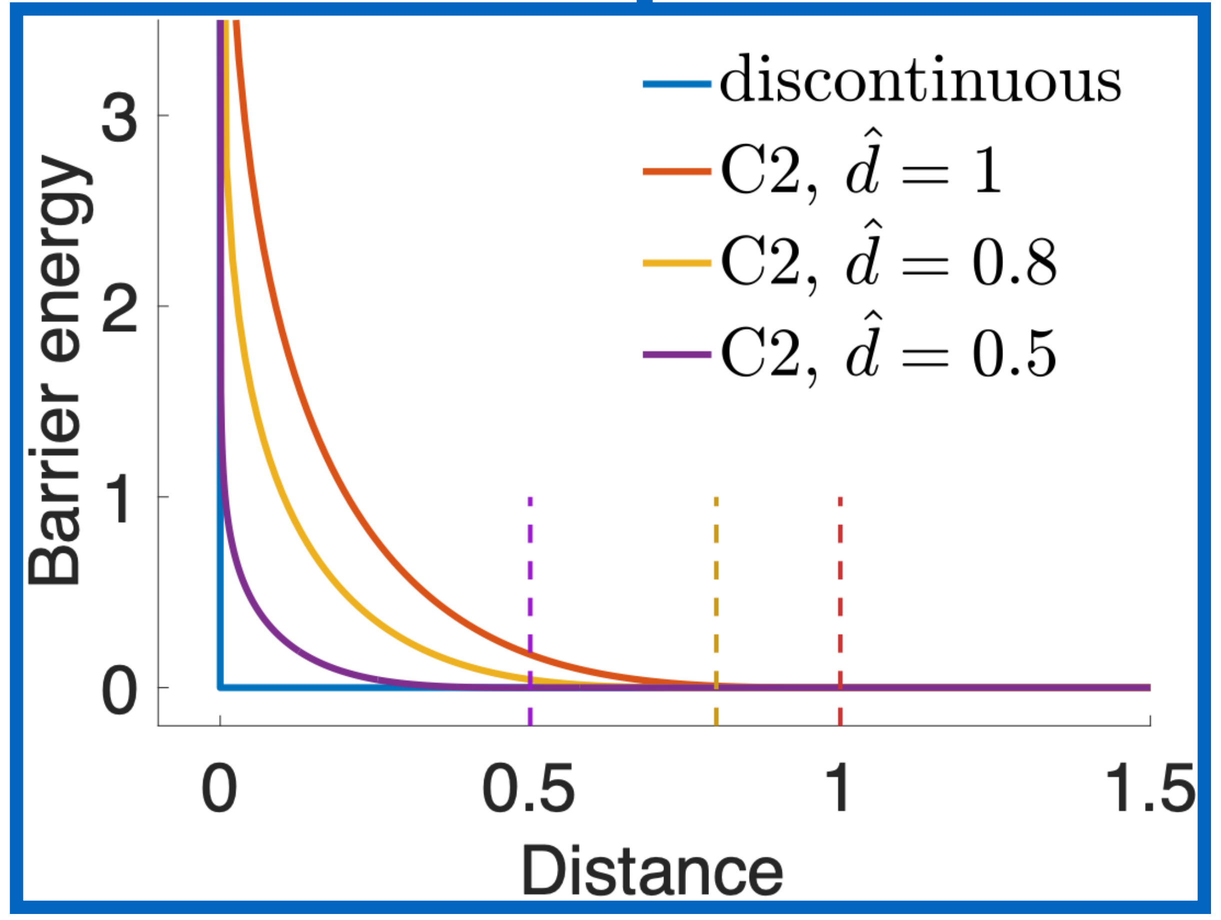
https://polyfem.github.io

Concluding Remarks

Incremental Potential Formulation

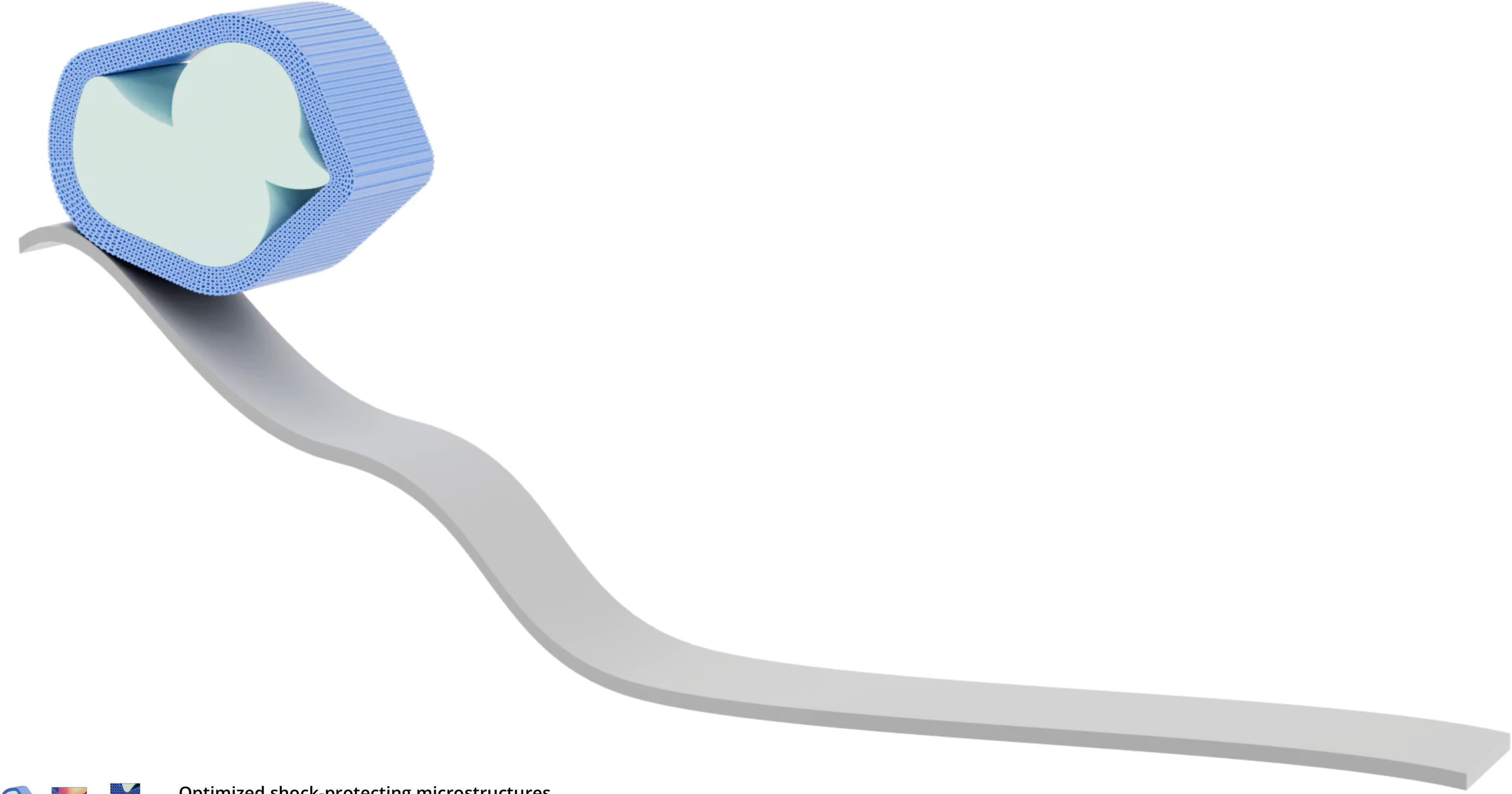
$$x^{t+1} = \underset{x}{\operatorname{argmin}}\, E(x) + B(x, \hat{d}) + D(x)$$

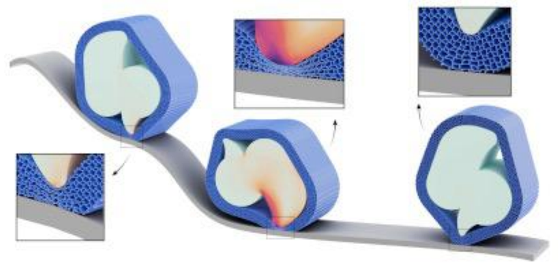
Inertia & Elasticity



Friction

Minimize using Newton's method, with line search to ensure lack of constraints violation



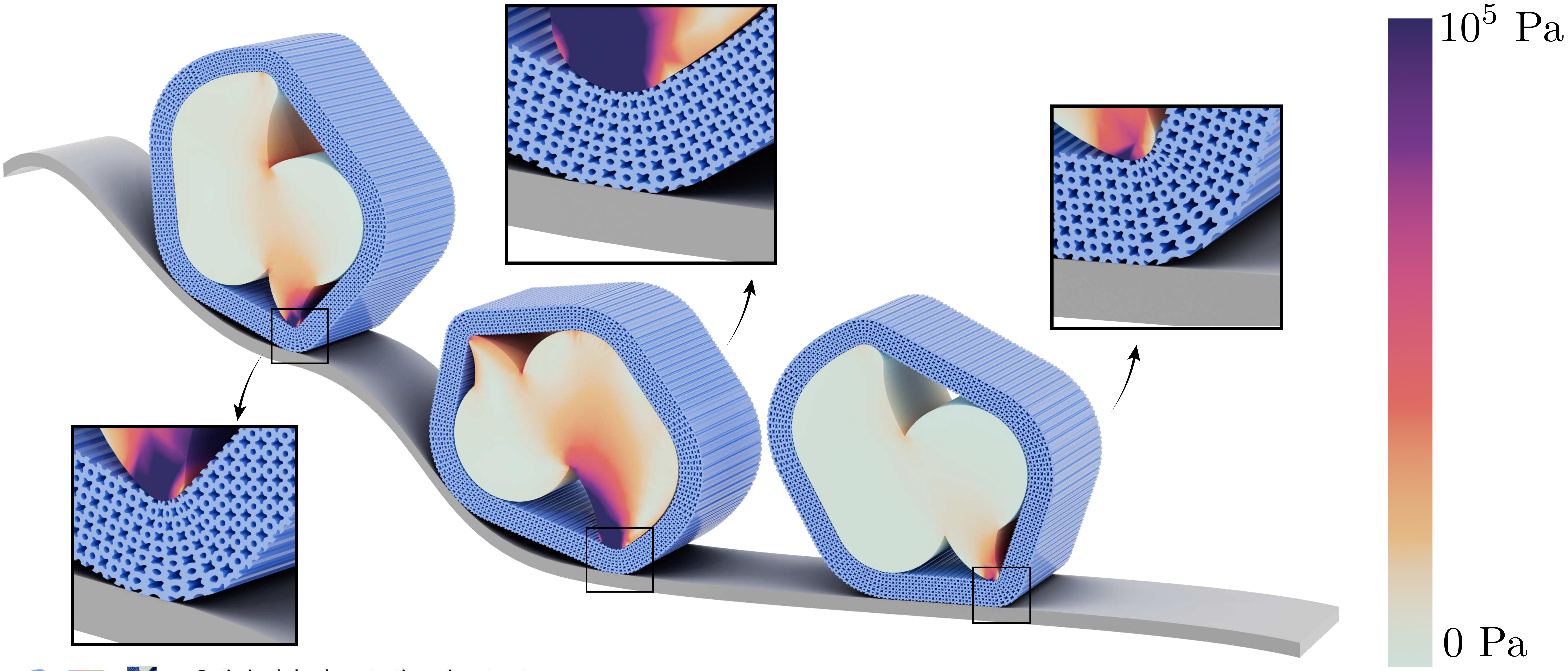


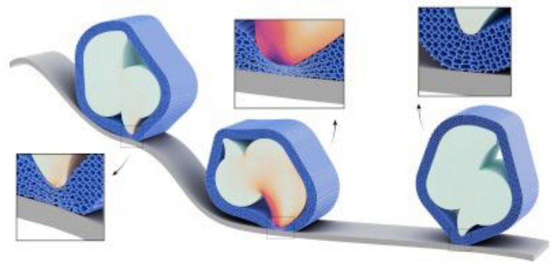
Optimized shock-protecting microstructures

Zizhou Huang, Daniele Panozzo, Denis Zorin,

ACM Transaction on Graphics (SIGGRAPH Asia), 2024

[Paper] [Video] [Code]



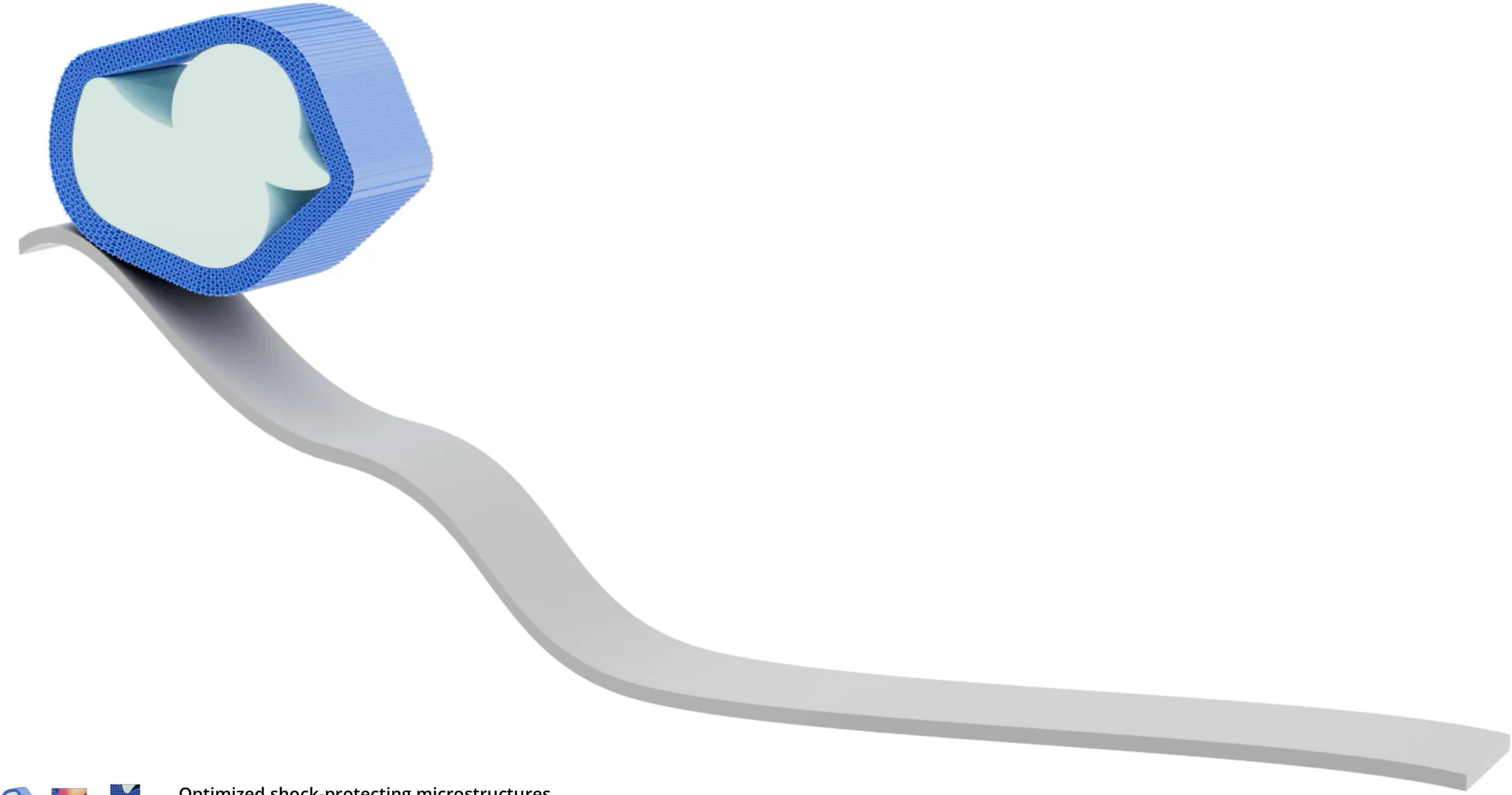


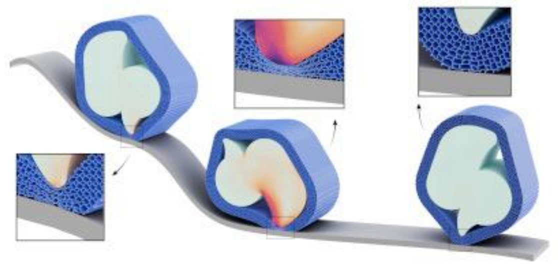
Optimized shock-protecting microstructures

Zizhou Huang, Daniele Panozzo, Denis Zorin,

ACM Transaction on Graphics (SIGGRAPH Asia), 2024

[Paper] [Video] [Code]



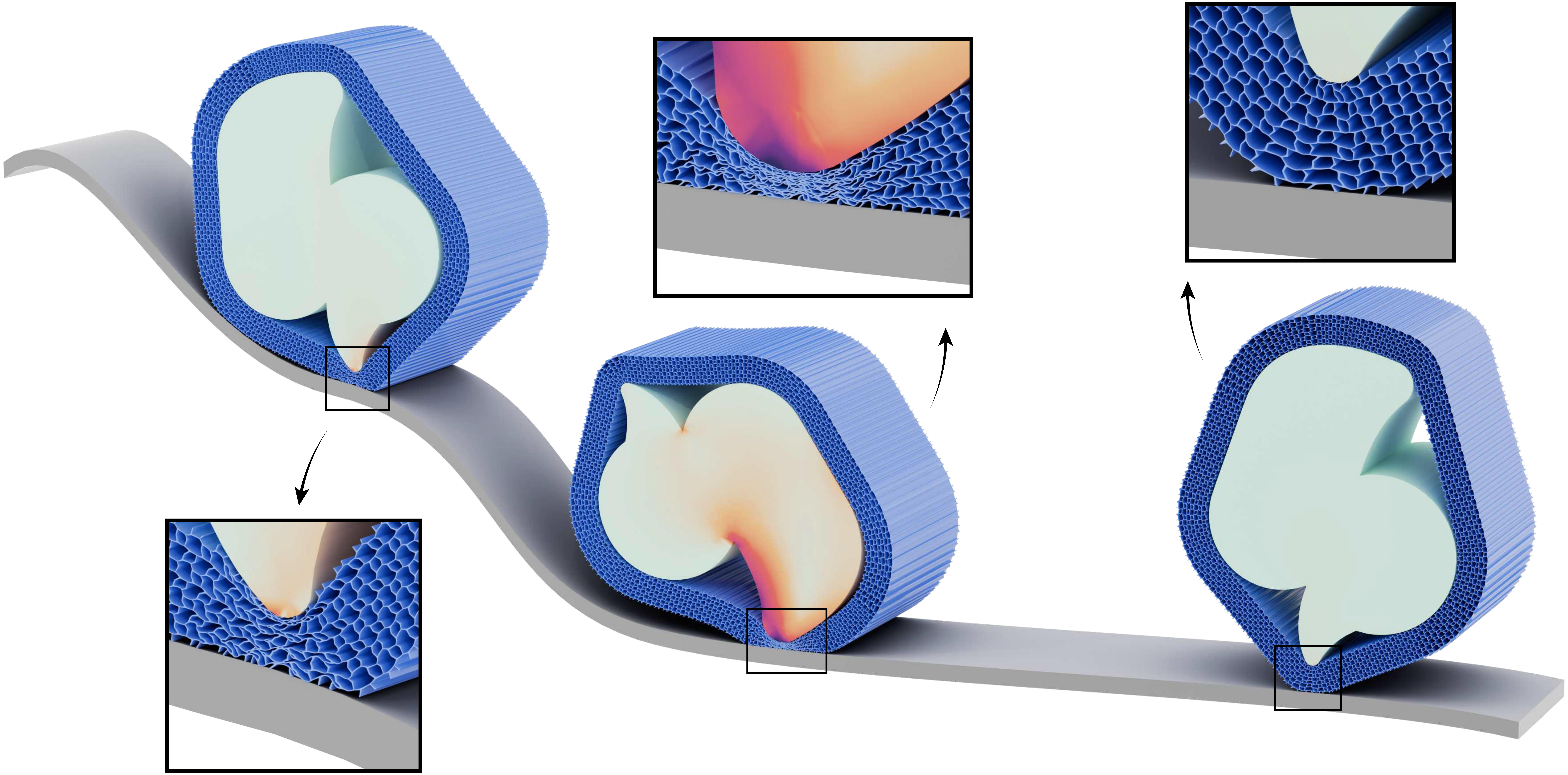


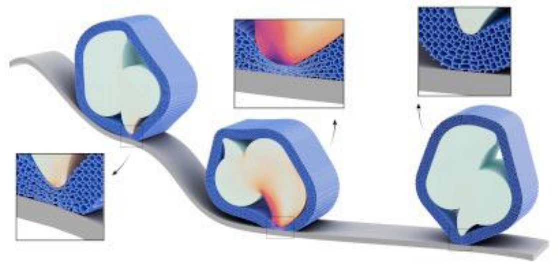
Optimized shock-protecting microstructures

Zizhou Huang, Daniele Panozzo, Denis Zorin,

ACM Transaction on Graphics (SIGGRAPH Asia), 2024

[Paper] [Video] [Code]





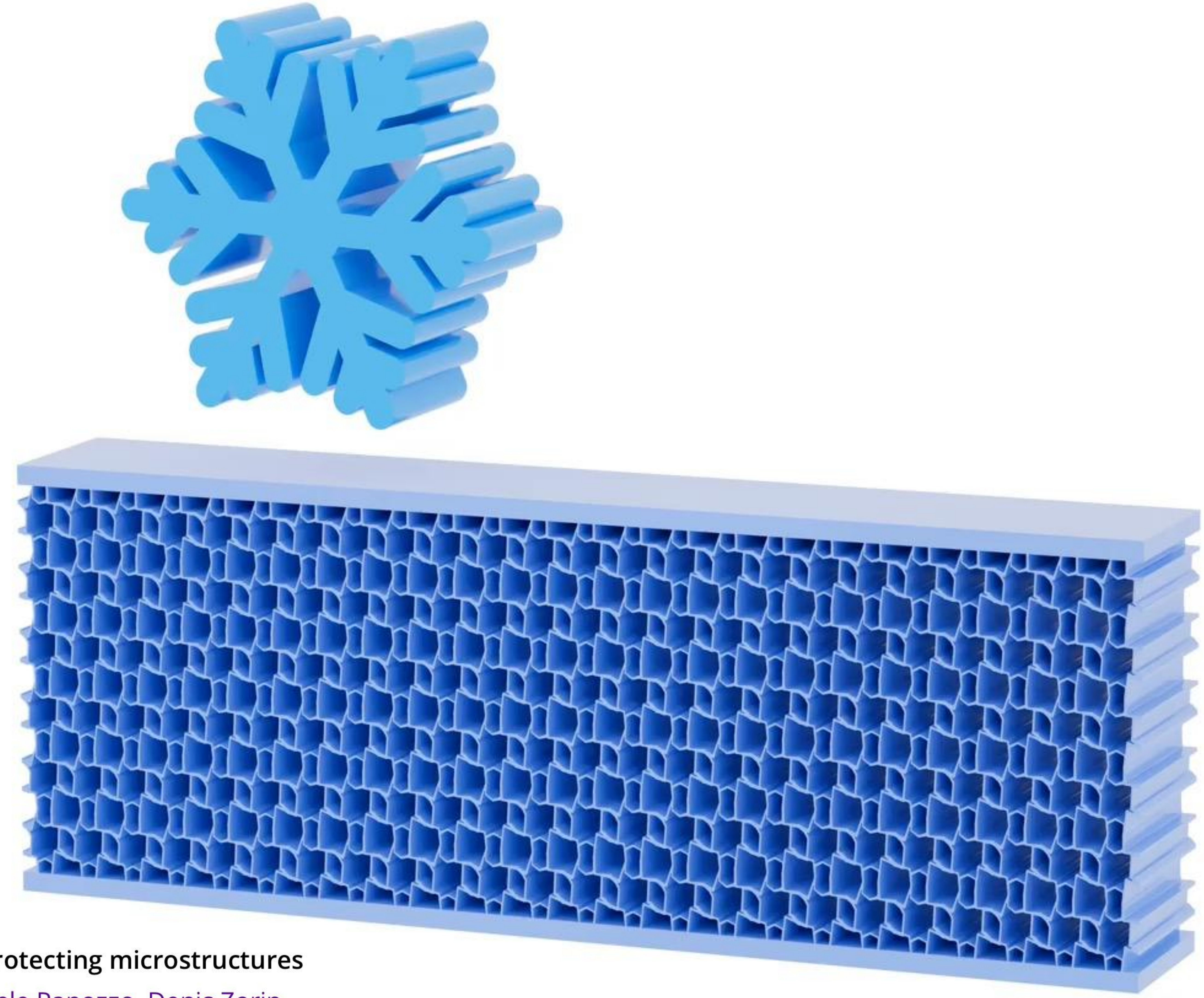
Optimized shock-protecting microstructures

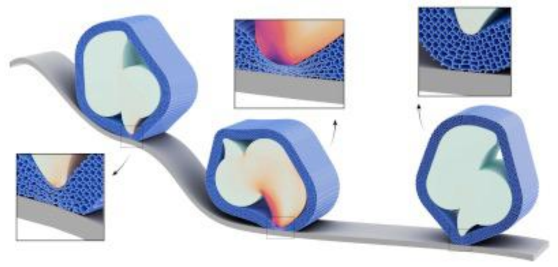
Zizhou Huang, Daniele Panozzo, Denis Zorin,

ACM Transaction on Graphics (SIGGRAPH Asia), 2024

[Paper] [Video] [Code]

Stress is reduced by 72%



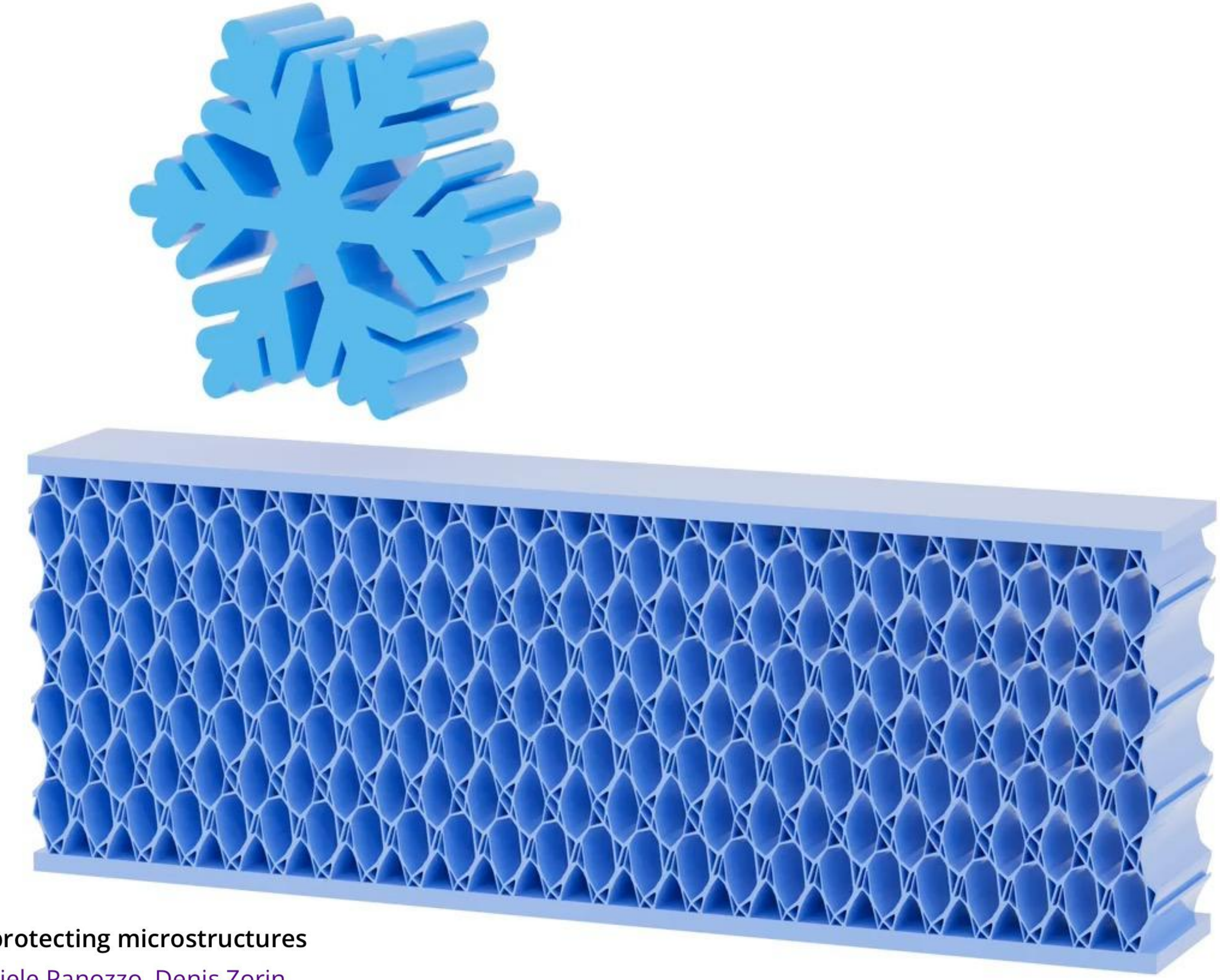


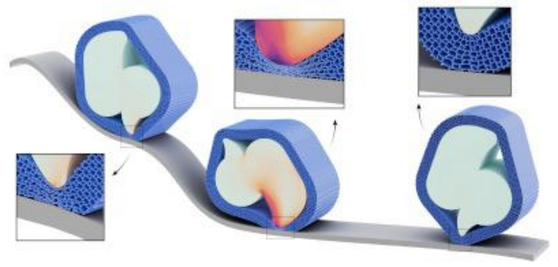
Optimized shock-protecting microstructures

Zizhou Huang, Daniele Panozzo, Denis Zorin,

ACM Transaction on Graphics (SIGGRAPH Asia), 2024

[Paper] [Video] [Code]





Optimized shock-protecting microstructures

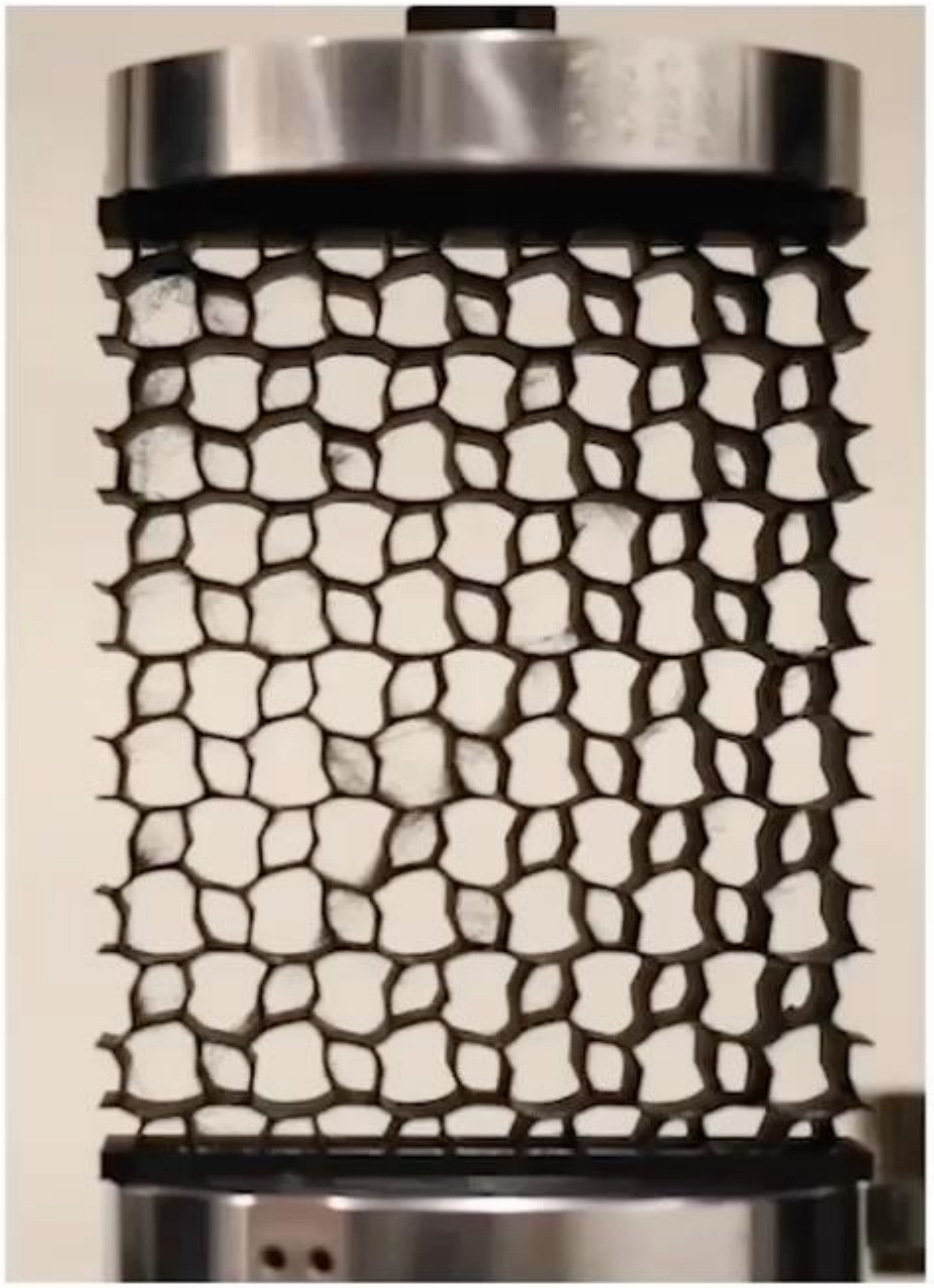
Zizhou Huang, Daniele Panozzo, Denis Zorin,

ACM Transaction on Graphics (SIGGRAPH Asia), 2024

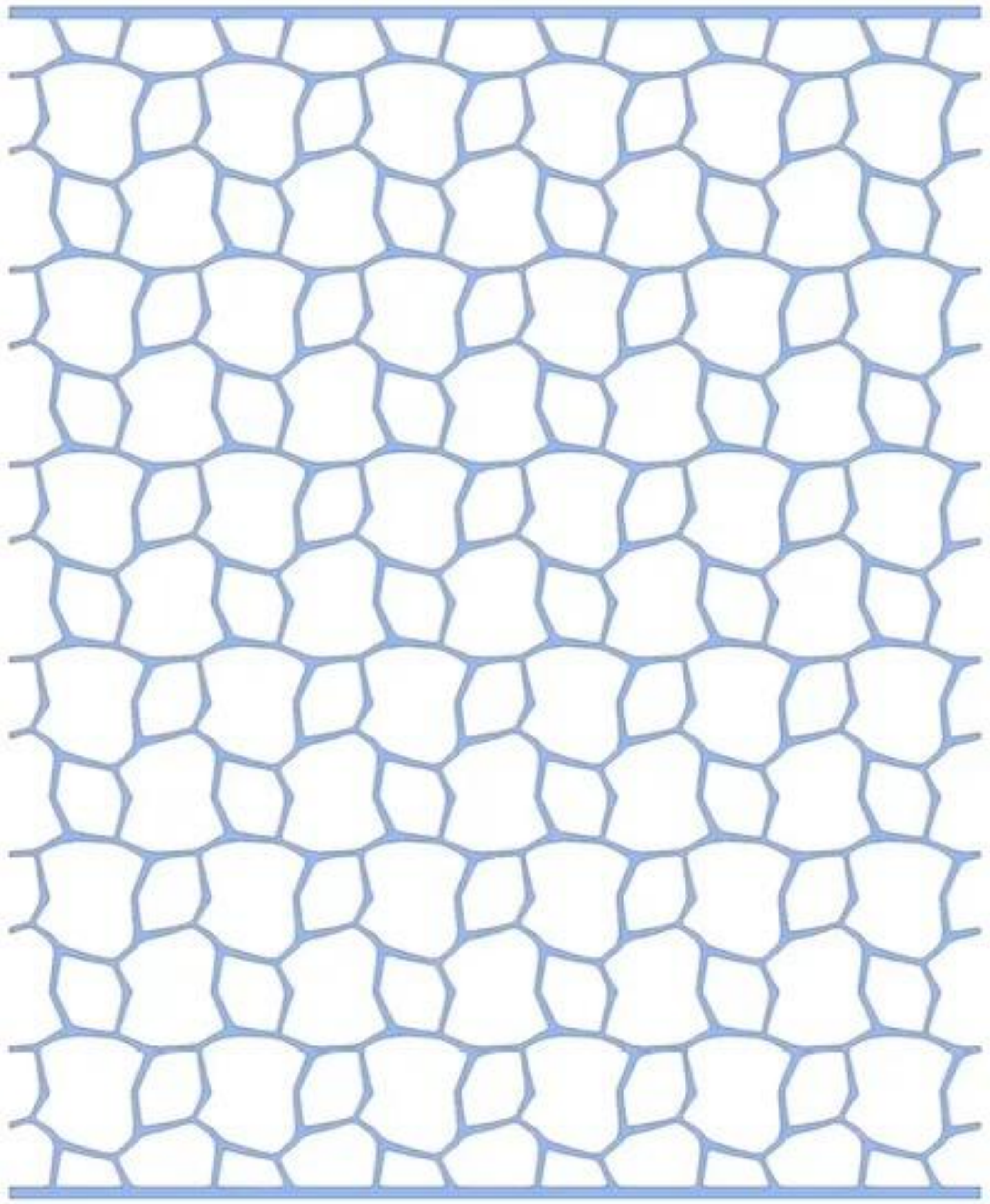
[Paper] [Video] [Code]

Video accelerated

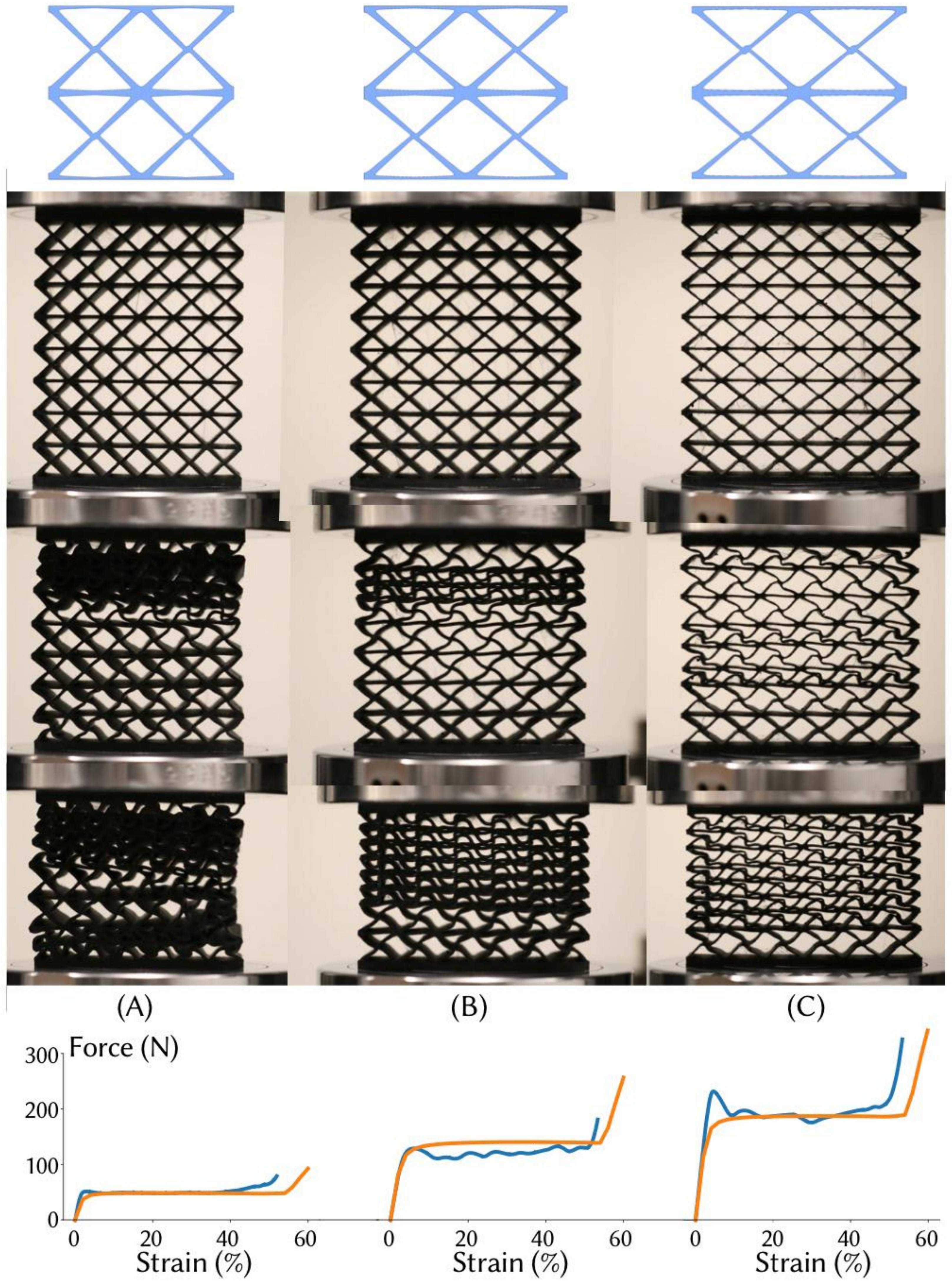
Compression test



Experiment



Full simulation



Geometric Predicates for Unconditionally Robust Elastodynamics Simulation

Daniele Panozzo
NYU Geometric Computing Lab