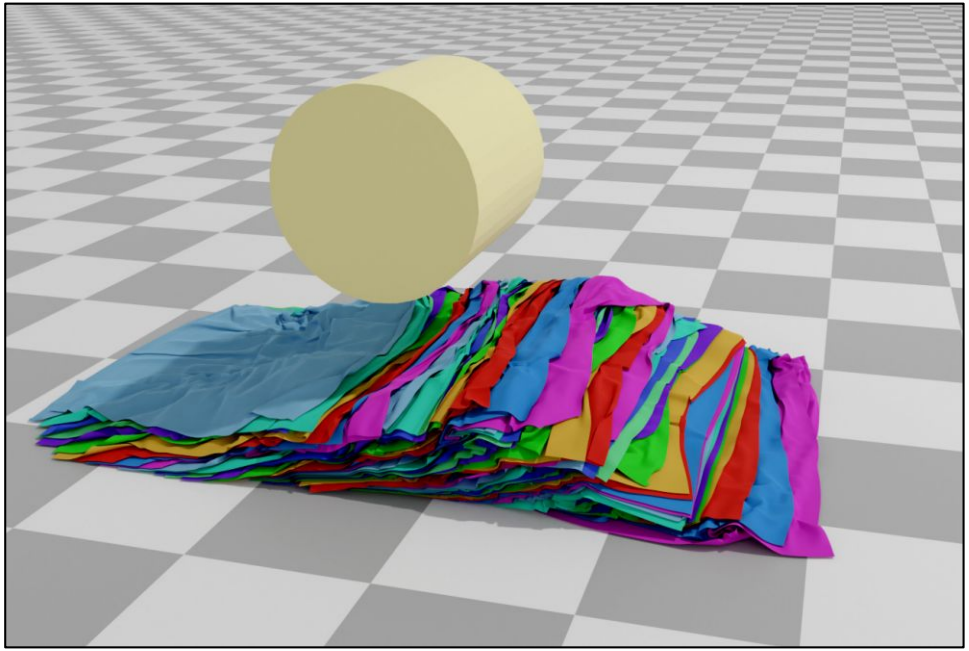
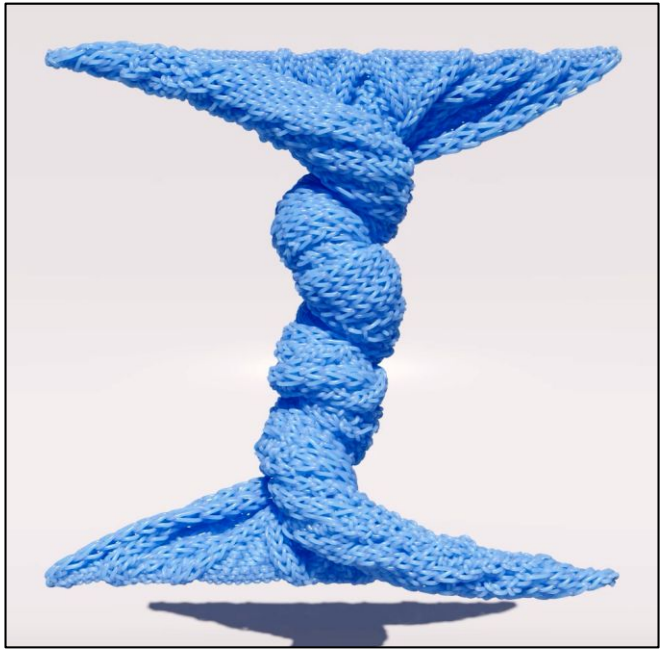
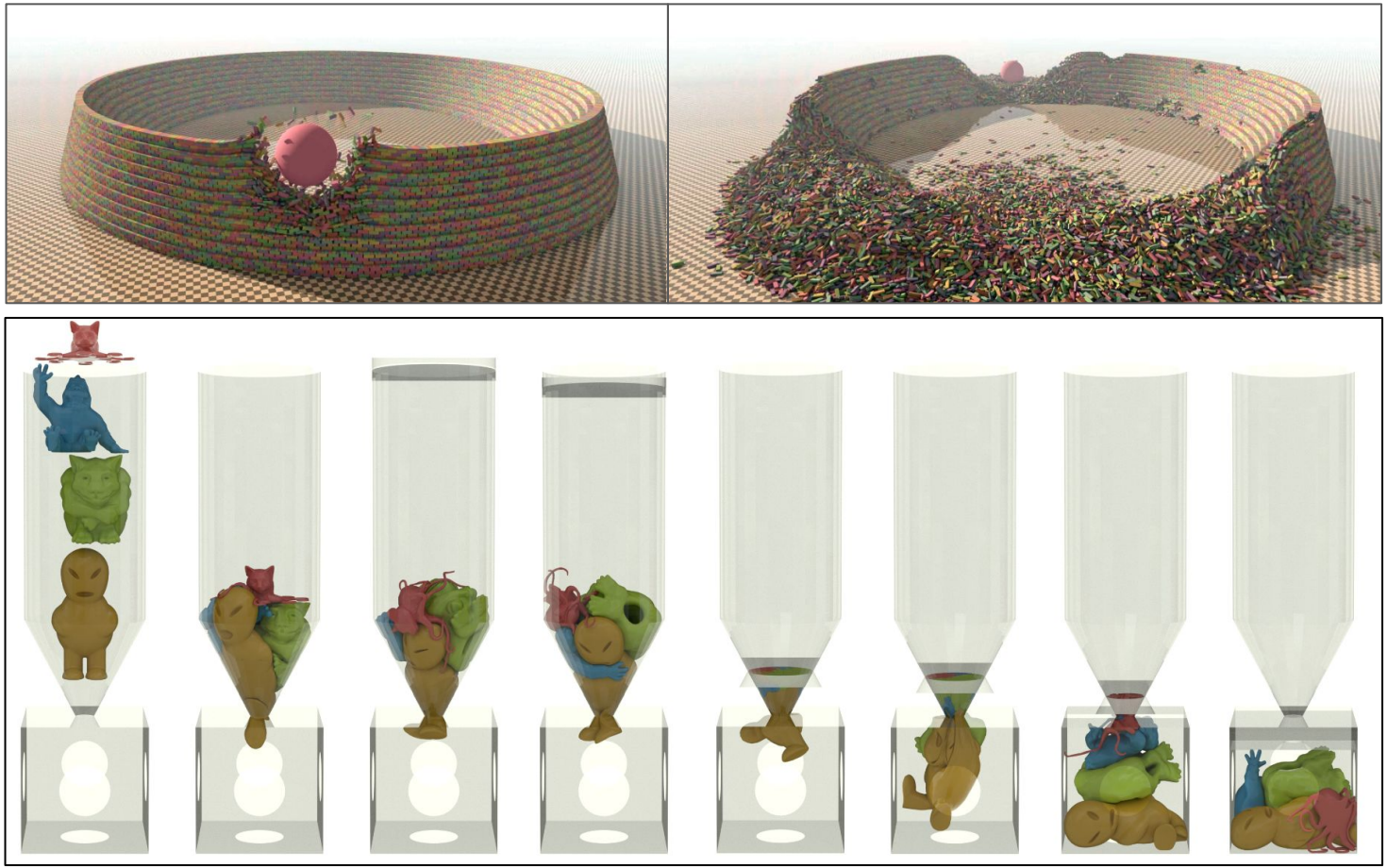
Offset Geometric Contact

Anka He Chen, Jerry HSU, Ziheng Liu, Miles Macklin, Yin Yang, Cem Yuksel [2026]





Problematic : Collisions



Incremental Potential Contact

Incremental Potential Contact: Intersection- and Inversion-free, Large-Deformation Dynamics

MINCHEN LI, University of Pennsylvania & Adobe Research
ZACHARY FERGUSON and TESEO SCHNEIDER, New York University
TIMOTHY LANGLOIS, Adobe Research
DENIS ZORIN and DANIELE PANOZZO, New York University
CHENFANFU JIANG, University of Pennsylvania
DANNY M. KAUFMAN, Adobe Research

Fig. 1. **Squeeze out:** Incremental Potential Contact (IPC) enables high-rate time stepping, here with $h = 0.01$s, of extreme nonlinear elastodynamics with contact that is intersection- and inversion-free at all time steps, irrespective of the degree of compression and contact. Here a plate compresses and then forces a collection of complex soft elastic FE models (181K tetrahedra in total, with a neo-Hookean material) through a thin, codimensional obstacle tube. The models are then compressed entirely together forming a tight mush to fit through the gap and then once through they cleanly separate into a stable pile.

Contacts weave through every aspect of our physical world, from daily household chores to acts of nature. Modeling and predictive computation of these phenomena for solid mechanics is important to every discipline concerned with the motion of mechanical systems, including engineering and animation. Nevertheless, efficiently time-stepping accurate and consistent simulations of real-world contacting elastica remains an outstanding computational challenge. To model the complex interaction of deforming solids in contact we propose Incremental Potential Contact (IPC) – a new model and algorithm for variationally solving implicitly time-stepped nonlinear elastodynamics. IPC maintains an intersection- and inversion-free trajectory regardless of material parameters, time step sizes, impact velocities, severity of deformation, or boundary conditions enforced.

Constructed with a custom nonlinear solver, IPC enables efficient resolution of time-stepping problems with separate, user-exposed accuracy tolerances that allow independent specification of the physical accuracy of the dynamics and the geometric accuracy of surface-to-surface conformation. This enables users to decouple, as needed per application, desired accuracies for a simulation's dynamics and geometry.

The resulting time stepper solves contact problems that are intersection-free (and thus robust), inversion-free, efficient (at speeds comparable to or faster than available methods that lack both convergence and feasibility), and accurate (solved to user-specified accuracies). To our knowledge this is the first implicit time-stepping method, across both the engineering and graphics literature that can consistently enforce these guarantees as we vary simulation parameters.

In an extensive comparison of available simulation methods, research libraries and commercial codes we confirm that available engineering and computer graphics methods, while each succeeding admirably in custom-tuned regimes, often fail with instabilities, egregious constraint violations and/or inaccurate and implausible solutions, as we vary input materials, contact numbers and time step. We also exercise IPC across a wide range

Authors' addresses: Minchen Li, University of Pennsylvania & Adobe Research, minchenli@gmail.com; Zachary Ferguson; Teseo Schneider, New York University; Timothy Langlois, Adobe Research; Denis Zorin; Daniele Panozzo, New York University; Chenfanfu Jiang, University of Pennsylvania; Danny M. Kaufman, Adobe Research, dannykaufman@gmail.com.

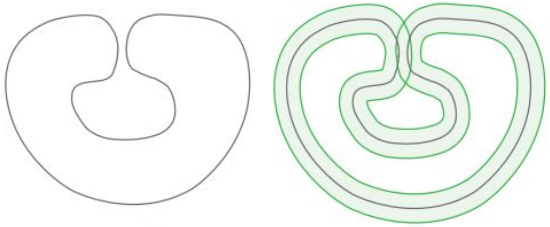
0730-0301/2020/7-ART49 $15.00
https://doi.org/10.1145/3386569.3392425

How to guarantee penetration free collisions with energy based approach ?

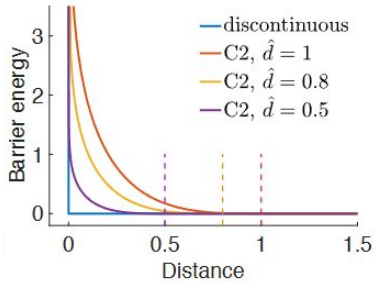
Energy based solve	(Vertex) collision energy
$x^{n+1} \leftarrow \underset{x}{arg\,min}\ G(x)$	$\sum_{a \in \mathcal{F}(v)} g(dis(\mathbf{x}_v, a), r)$

needs smooth energy ⇔ can't represent infinitely stiff collision

contact mesh
volumetric mesh with radius



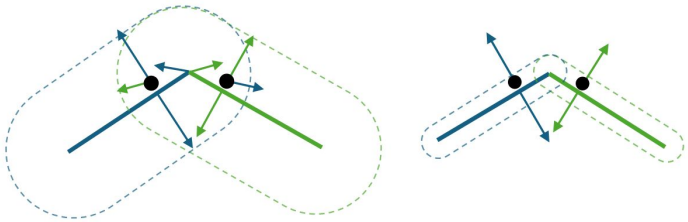
g ⇒ barrier function



collision forces are strong enough to avoid collision when **dis($\mathbf{x}_v$**, a) is small

Why is IPC slow ?

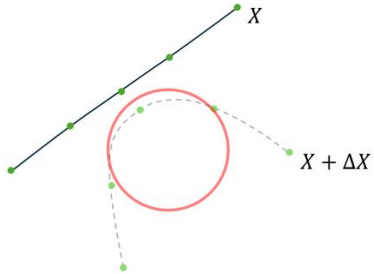
Correction directions



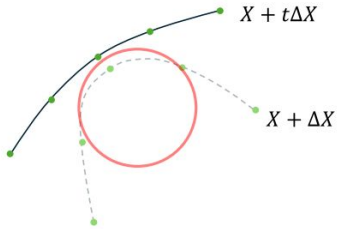
(c) *IPC with large contact radius*

(d) *IPC with small contact radius*

Correction step size



(a) *Previous Position and a Full Optimization Step*



(b) *Iteration Using IPC's Scheme*

Contact normal are broken on overlap

↓

Fixed by reducing contact radius

↓

Stiffer contact energy !

Slow convergence !

Can't move particle too far

↓

Very small steps when close

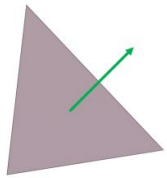
↓

Lots of iterations to achieve convergence

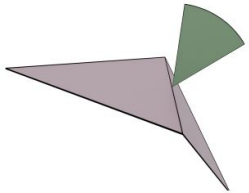
Slow convergence !

Better surface normals definition

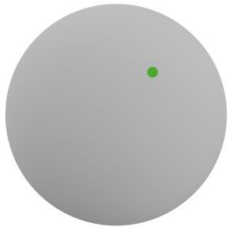
Polyhedral Gauss Map [Gil07] ⇒ Extends Gauss–Bonnet theorem to a polyhedral surface



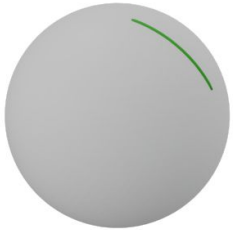
(a) *Face Normals*



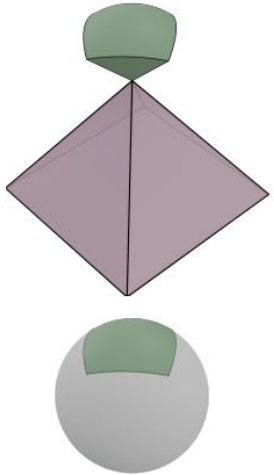
(b) *Edge Normals*



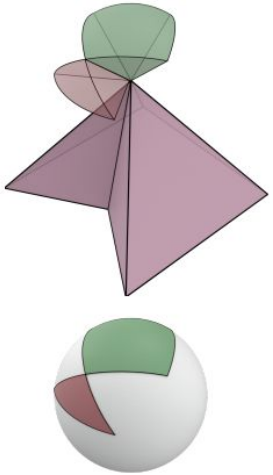
(c) *Face Normal Spherical Indicatrix*



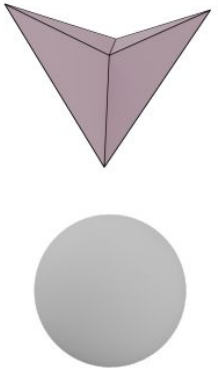
(d) *Edge Normals Spherical Indicatrix*



(a) *Convex Vertex*



(b) *Mixed Vertex*



(c) *Saddle Vertex*

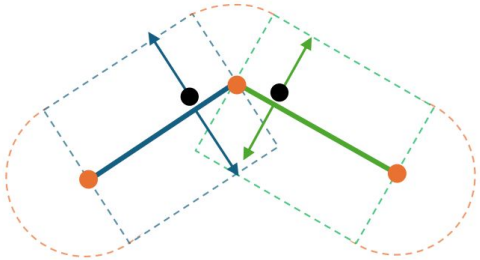
Much better contact mesh

Coherent normals along mesh

[Gil07] - Gilberto Echeverria : The Polyhedral Gauss Map and discrete curvature measures in geometric modelling

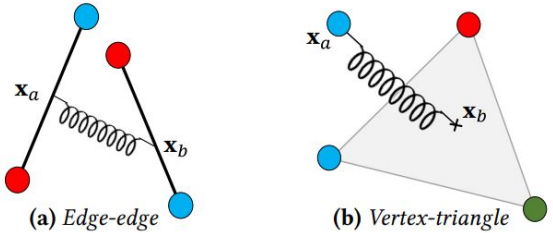
Contact mesh construction

Contact mesh with OCG



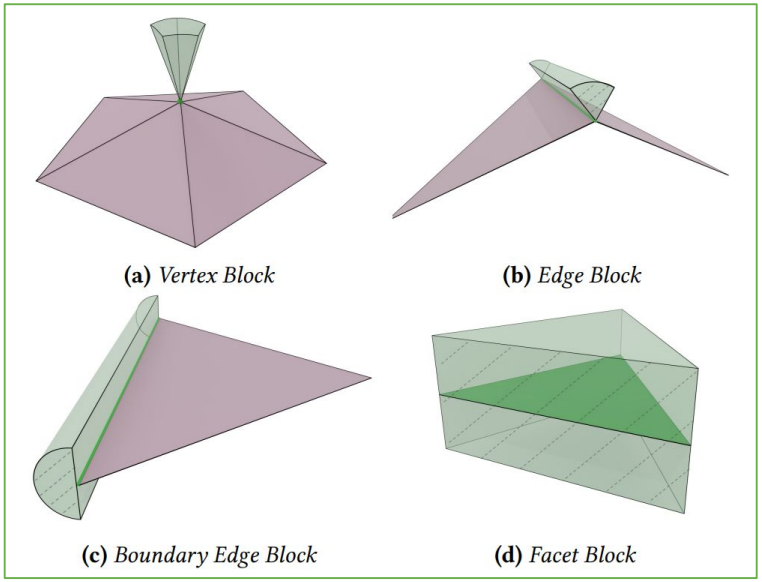
Build in blocks : vertex, edge, boundary edge, face

2 types of collisions



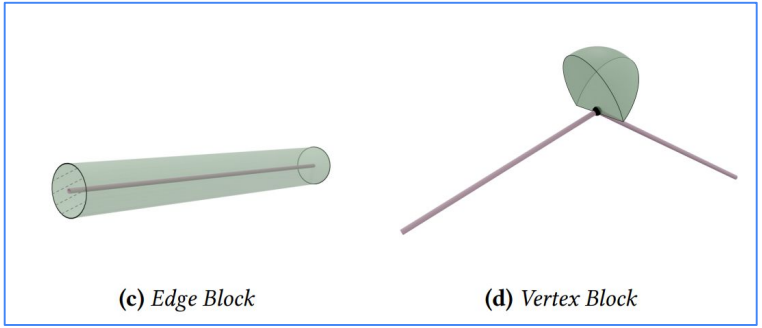
(a) *Edge-edge* **(b)** *Vertex-triangle*

"Wireframe"	Surface



(a) *Vertex Block* **(b)** *Edge Block*

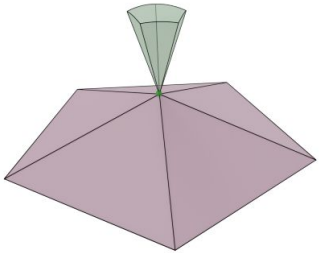
(c) *Boundary Edge Block* **(d)** *Facet Block*



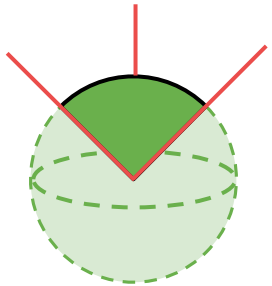
(c) *Edge Block* **(d)** *Vertex Block*

Surface Blocks

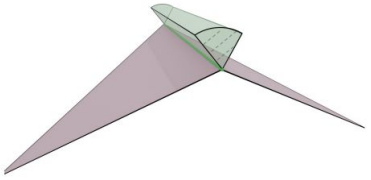
4 primitives : sphere (vertex), cylinder (edge + border edge), prism (face)



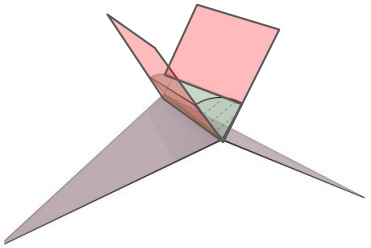
(a) *Vertex Block*



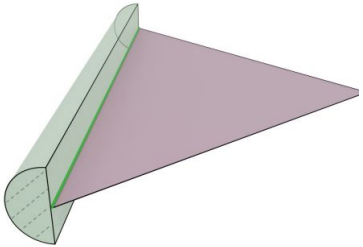
sphere
cut by n planes



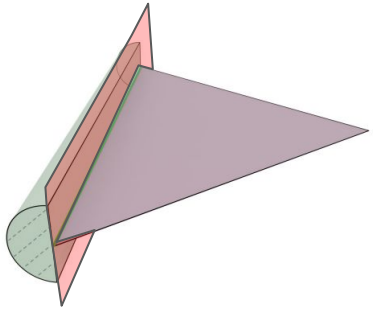
(b) *Edge Block*



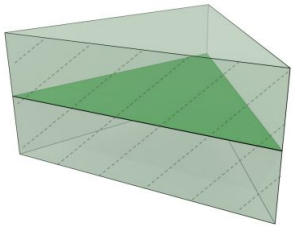
cylinder
cut by 2 planes



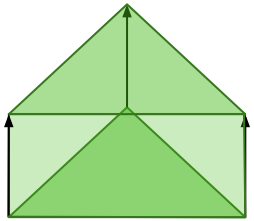
(c) *Boundary Edge Block*



cylinder
cut by 1 planes



(d) *Facet Block*

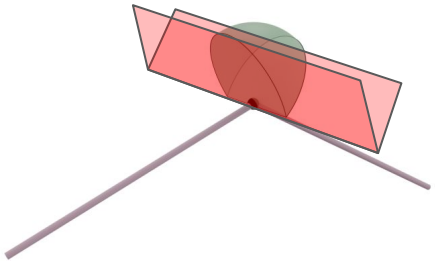


Prism
face extrusion

Wireframe Blocks

2 primitives : sphere (vertex), cylinder (edge)

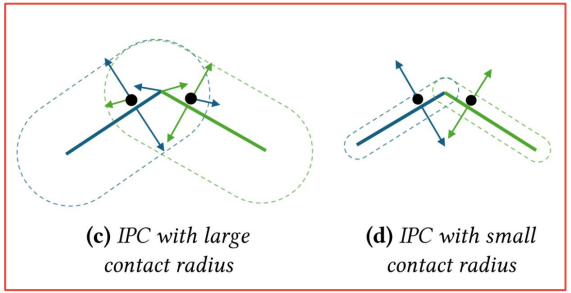
(c) *Edge Block*



(d) *Vertex Block*

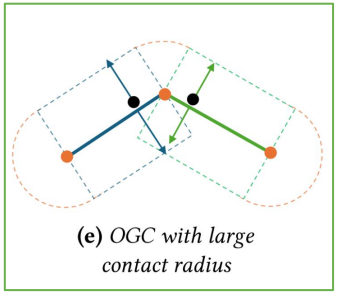
You get the idea

- Nice contact normal formulation
- Contact mesh block : sphere, cylinder, prism
- Build 2 contact mesh (surface + wireframe)



(c) *IPC with large contact radius*

(d) *IPC with small contact radius*



(e) *OGC with large contact radius*

Collision energy

Vertex (v) collision energy

$$\sum_{a \in \mathcal{F}(v)} g(dis(\mathbf{x}_v, a), r)$$

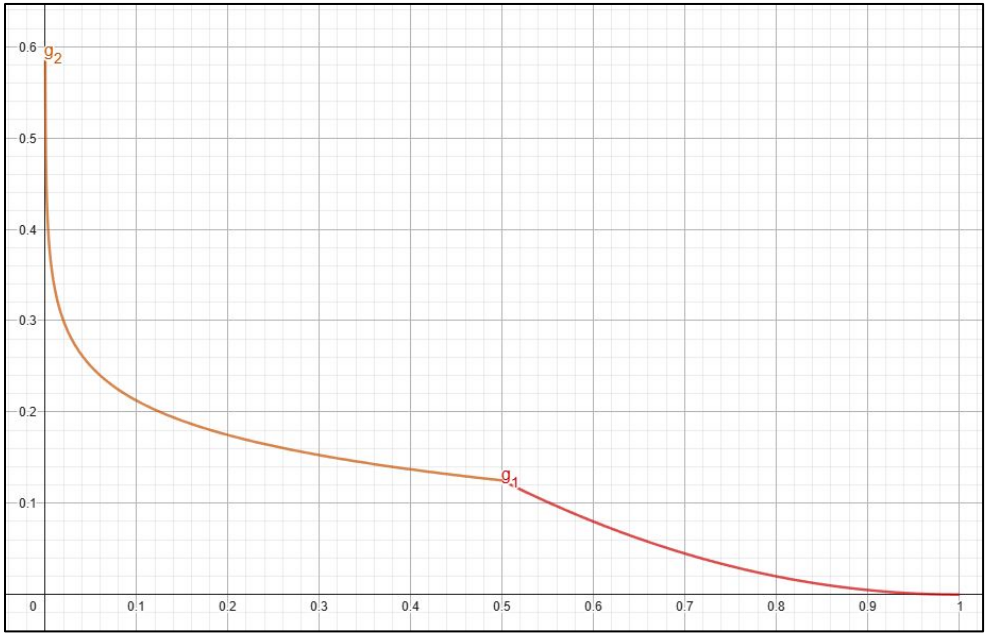
$$g(d,r) = \begin{cases} \frac{k_c}{2}(r-d)^2 & \text{if } \tau \le d \le r \\ -k_c' log(d) + b & \text{if } 0 < d < \tau \end{cases}$$

$$k_c' = \tau k_c (\tau - r)^2$$

$$b = \frac{k_c}{2}(r-\tau)^2 + k_c' log(\tau)$$

$$\tau = \frac{r}{2}$$

$k_c \Rightarrow$ collision stiffness
$r \Rightarrow$ collision mesh radius
$d \Rightarrow$ distance to collision mesh



Combine log (barier and quadratic (spring) energy

Why not ⇒ same idea than [Ando24]

Distance aware correction

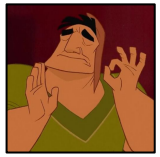
Avoid collision when optimizing

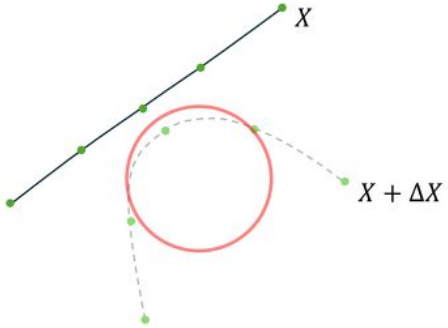
Vertex max distance correction
$b_v = \gamma_p \min(d_{\min,v}, d^E_{\min,v}, d^T_{\min,v})$
Distance to the closest primitive
$\mathbf{x_i} + \frac{b_i}{\|\Delta \mathbf{x_i}\|} \Delta \mathbf{x_i}$

max distance is defined per vertex

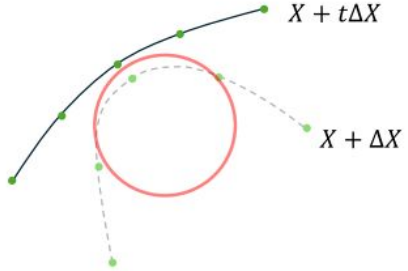
Only small correction for close vertex

Much better convergence !

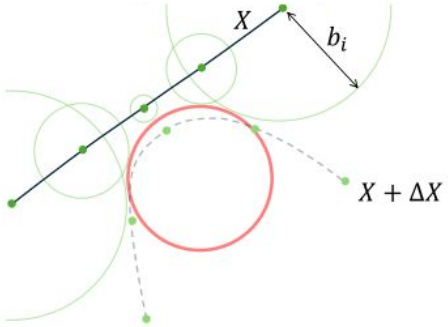




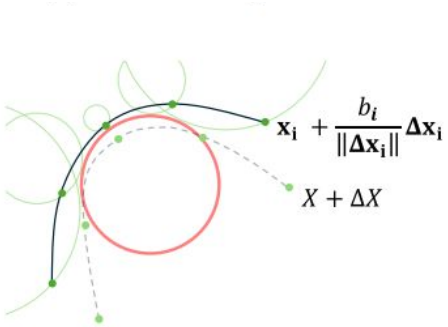
(a) *Previous Position and a Full Optimization Step*



(b) *Iteration Using IPC's Scheme*



(c) *Conservative Bounds*

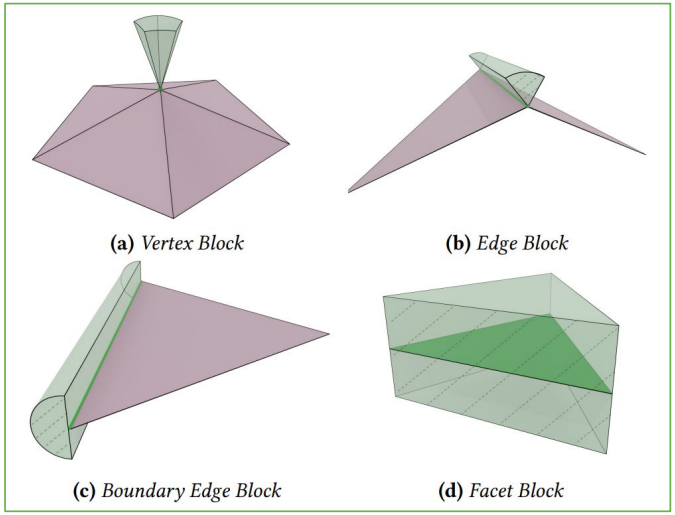


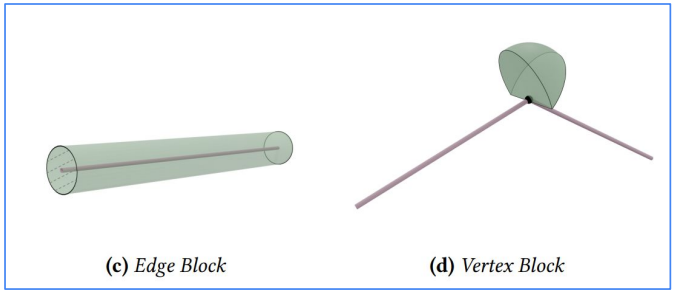
(d) *Iteration Using by Our Scheme*

Some technical details

- Full GPU pipeline
- 2 BVH of blocks (surface + wireframe)
- 1 collision detection per time step
- Avoid collisions during opti with max distance
- Use *VBD* for optimisation [Che24]
- Don't use CCD ⇒ Use max distance

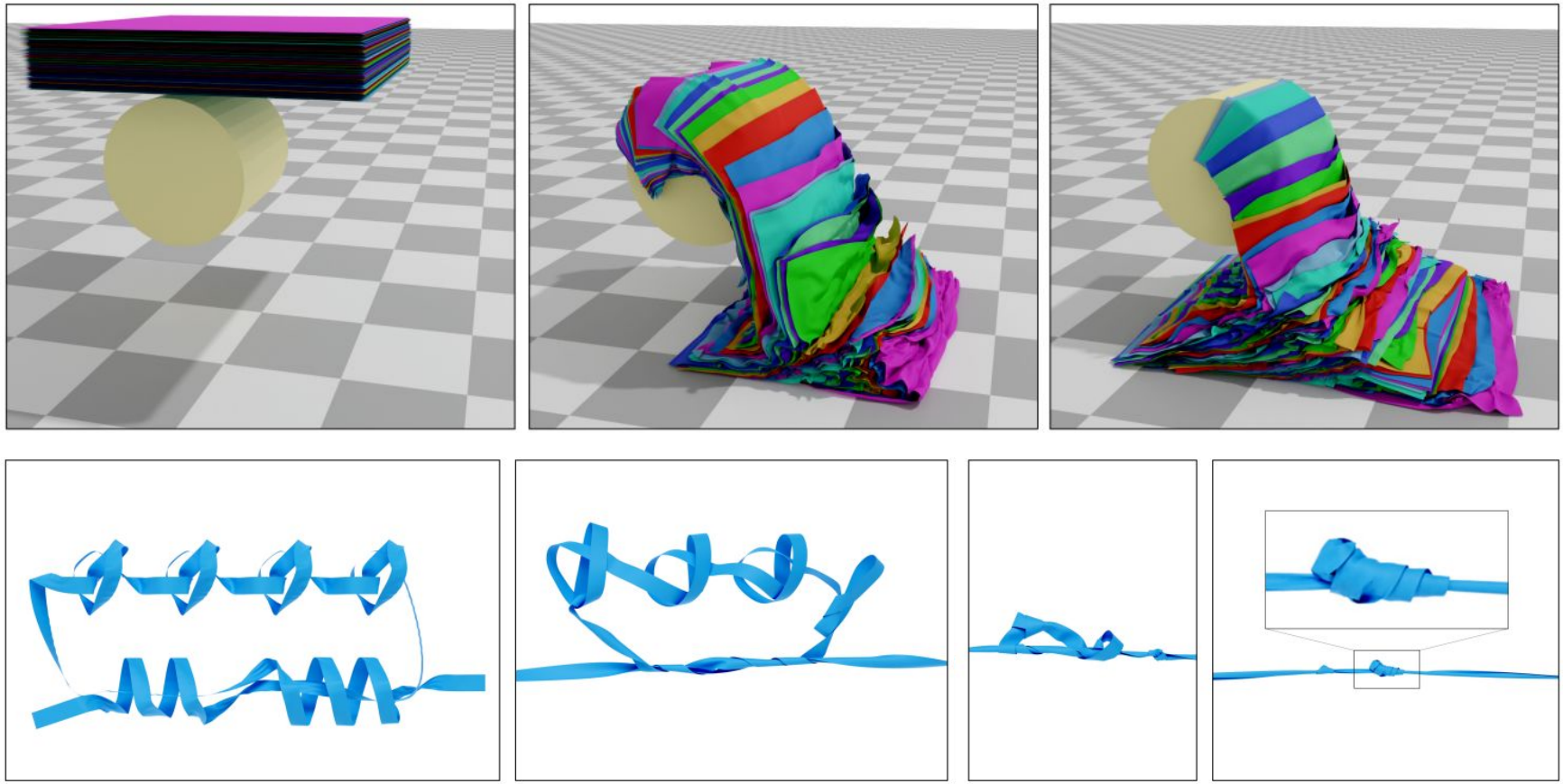
much faster but suppose : $||\Delta x|| < \text{radius}$



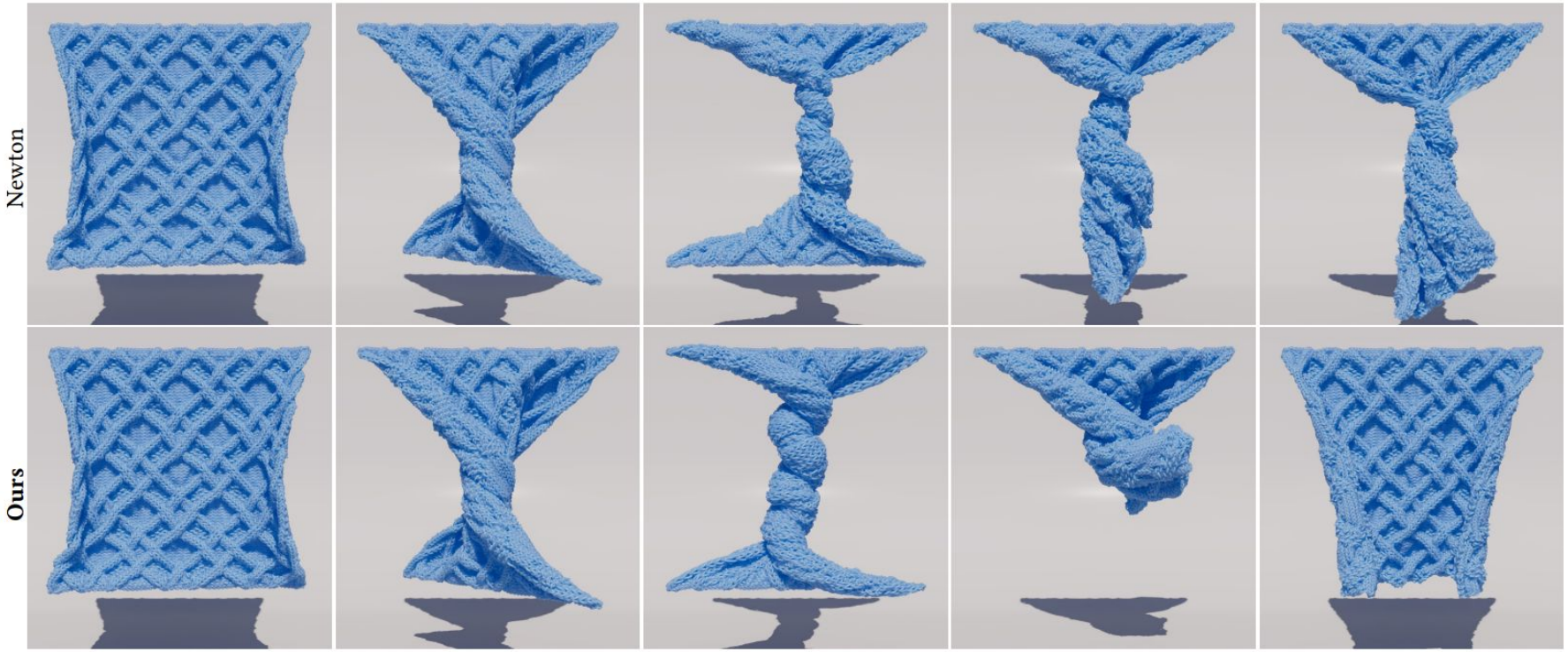


[Che24] - Anka He Chen et al : Vertex Block Descent

Results : Robust



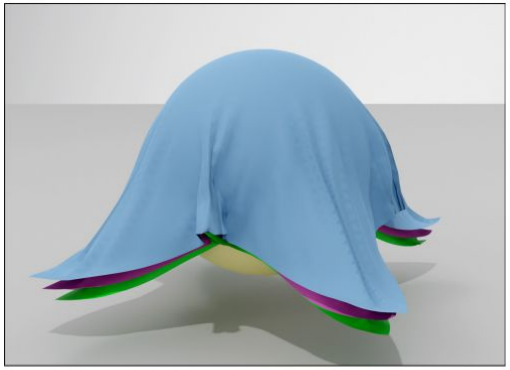
Results : Robust 2

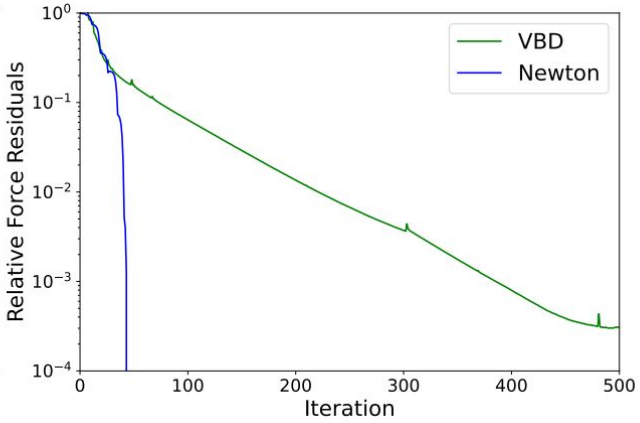


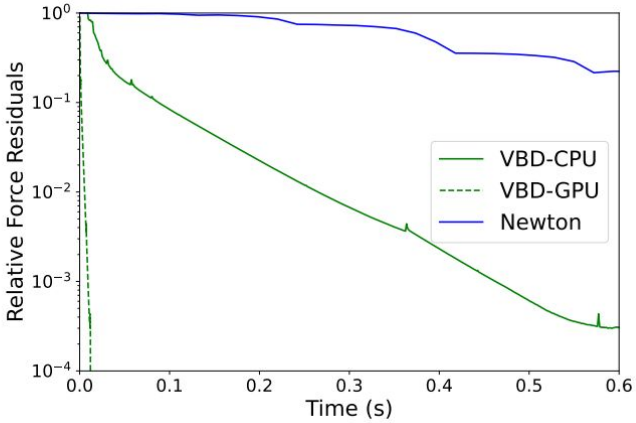
Local optimisation (VBD) guaranty penetration free collision where global resolution methods as Newton can’t

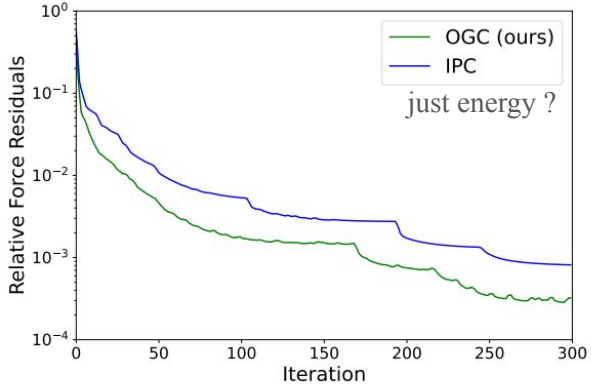
Nice ~

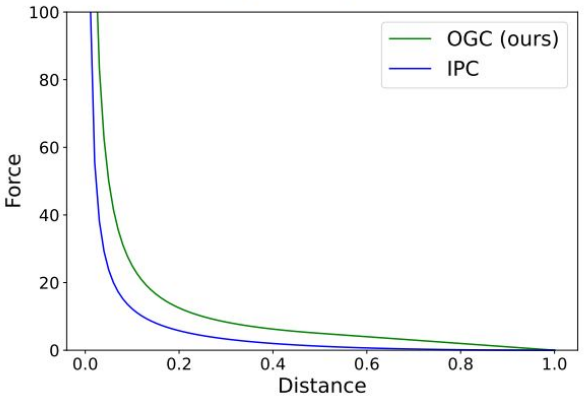
Results : Convergence











Results : some numbers

Experiment Name	Number of		Contact &Fiction			Simulation Parameters		Time per step (avg./max)	
	Vert.	Primitives	k_c	μ_c, ϵ_v	$r(mm)$	Time Step (sec.)	Iterations	CPU VBD	GPU VBD
50 Layers of Cloth (Figure 10)	1M	1.96M	1e5	0.2, 1e-2	2	1/1200	40	0.21/0.55s	6.3/11.5ms
Tightening a knot (Figure 11)	48K	92K	1e5	0.4, 1e-2	2	1/300	50	122/180ms	4.4/6.8ms
Twisting Cloth (Figure 12)	10K	19.6K	1e5	0.2, 1e-2	2	1/300	10	21/30ms	0.9/1.5ms
Cloth on Body (Figure 14)	15.6K	29K	1e5	0.5, 1e-2	2	1/200	20	30/42ms	1.2/1.4ms
Robot and T-shirt (Figure 14)	13.8K	27.4K	1e5	0.5, 1e-2	2	1/600	10	NA	1.8/2.2ms
Yarn Stretch (Figure 15)	65K	65K	2e-3	0.1, 1e-3	1.5	4e-4	4	NA	0.23/0.30ms
Yarn Twist (Figure 16)	65K	65K	2e-3	0.1, 1e-3	1.5	4e-4	4	NA	0.25/0.33ms
3 Layers of Cloth on Sphere (Figure 17)	14.7K	28.6K	1e4	0.5, 1e-2	2	1/100	NA	See figure	NA
1 Layer of Cloth on Sphere (Figure 18)	4.9K	9.5K	1e4	0.5, 1e-2	5	1/100	40	62/75ms	1.2/3.2ms
Twisting Volumetric Mat (Figure 19)	15K	46.8K	1e5	0.2, 1e-2	2	1/240	20	NA	5.5/8.5ms

Real time ?

6.3 - 11.5ms
4.4 - 6.8ms
0.9 - 1.5 ms
1.2 - 1.4 ms
1.8 - 2.2 ms
0.23 - 0.30 ms
0.25 - 0.33 ms
x
1.2 - 3.2 ms
5.5 - 8.5 ms

1 frame = 60Hz

126 - 230ms
5 - 34ms
4.5 - 7.3 ms
4 - 4.6 ms
18 - 22 ms
153 - 200 ms
167 - 220 ms
x
2.0 - 5.4 ms
22 - 34 ms

Hard scenarii needs small dt / more collision detection

↓

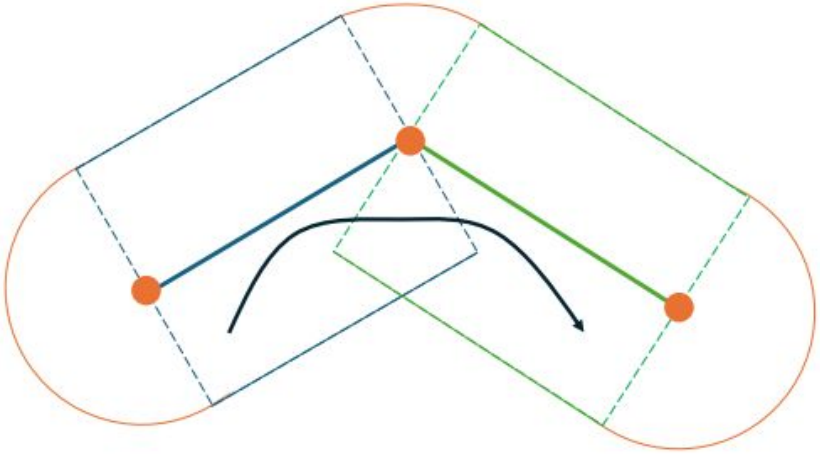
much faster but suppose : $||\Delta x|| < radius$

Globally impressive !

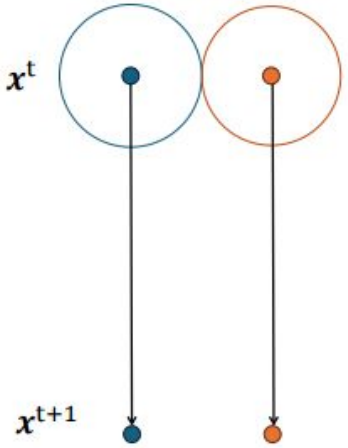
2 order of magnitude faster than IPC

Quite robust (if dt small enough)

Limitations



(a) *Discontinuity*



(b) *High velocity*

Video time