On the Velocity of an Implicit Surface

Jos **Stam** and Ryan **Schmidt**

Autodesk, Inc.

University of Toronto

Toronto, Canada

Research in Industry

Cool Math → Application

Application → Cool Math

Research in Industry

Cool Math → Application

Application → Cool Math

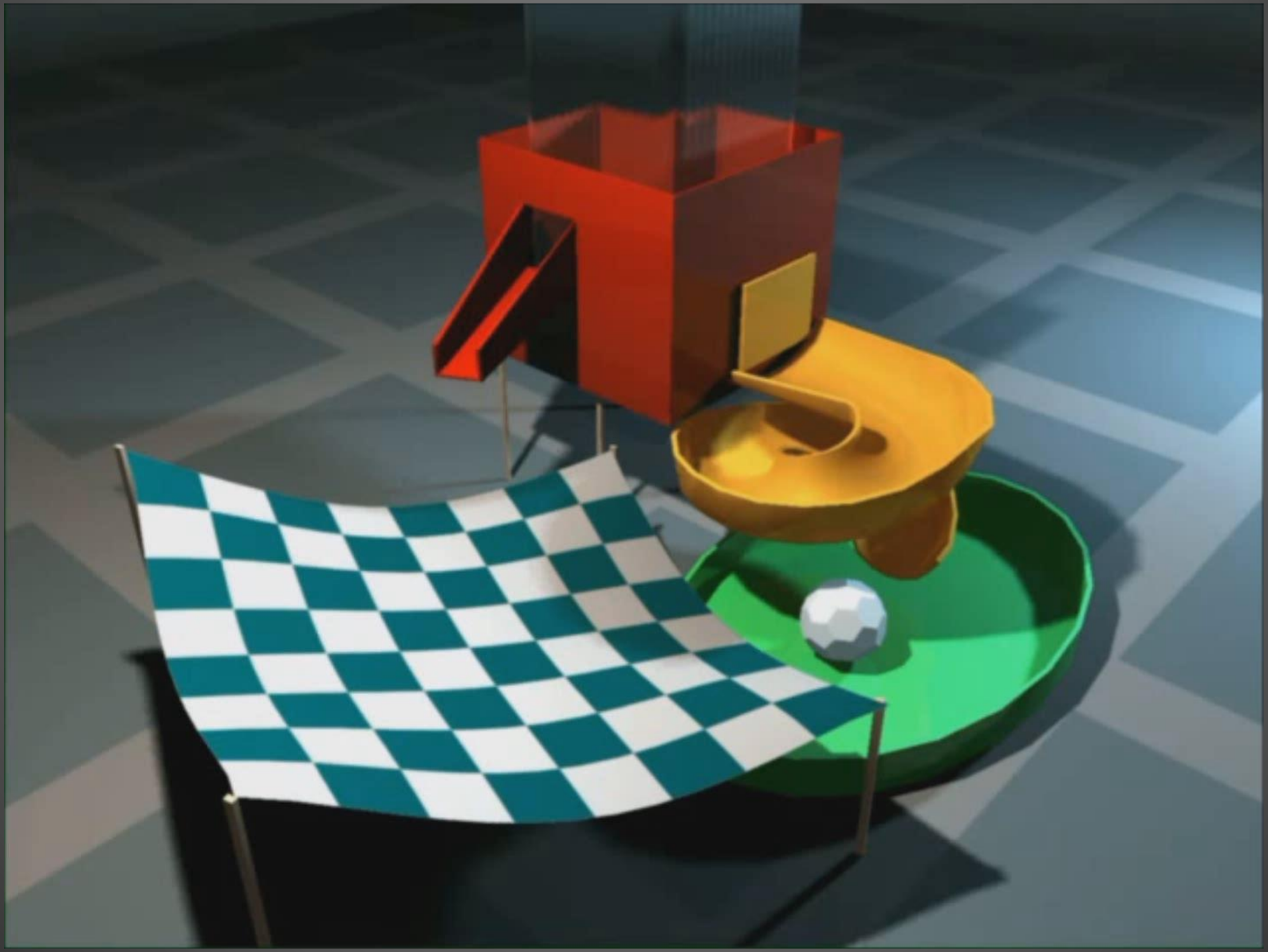
Motivation

Nucleus

nParticles

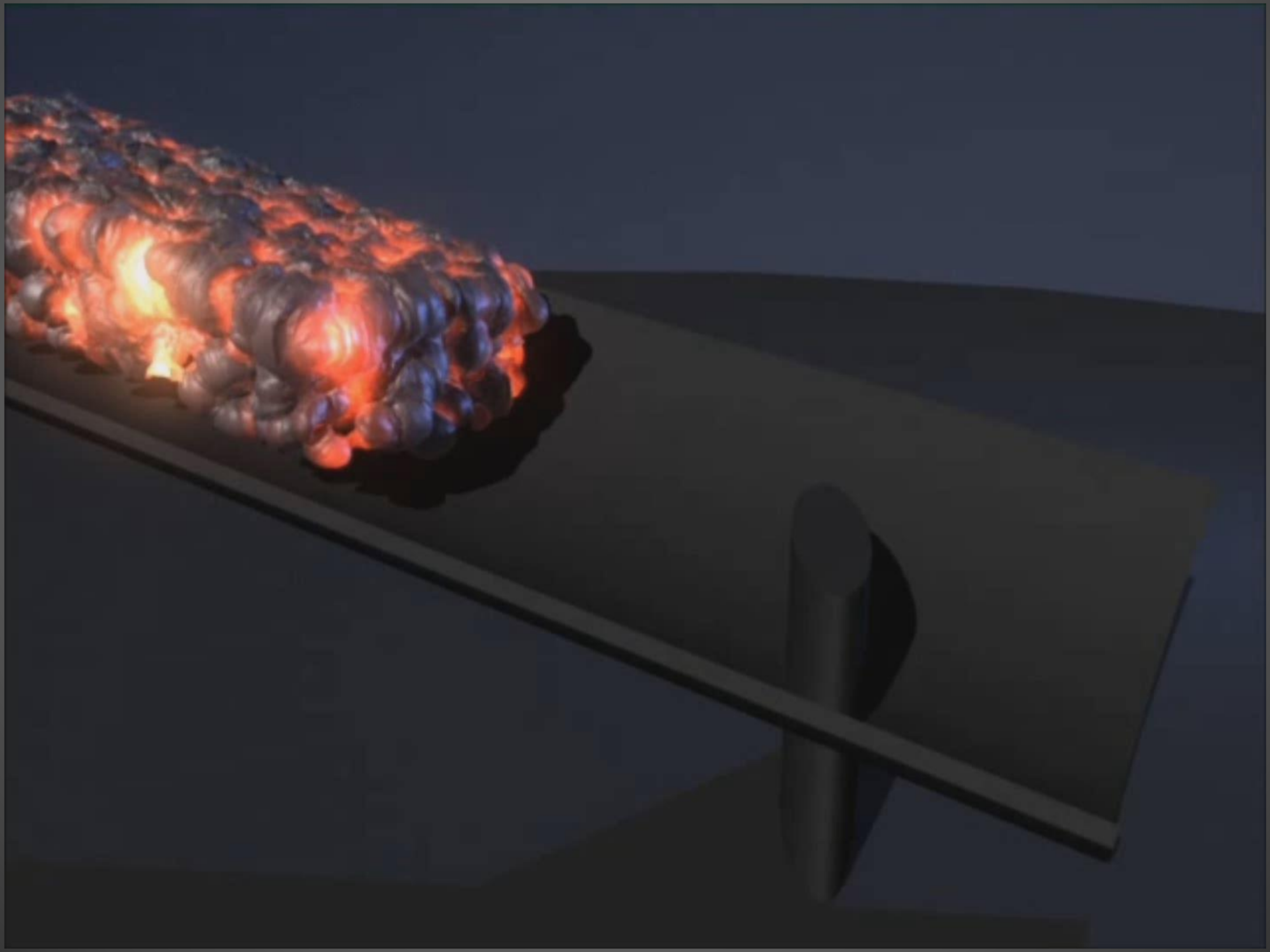
(February 2009)

Motivation



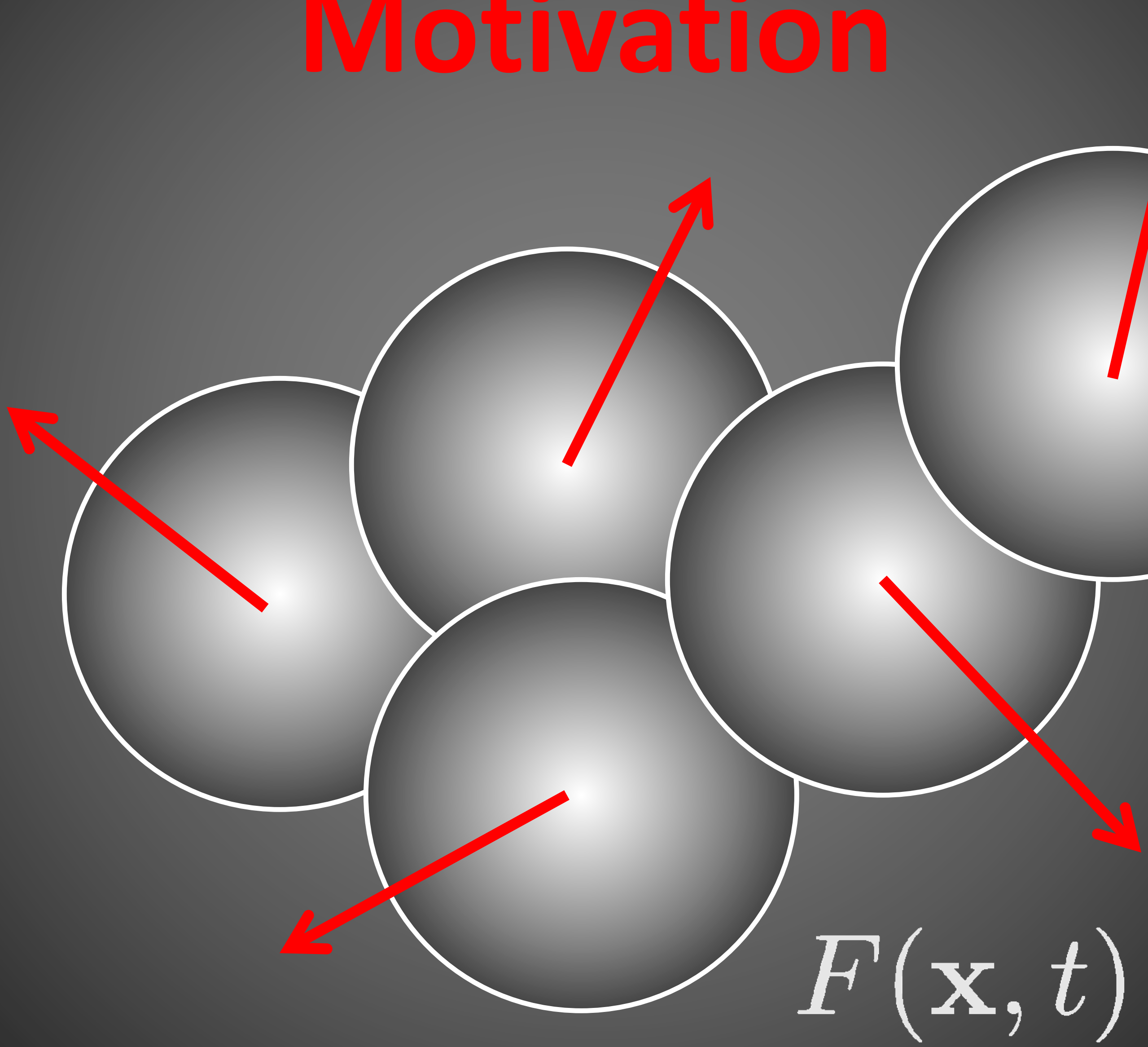
Created by Duncan Brinsmead using MAYA Nucleus.

Motivation

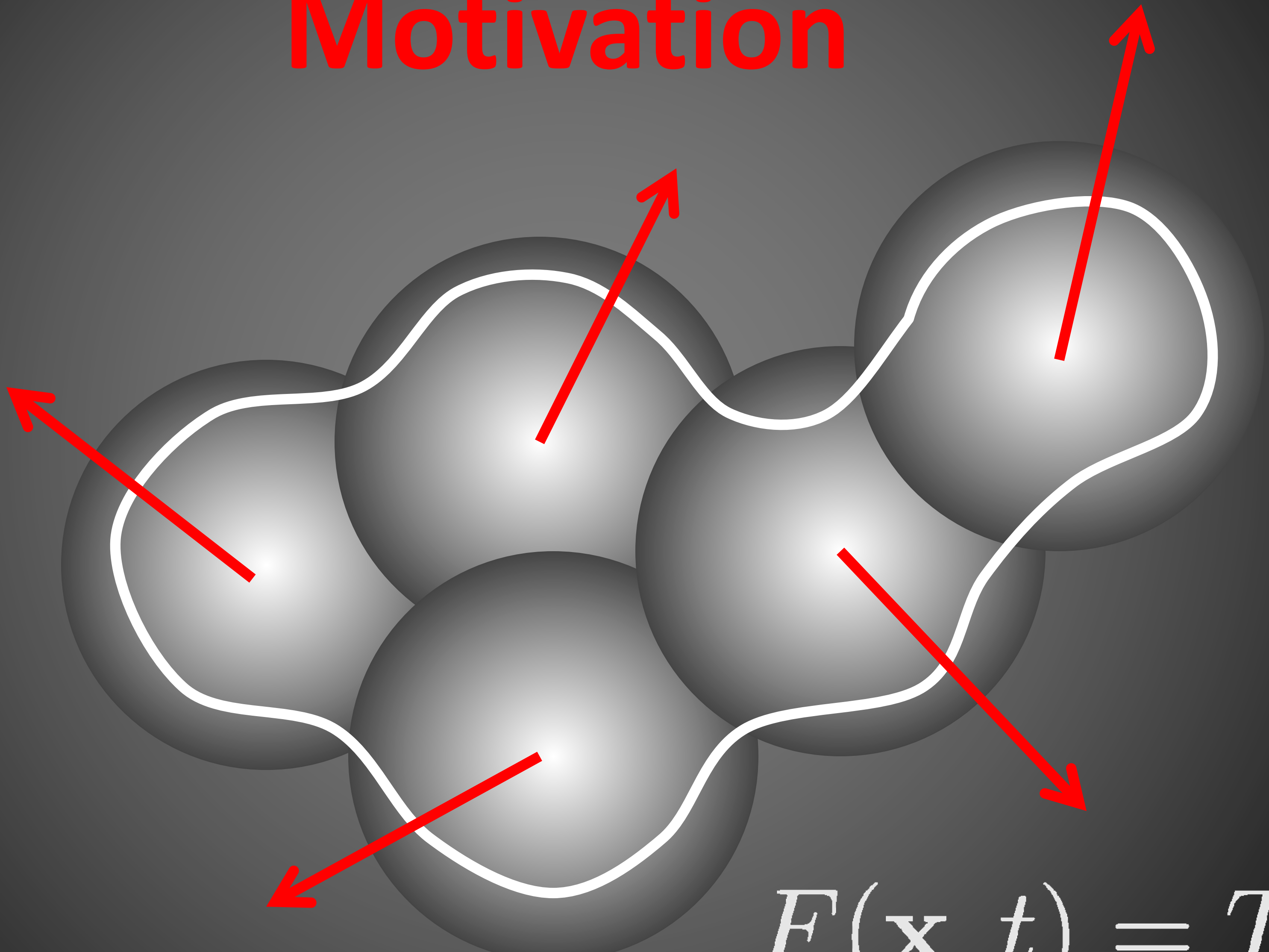


Created by Duncan Brinsmead using MAYA Nucleus.

Motivation

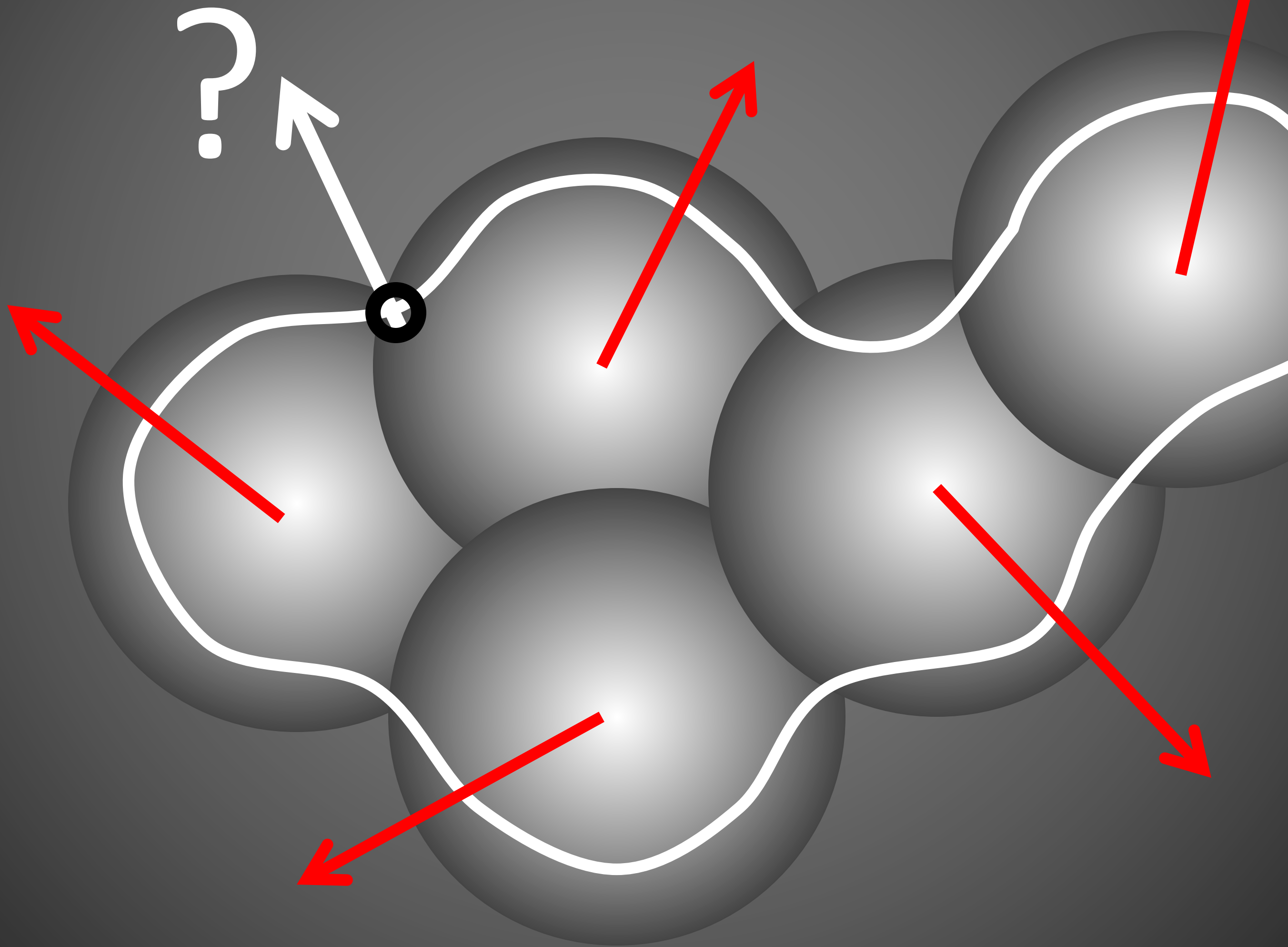


Motivation

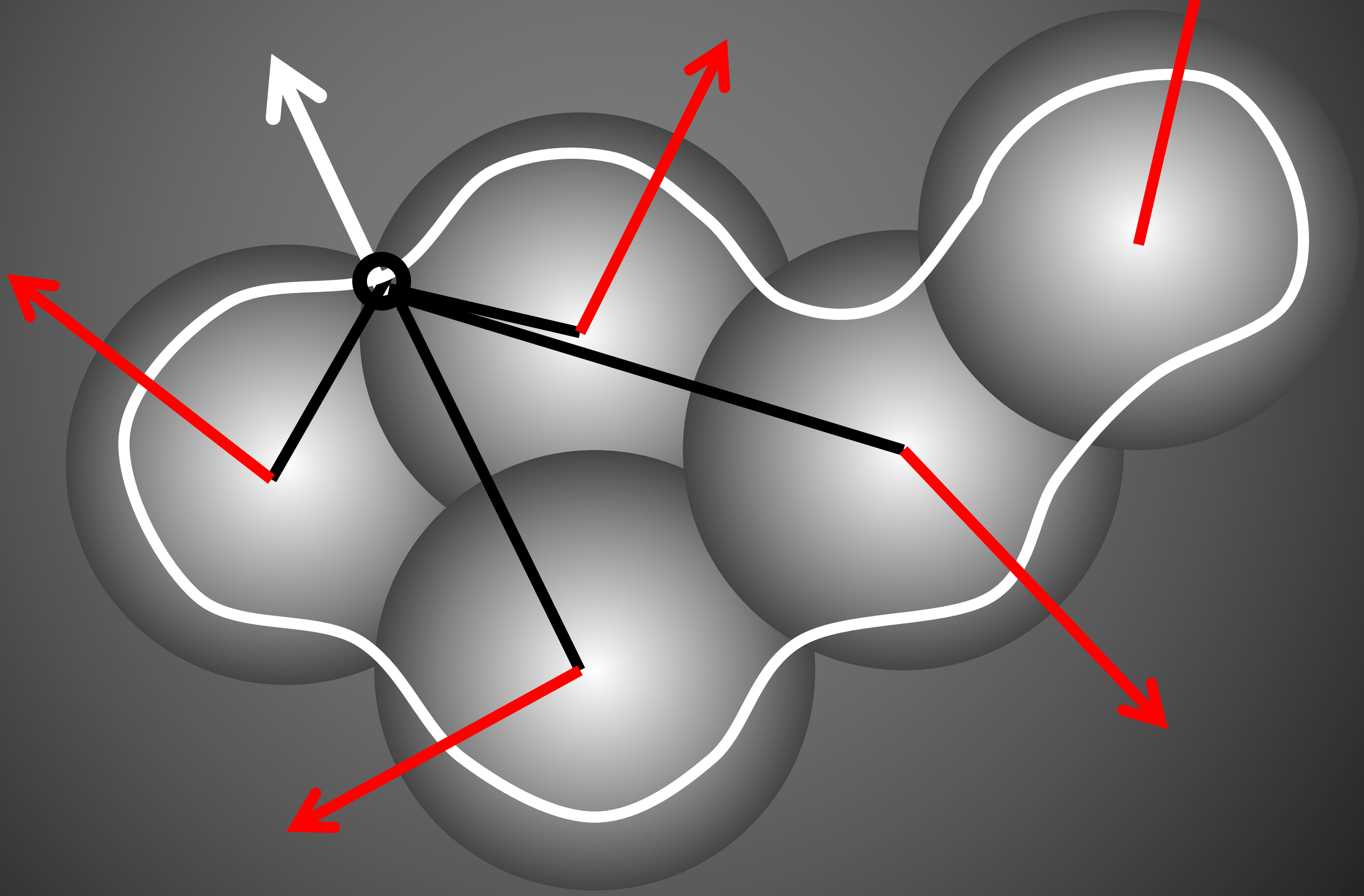


$$F(\mathbf{x}, t) = T$$

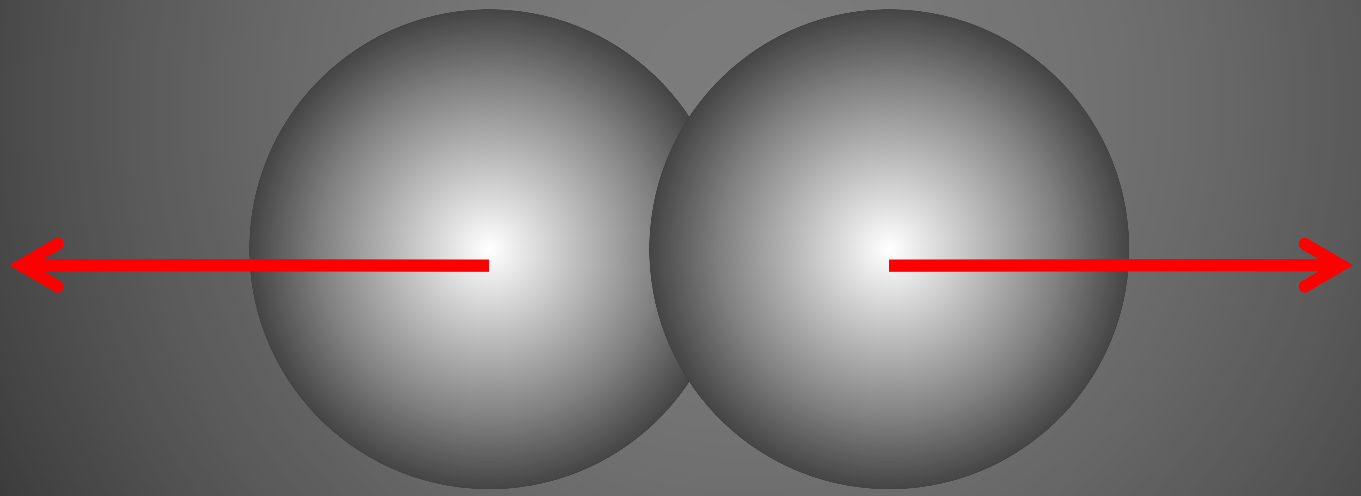
Motivation



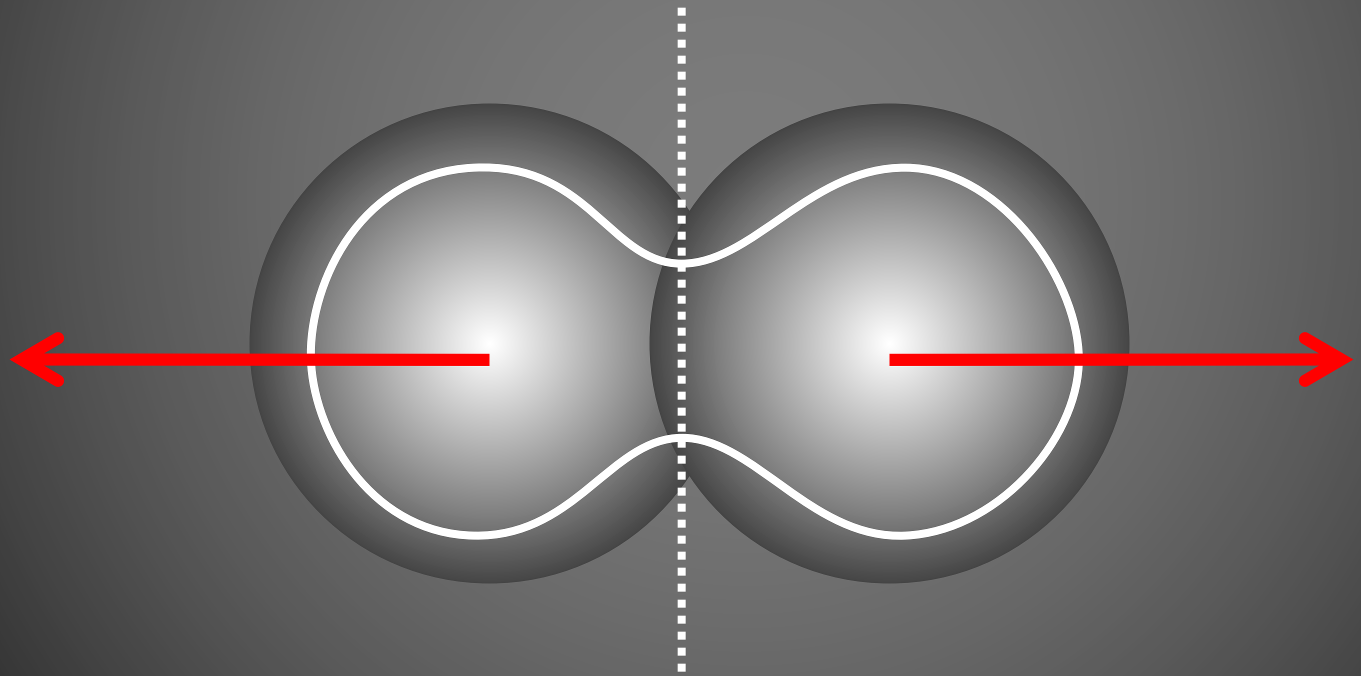
“Obvious Solution”



"Obvious Solution"

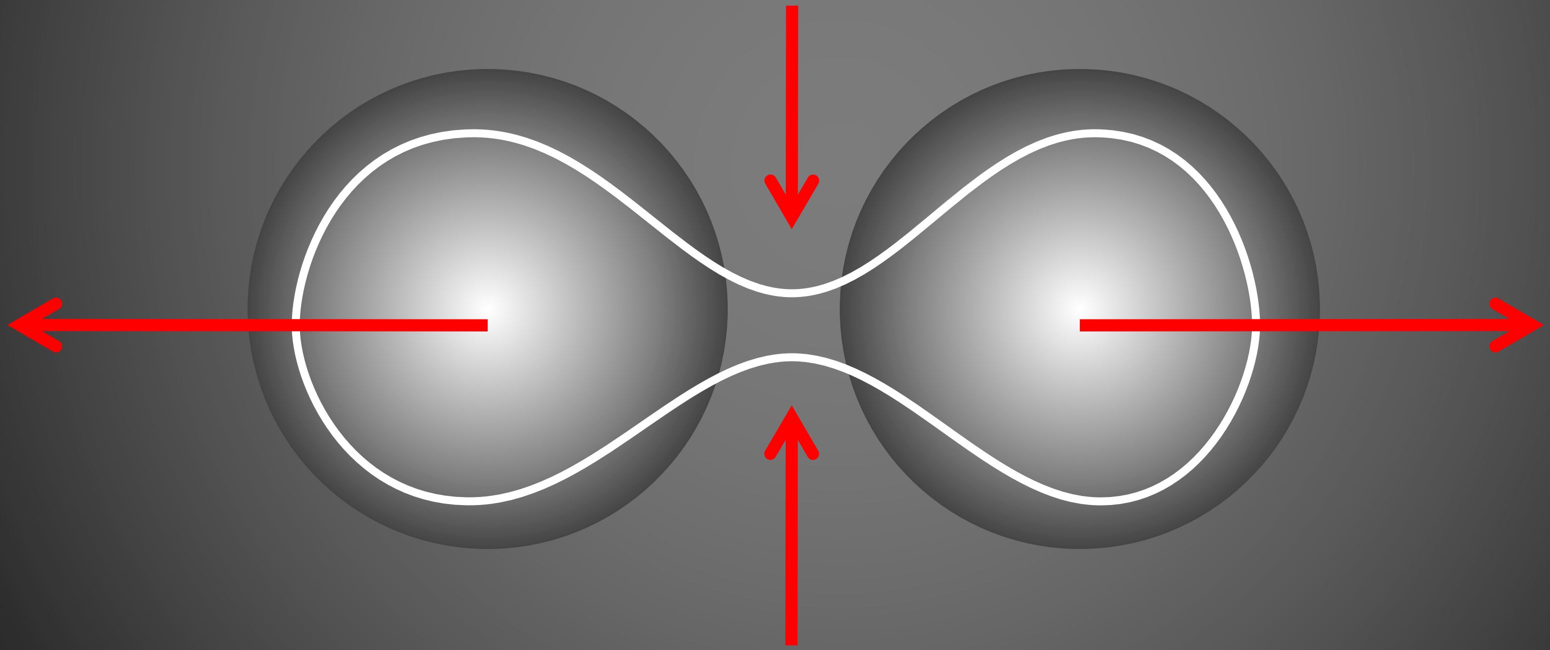


“Obvious Solution”



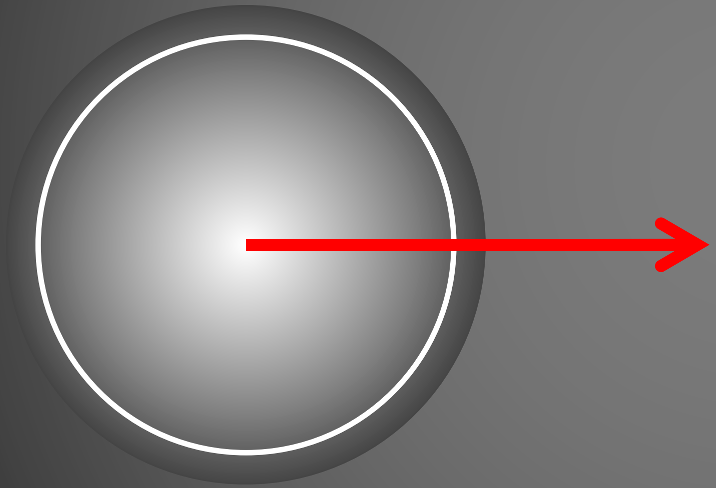
Predicts Zero velocity

“Obvious Solution”

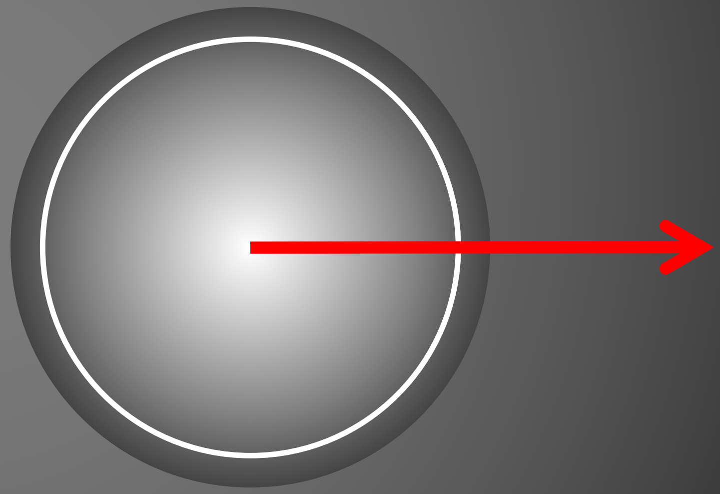


Actually: **NON**-Zero velocity

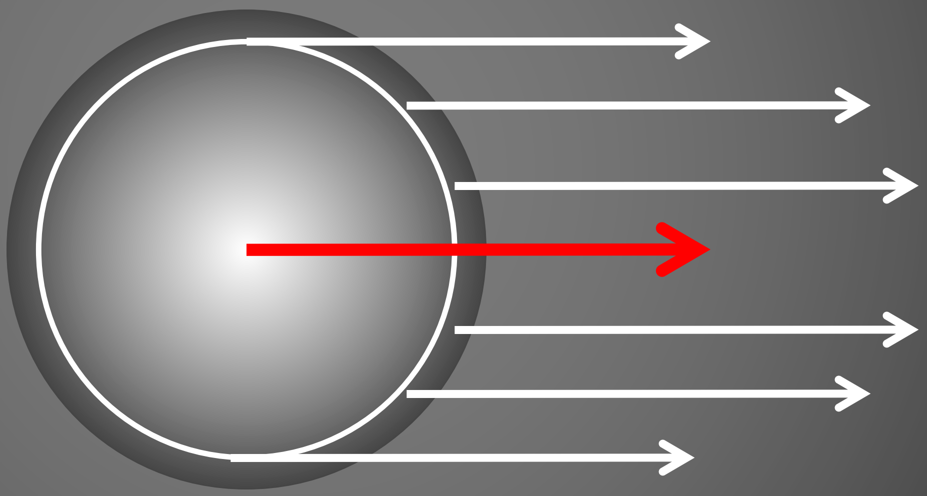
Simple Example



Simple Example



Simple Example



This is what we expect…

Cool Math

Time to formalize this stuff.

Cool Math

$$\Gamma(t) = \{\mathbf{x} | F(\mathbf{x}, t) = 0\}$$

Go beyond blobs:
Time evolving Implicits.

Cool Math

$$\dot{F}(\mathbf{x}, t) = \dot{0} = 0$$

Cool Math

$$\frac{\partial F}{\partial t} + \nabla F \cdot \dot{\mathbf{x}} = 0$$

Cool Math

$$\nabla F \cdot \dot{\mathbf{x}} = -\frac{\partial F}{\partial t}$$

Cool Math

$$\dot{\mathbf{x}} = -(\nabla F)^{\dagger} \frac{\partial F}{\partial t}$$

Cool Math

$$A^{\dagger} = \left(A^{T}A\right)^{-1}A^{T}$$

Moore-Penrose Pseudo-Inverse

Cool Math

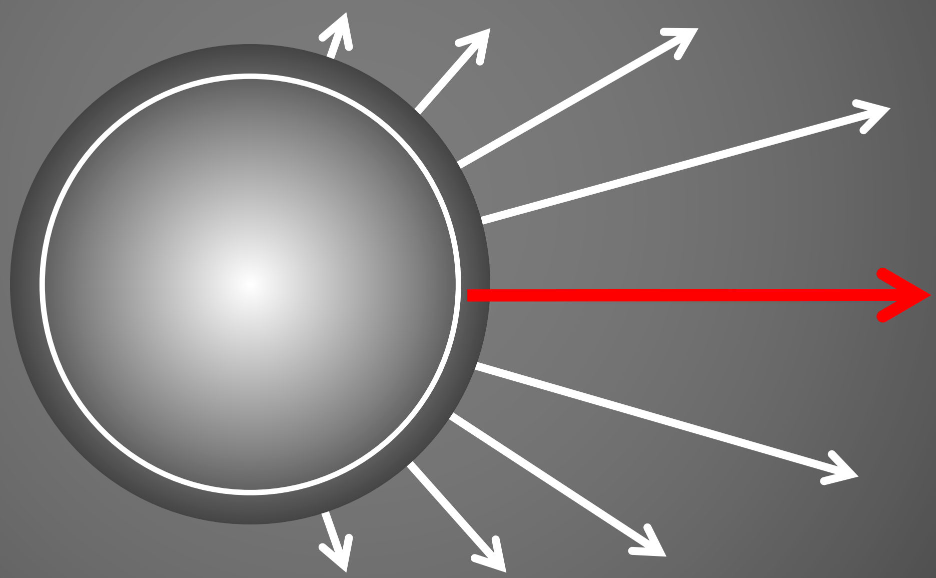
$$\dot{\mathbf{x}} = -\frac{\nabla F}{|\nabla F|^2}\frac{\partial F}{\partial t}$$

Cool Math

$$\dot{\mathbf{x}} = a_F \, \mathbf{n}$$

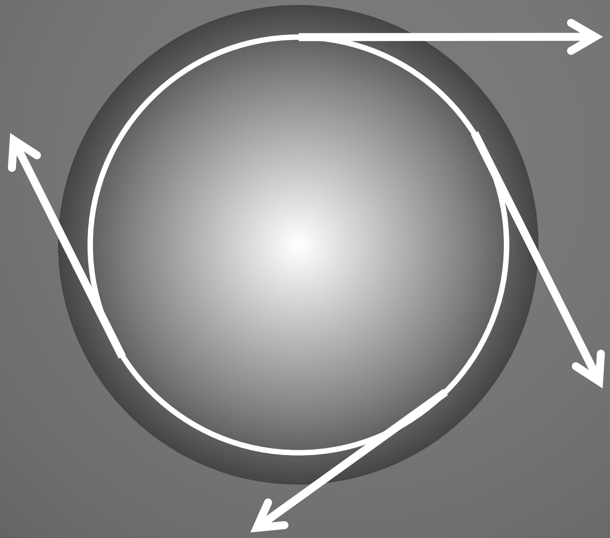
Fixes normal velocity

Back to Simple Example



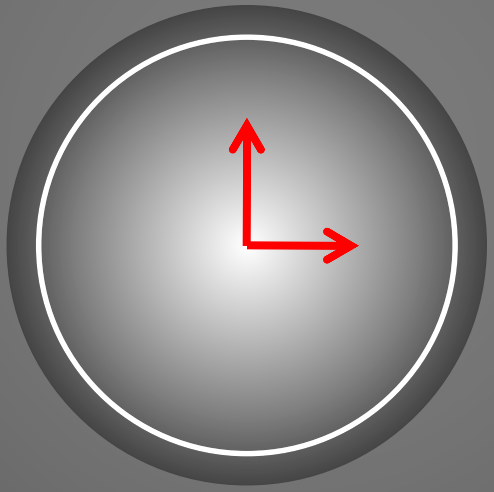
Normal velocity

Simple Example

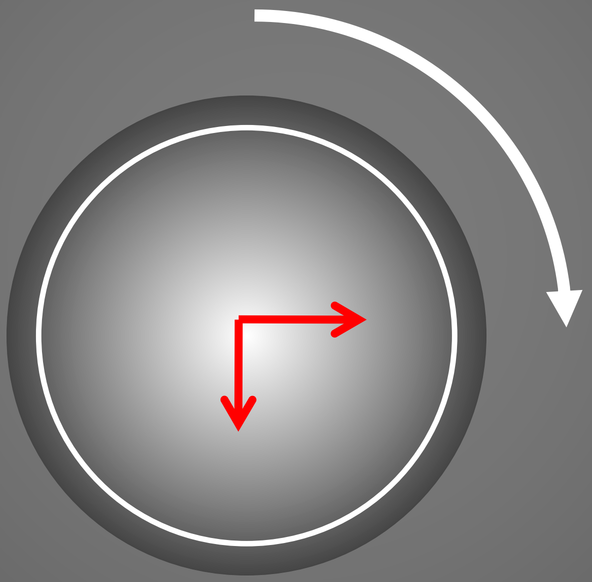


Tangential velocity does not change the shape.

Simple Example



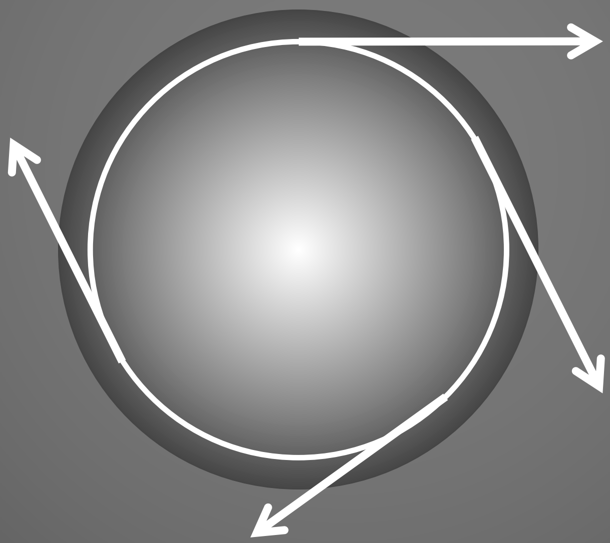
Simple Example



Tangential velocity **irrelevant**

In Theory

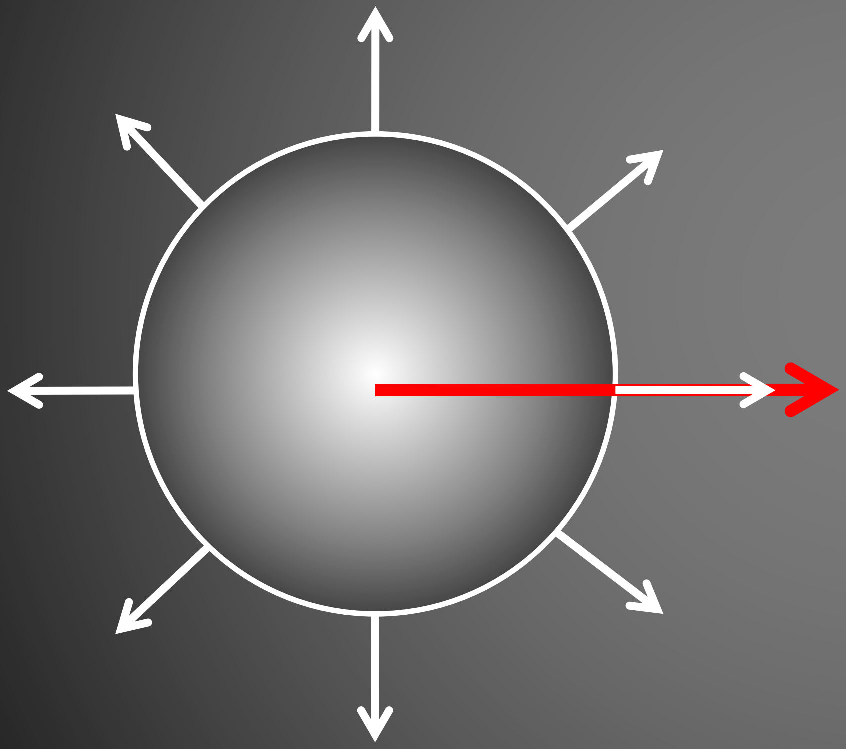
Key Insight



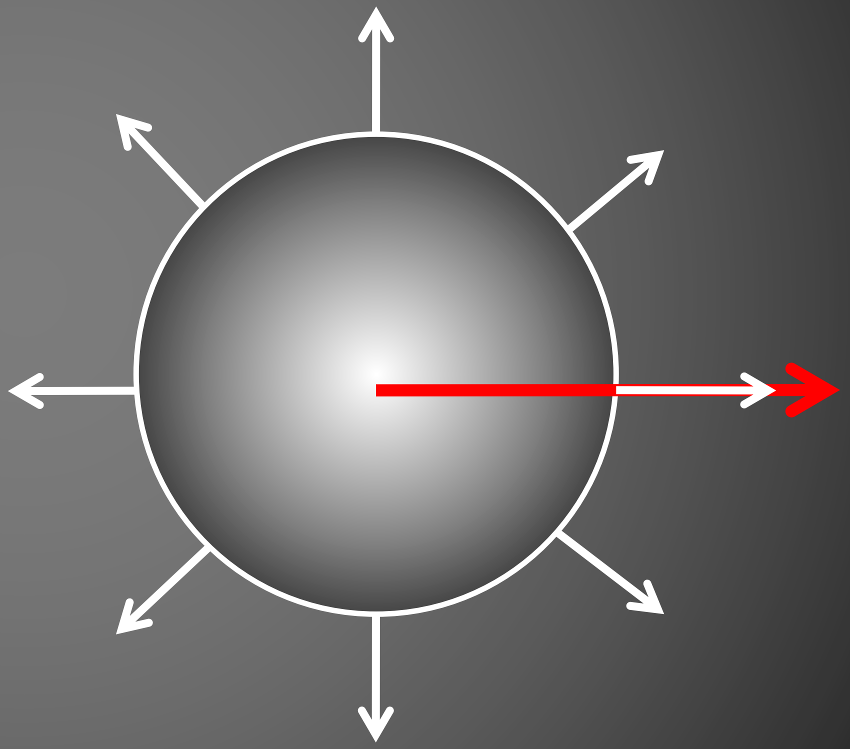
Tangential velocity is **key**

In Practice

Key Idea



Key Idea



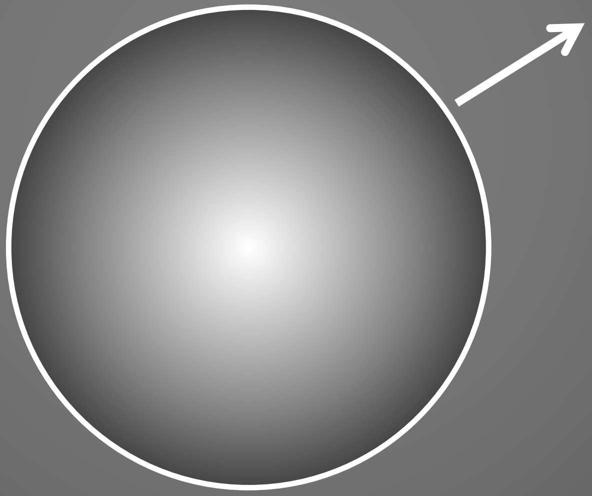
Normals do not change

Key Idea

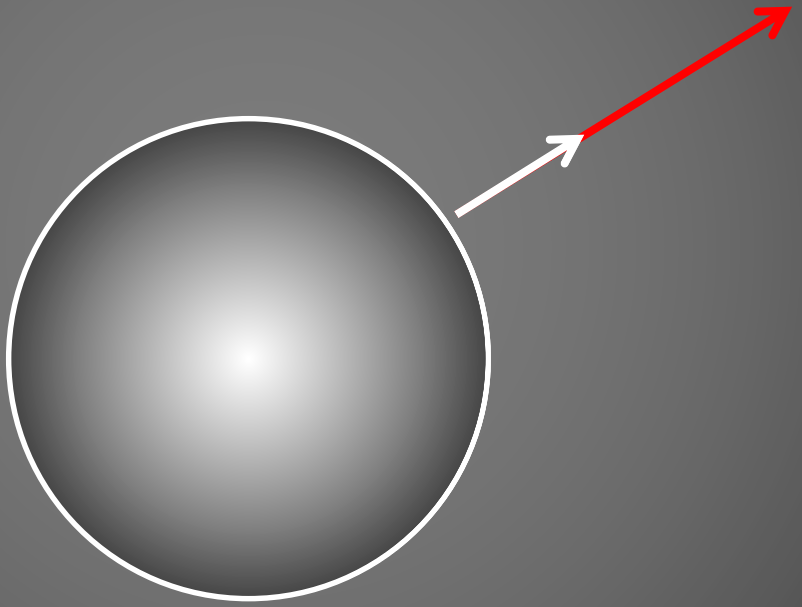
Add the constraint:

$$\frac{d}{dt}\left(\mathbf{n}(\mathbf{x}, t)\right) = 0$$

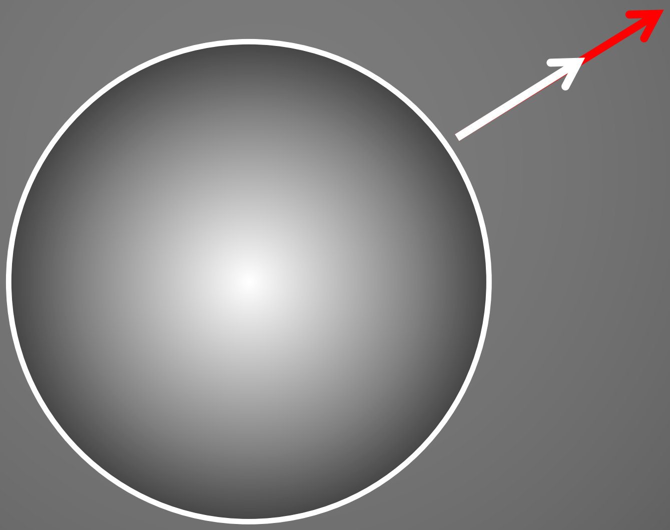
Why is this a sensible Idea?



Why is this a sensible Idea?

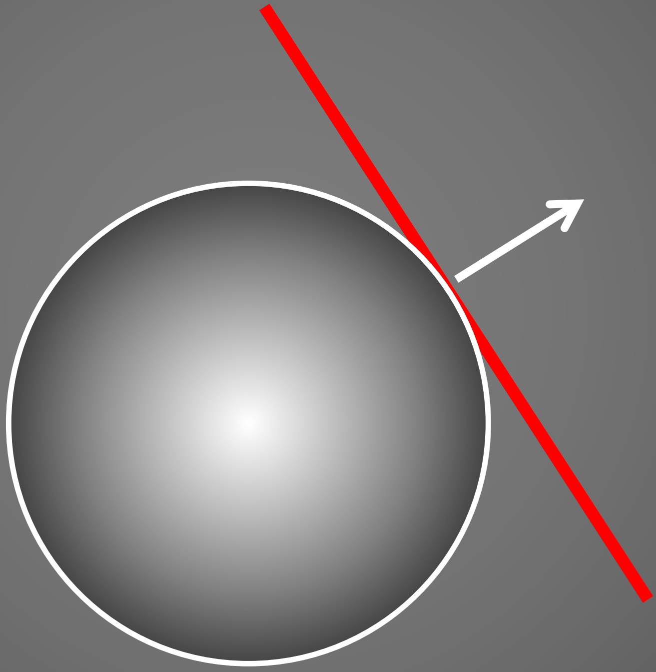


Why is this a sensible Idea?



Same normal.

Why is this a sensible Idea?



Fixes Tangential Component

Cool Math

$$\frac{d}{dt}\left(\mathbf{n}(\mathbf{x}, t)\right) = 0$$

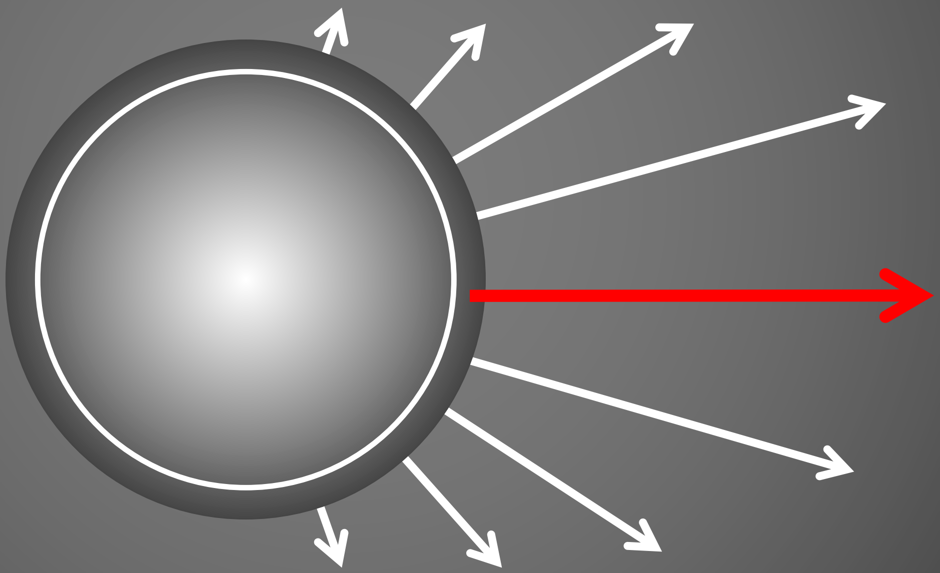
Normals do not change.

Cool Math

$$\mathbf{P}_{\mathbf{n}}\mathbf{H}_F\,\dot{\mathbf{x}} = -\mathbf{P}_{\mathbf{n}}\,\frac{\partial}{\partial t}\nabla F,$$

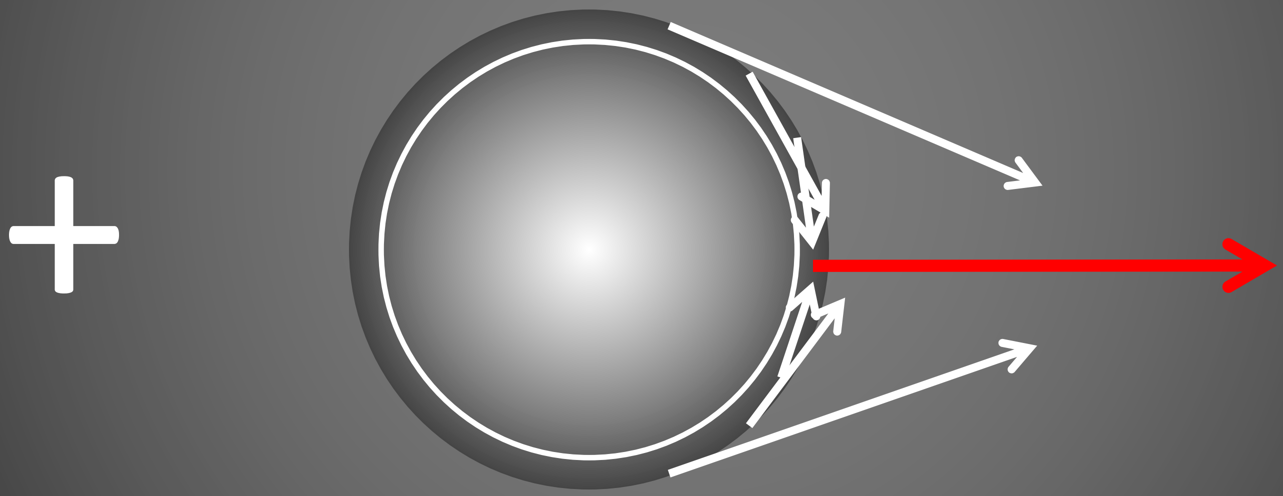
Formula for tangential velocity

Simple Example



Normal velocity

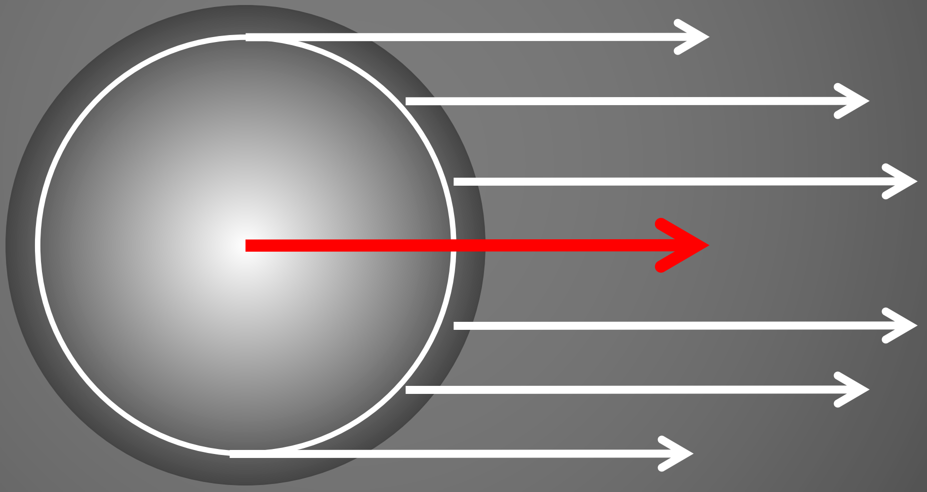
Simple Example



Tangential velocity

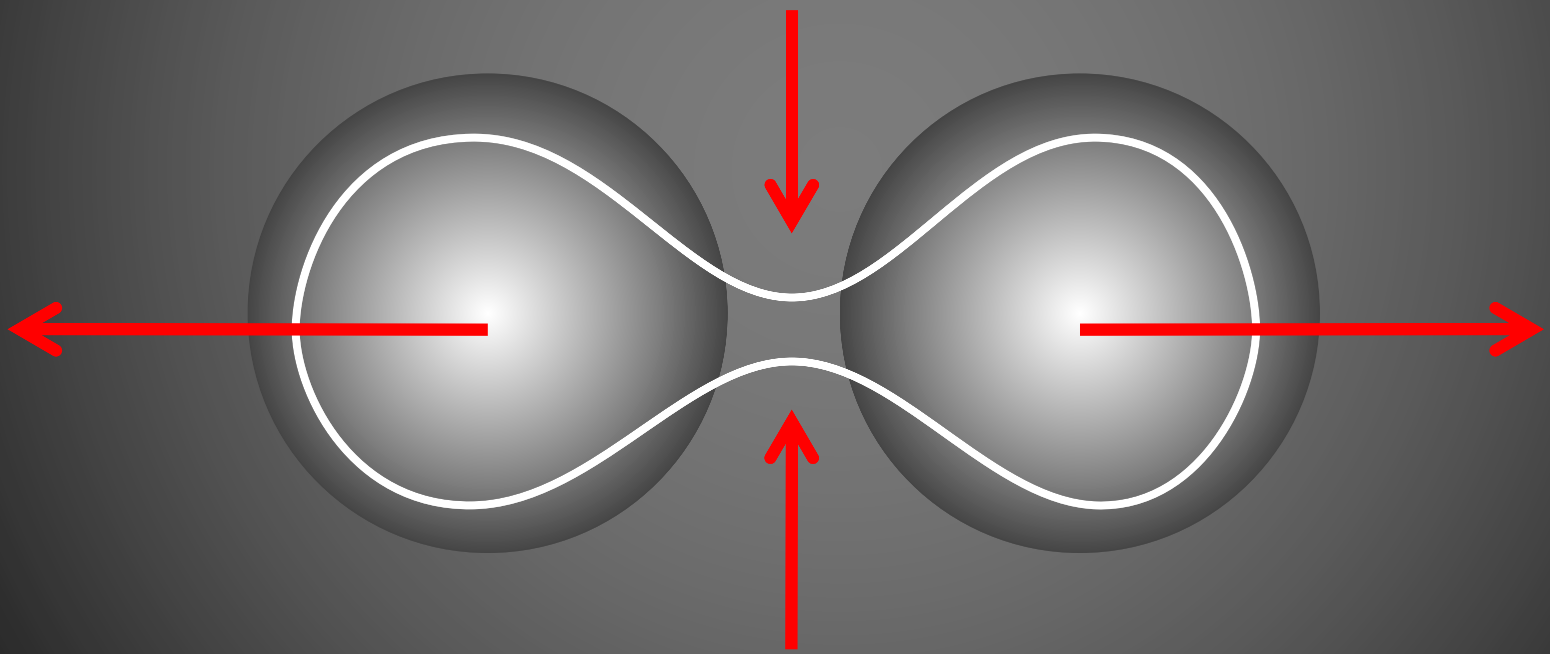
Simple Example

=



Total velocity

Back to “Obvious Solution”



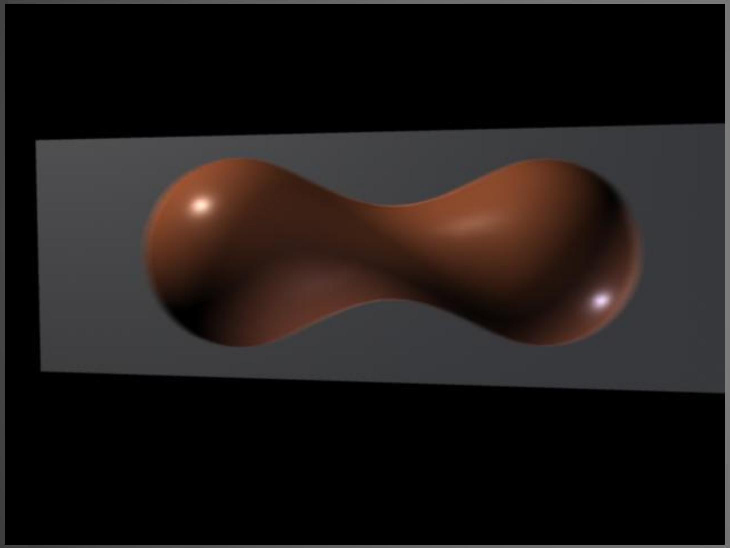
Add “obvious” tangential velocity to normal

Applications

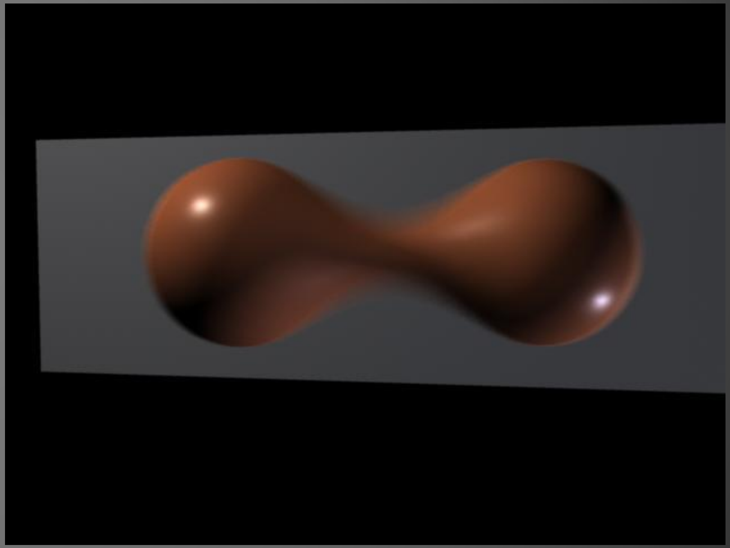
Motion Blur

Particle Surface Tracking

Motion Blur

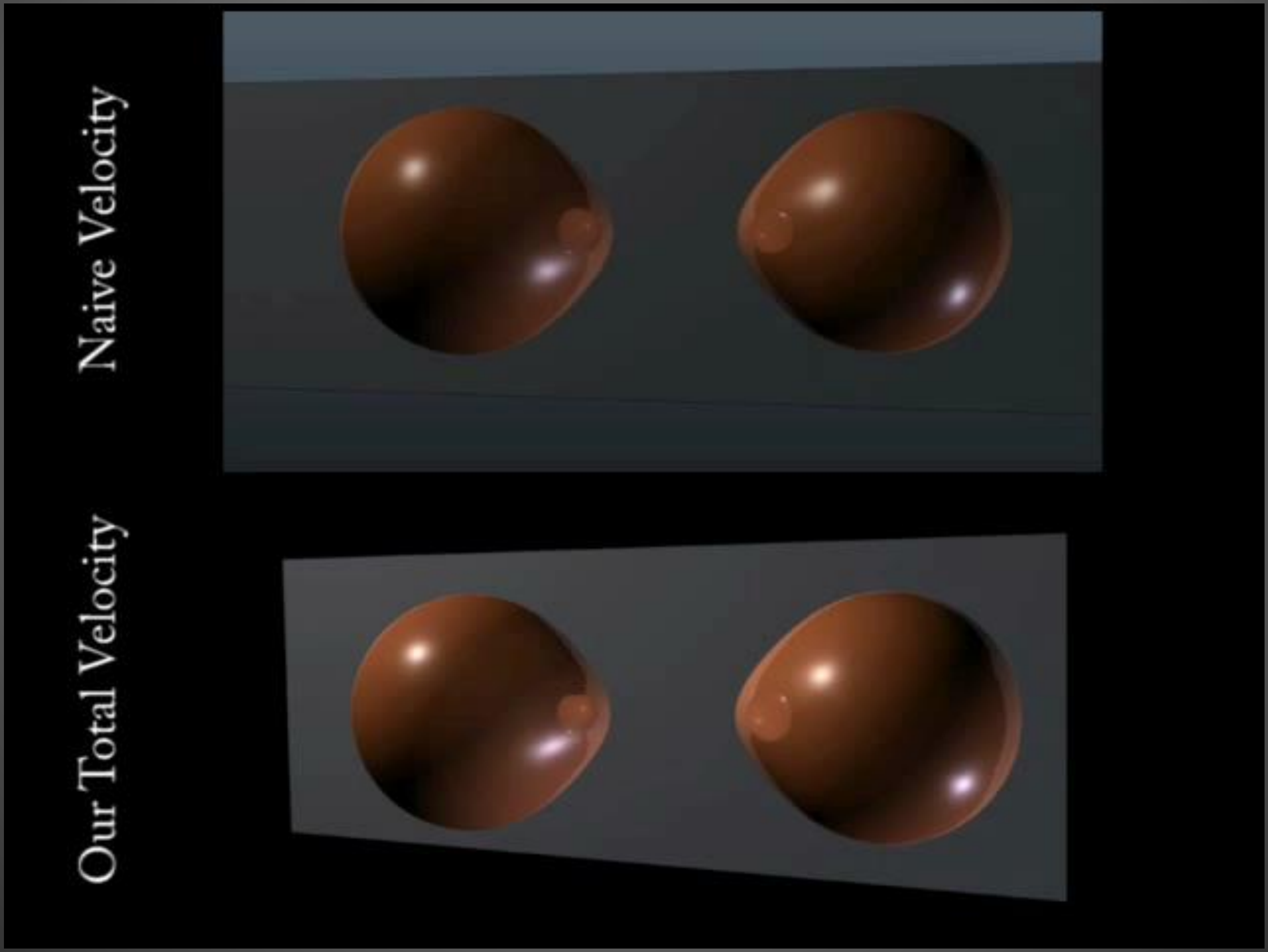


Naive Velocity

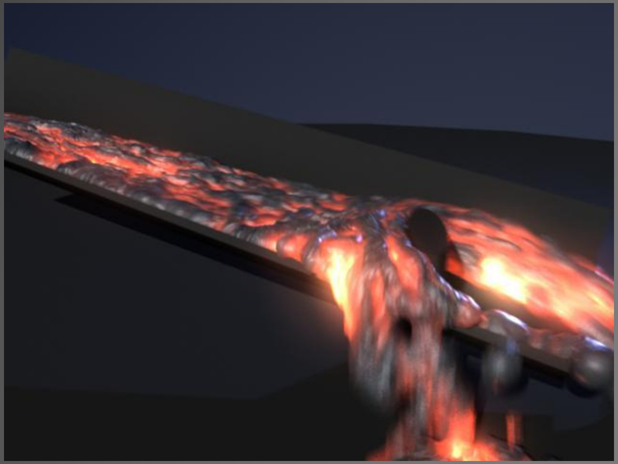


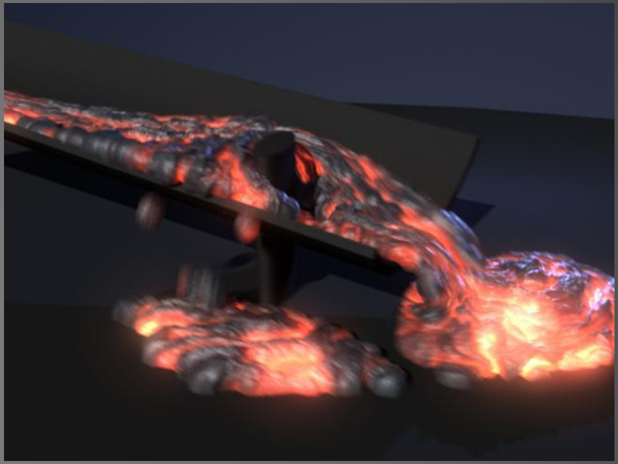
Total Velocity

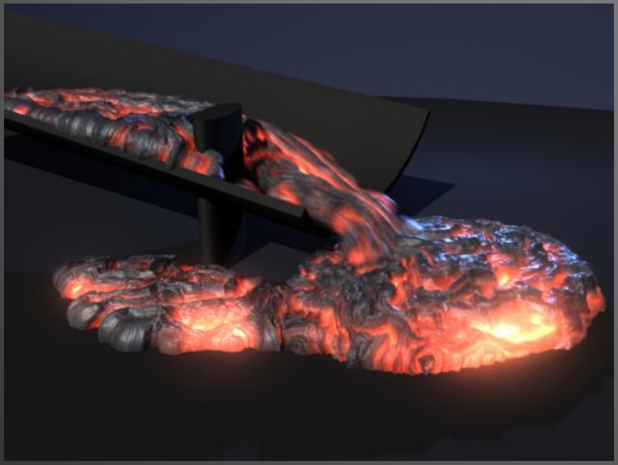
Motion Blur

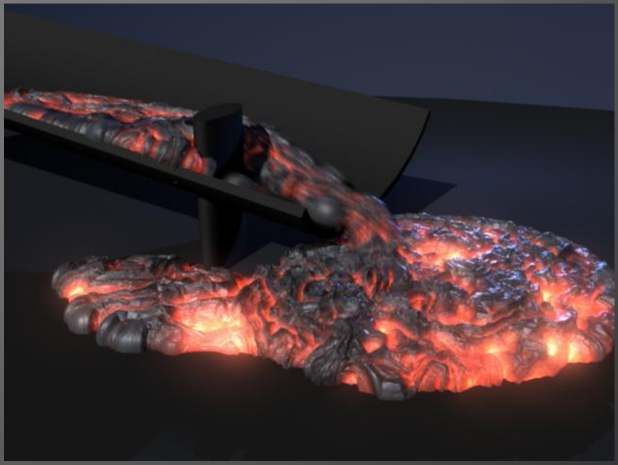


Motion Blur

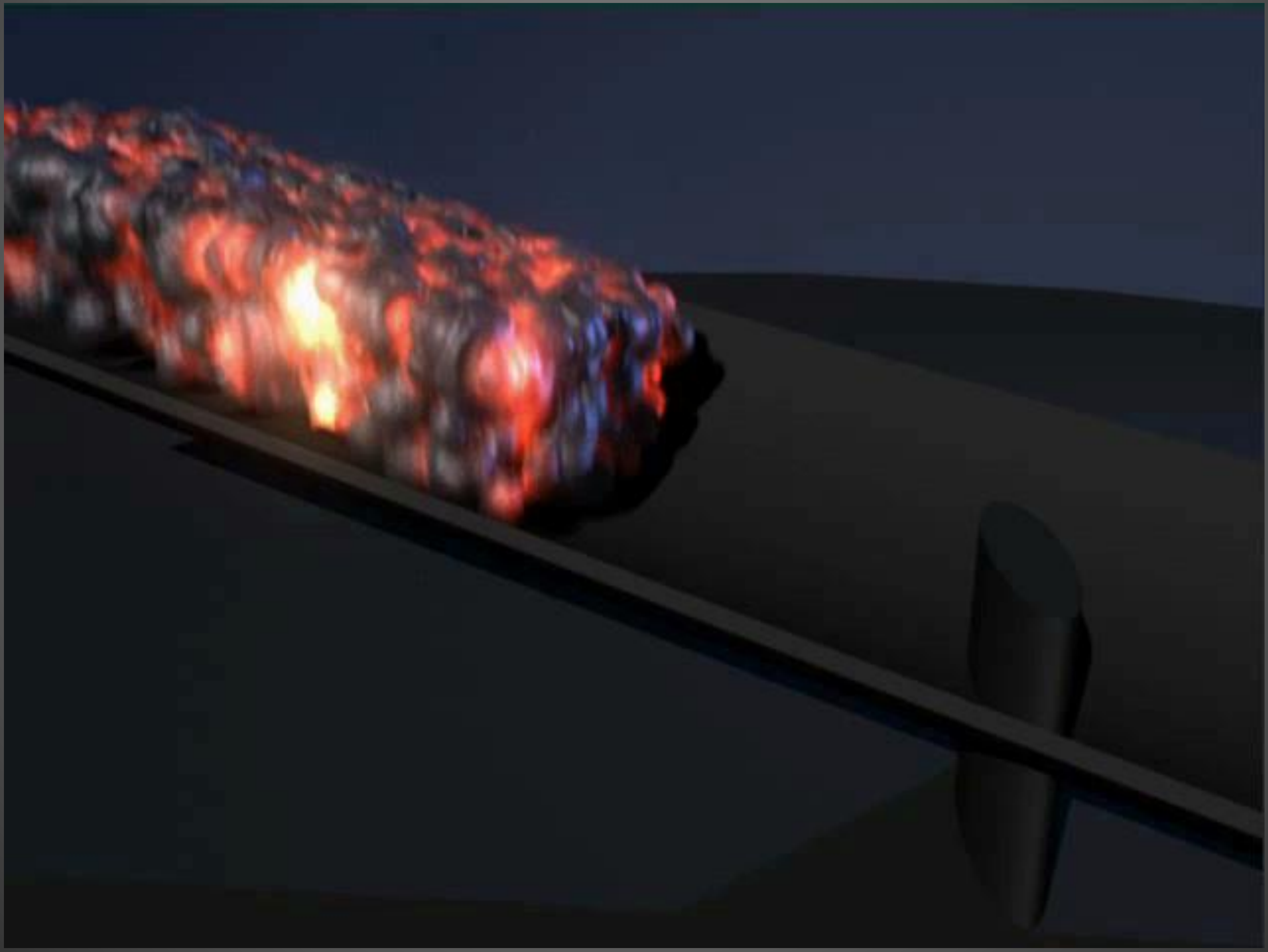






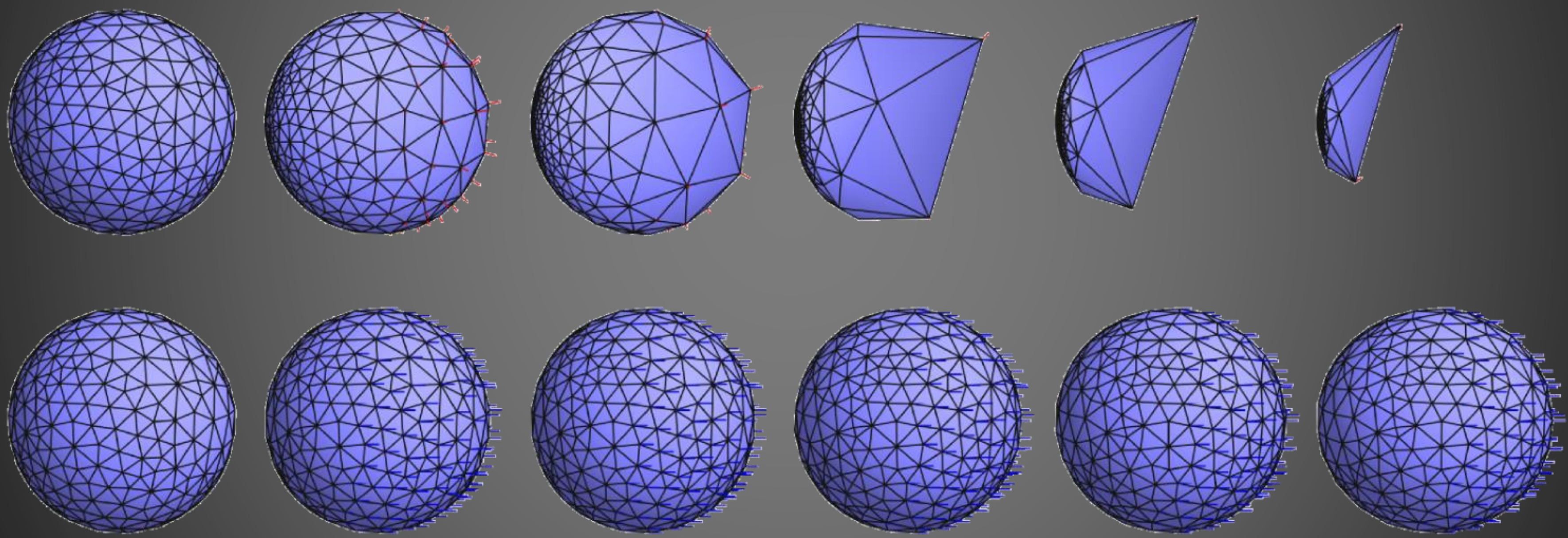


Motion Blur

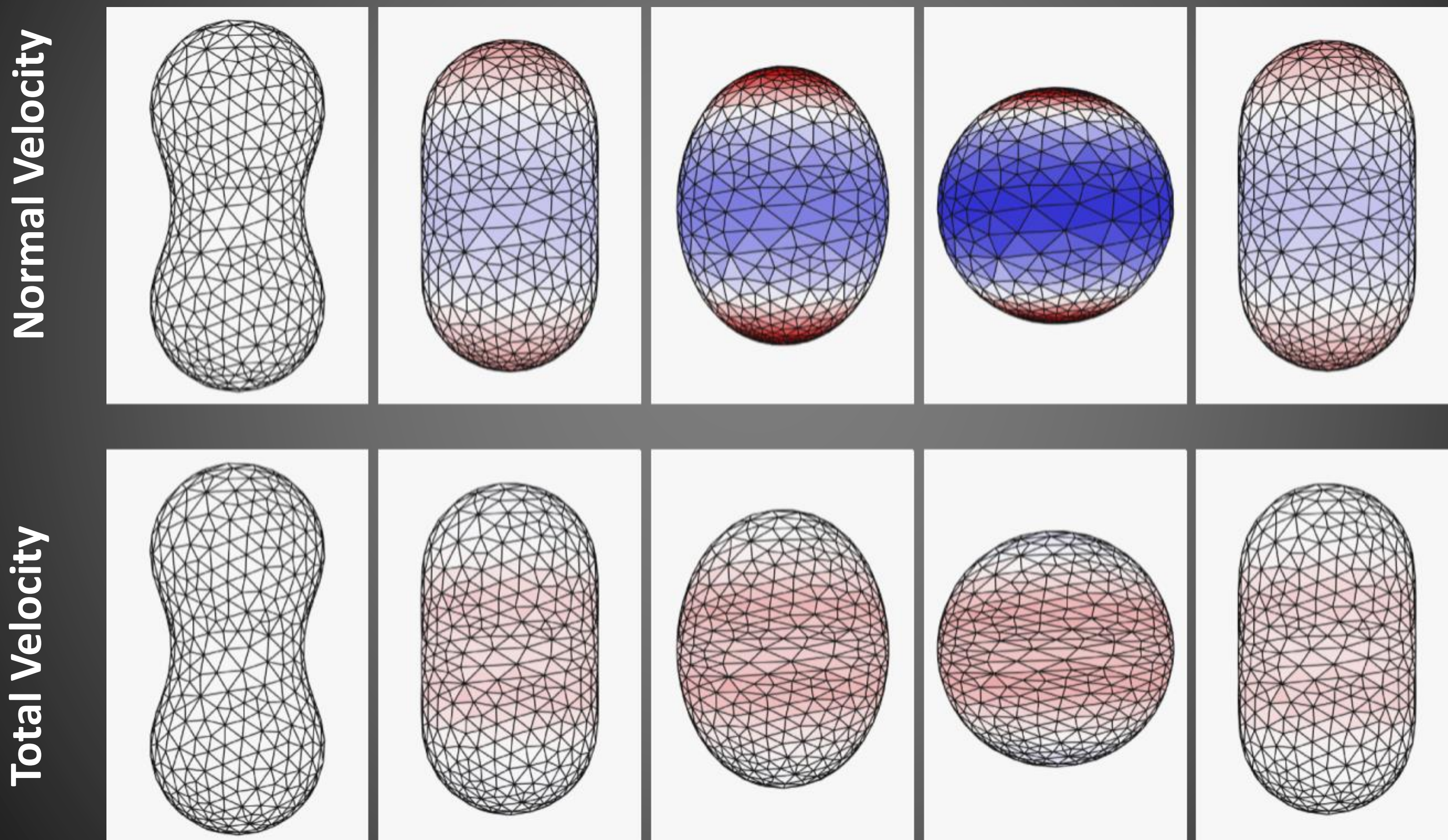


Particle Surface Tracking

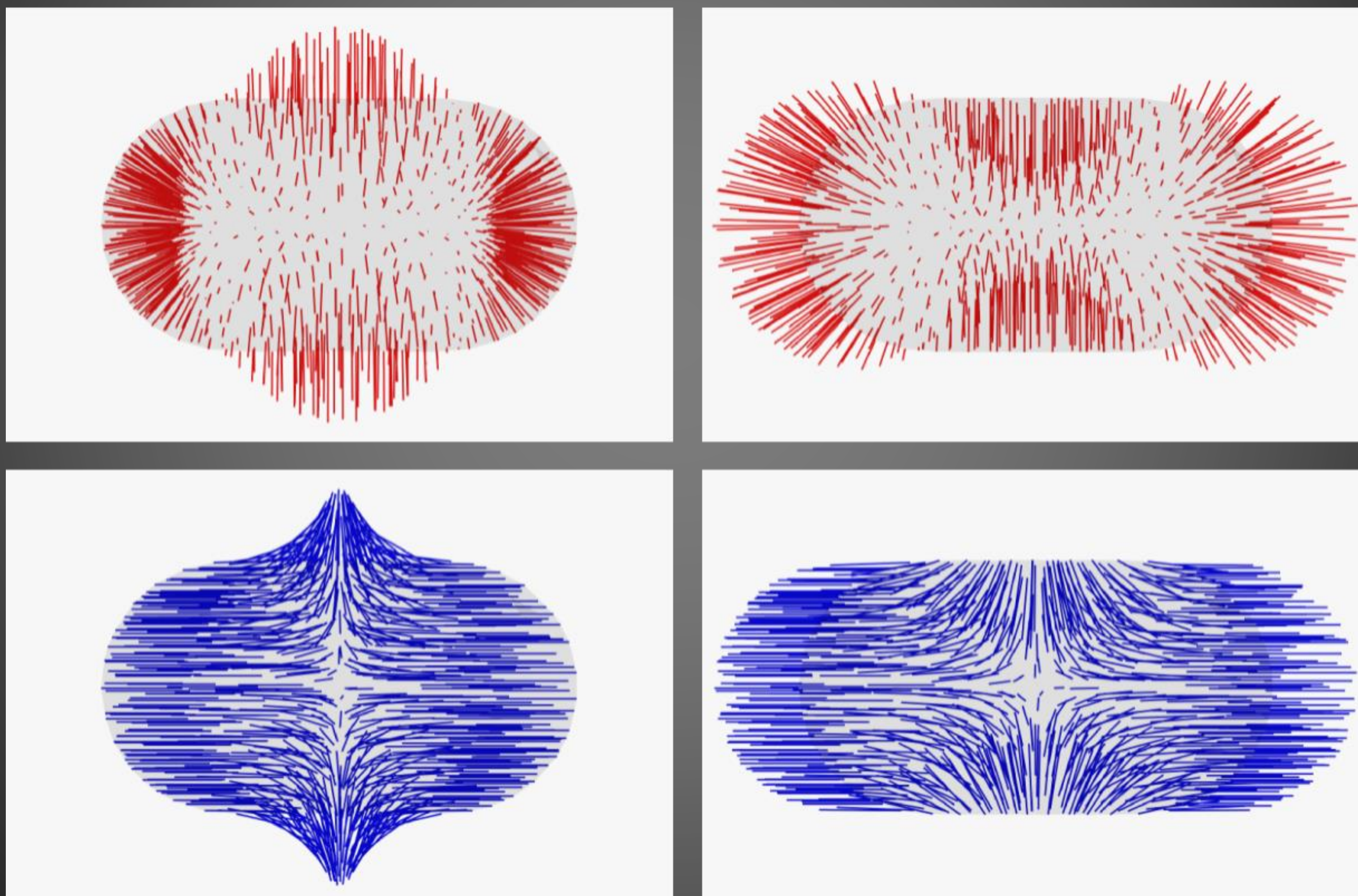
Translation



Deformation



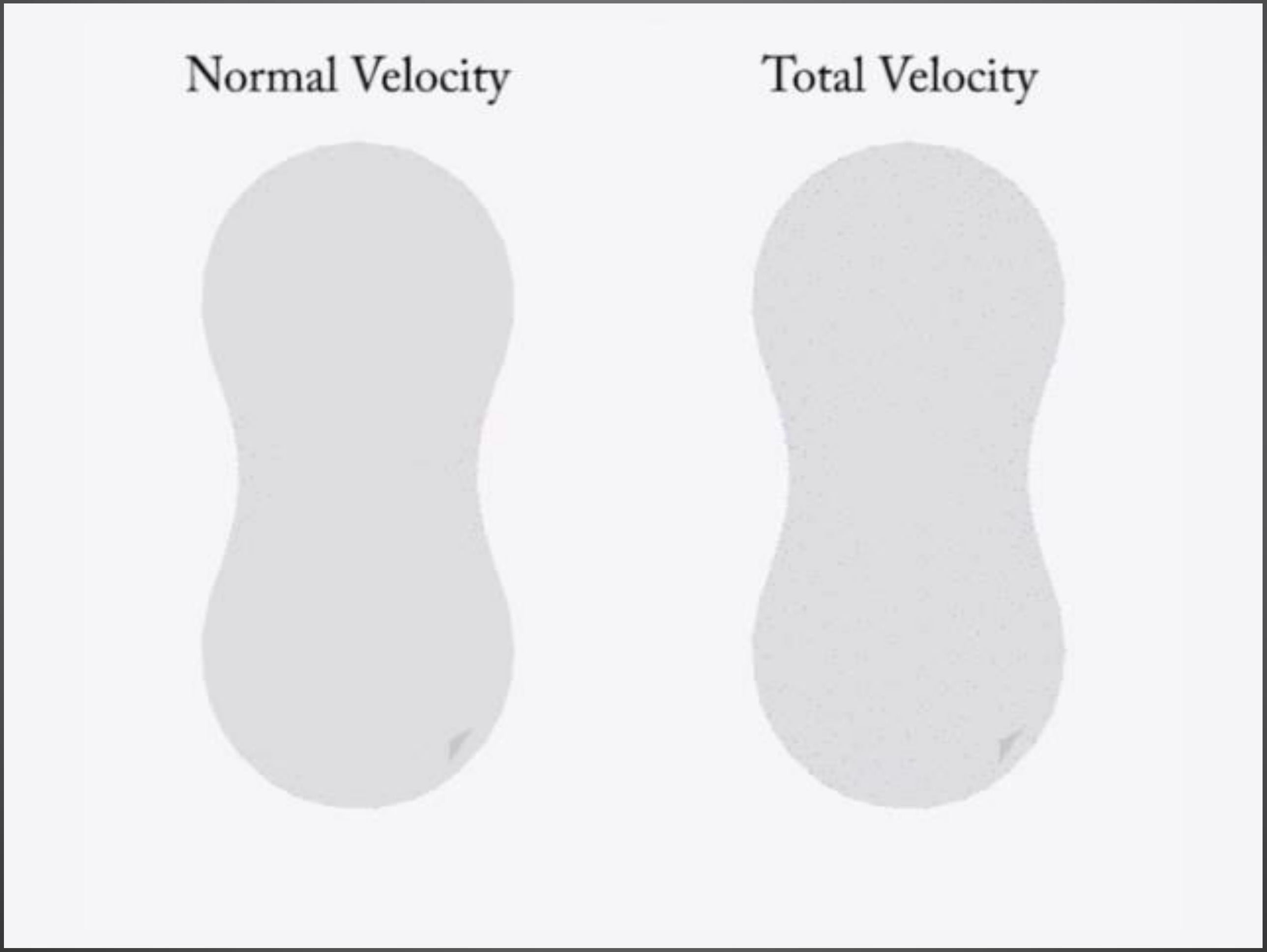
Deformation



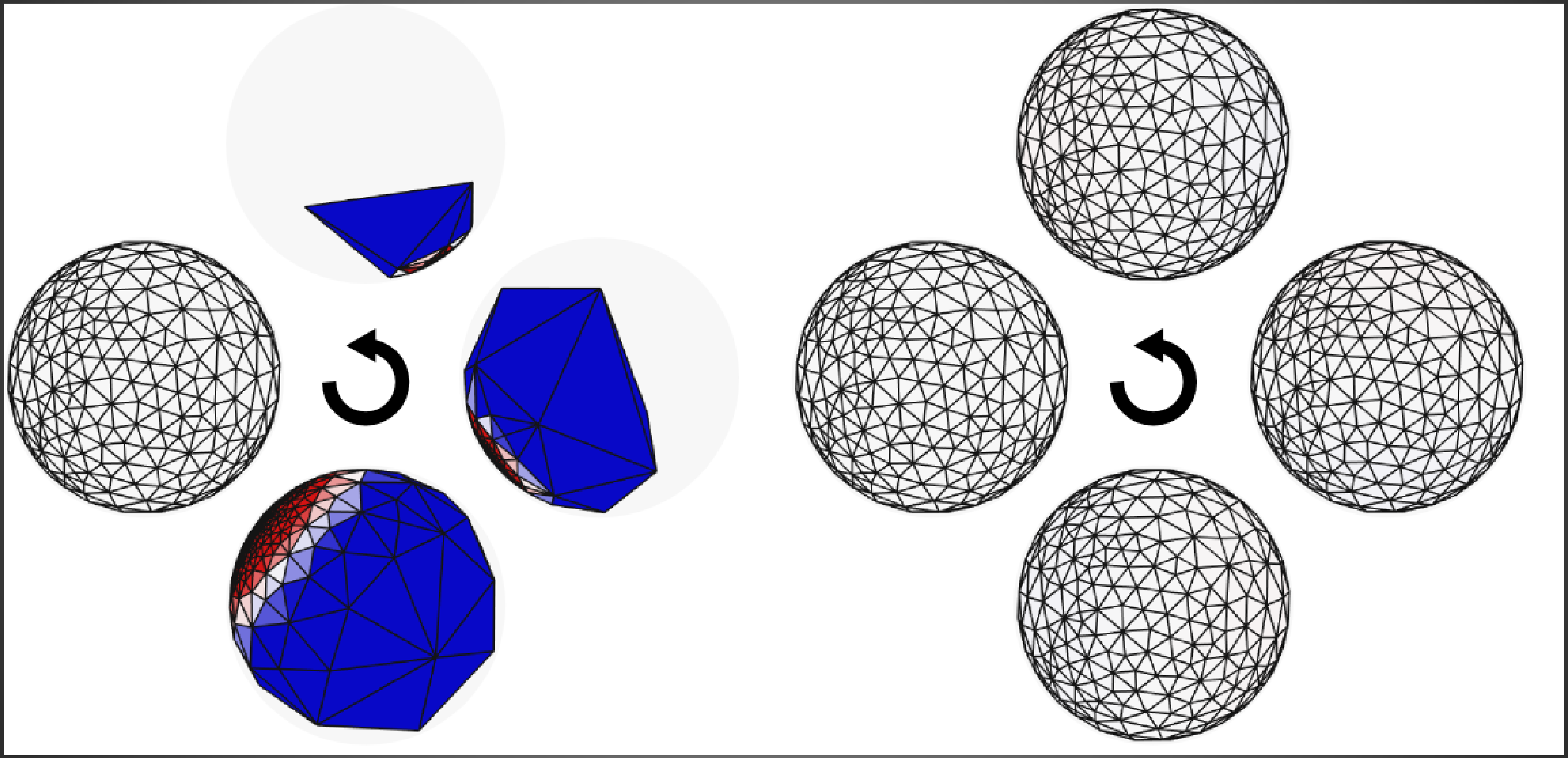
Normal Velocity

Total Velocity

Deformation



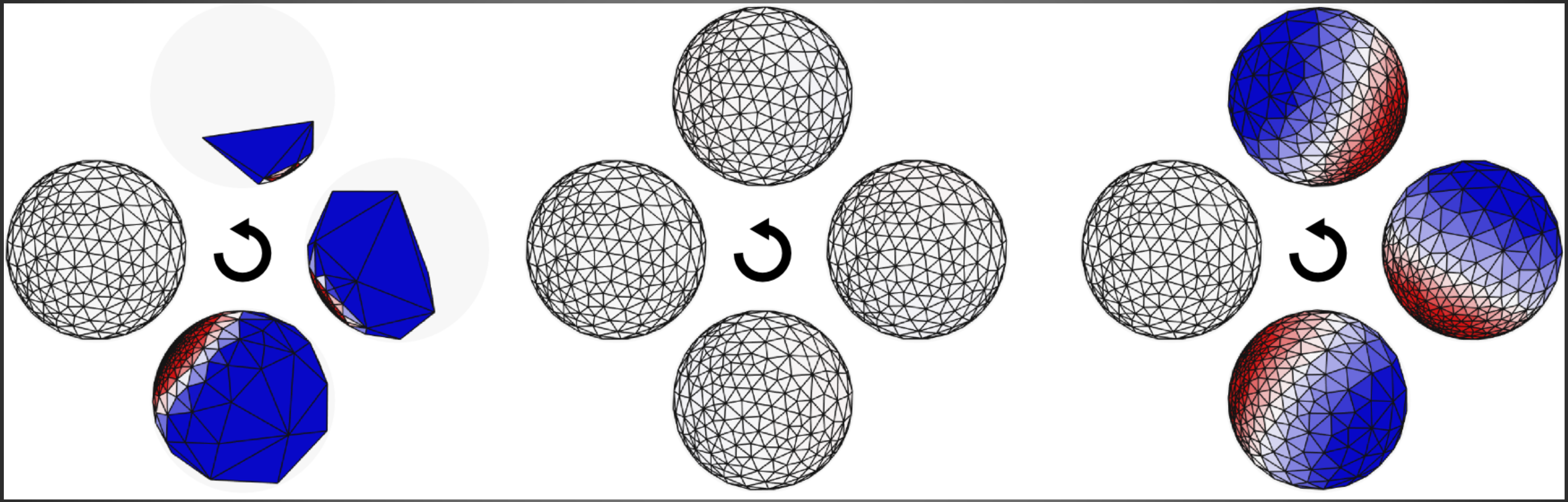
Rotation



Normal Velocity

Total Velocity

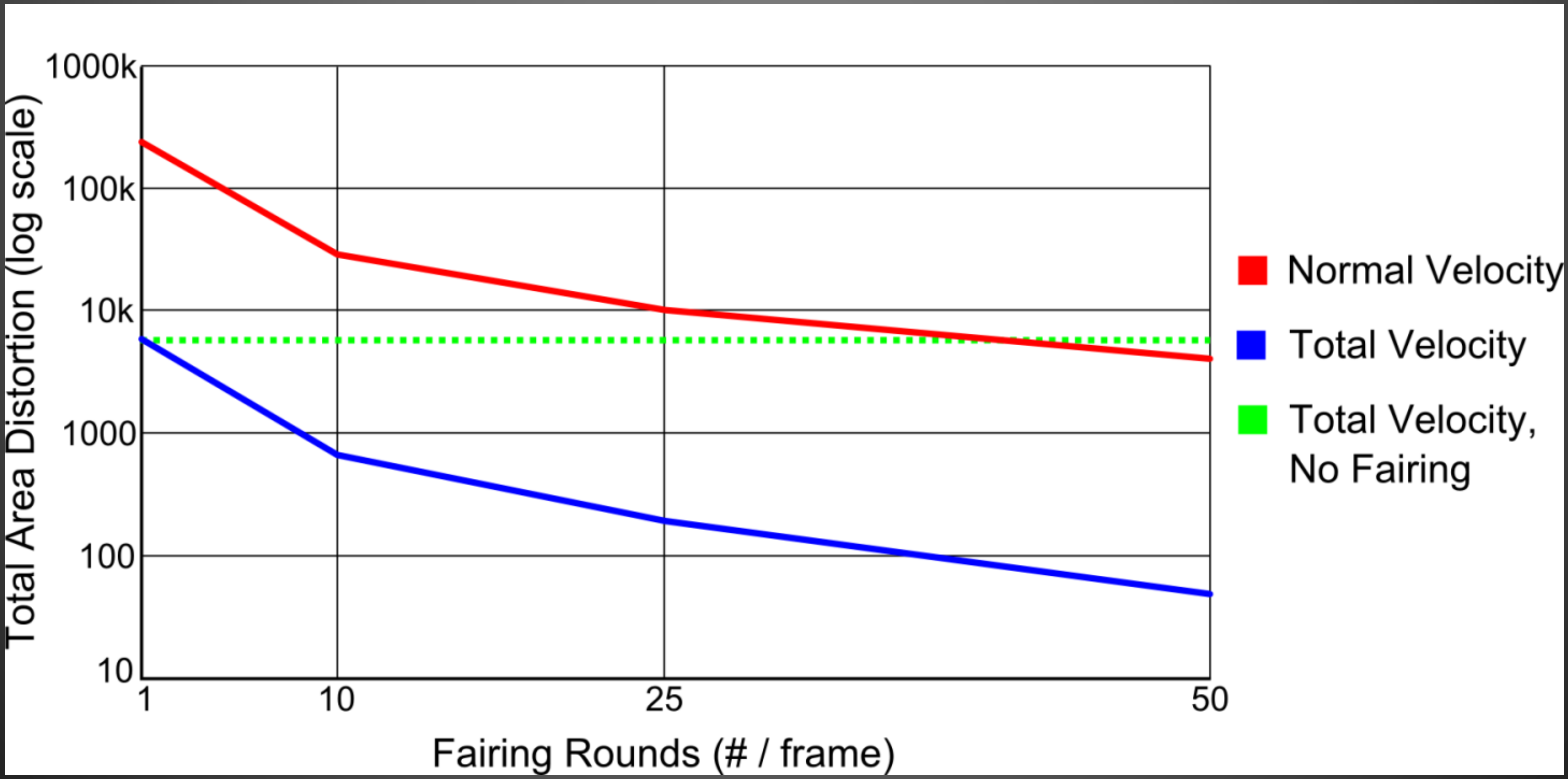
Rotation + Fairing



Normal Velocity   Total Velocity   Normal Velocity w/ Fairing

Rotation + Fairing

Need to do a lot of fairing to Normal Velocity to approach quality of Total Velocity

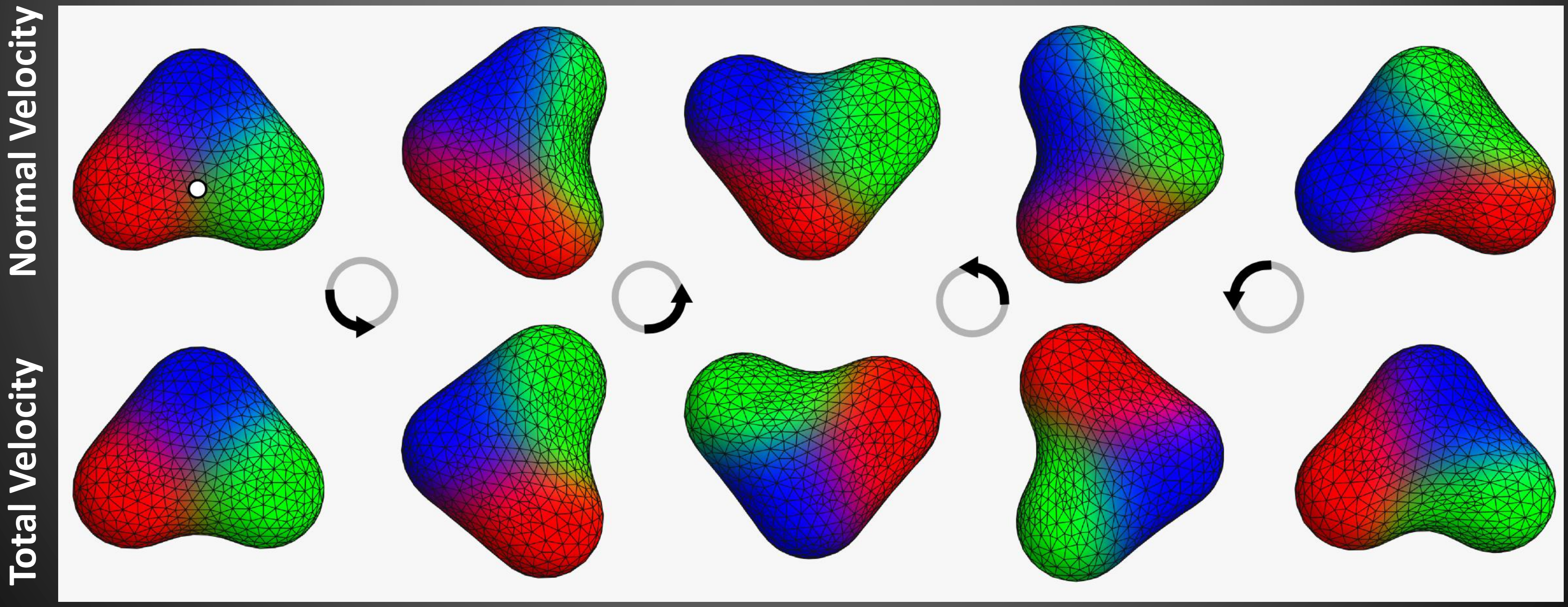


Rotation + Fairing

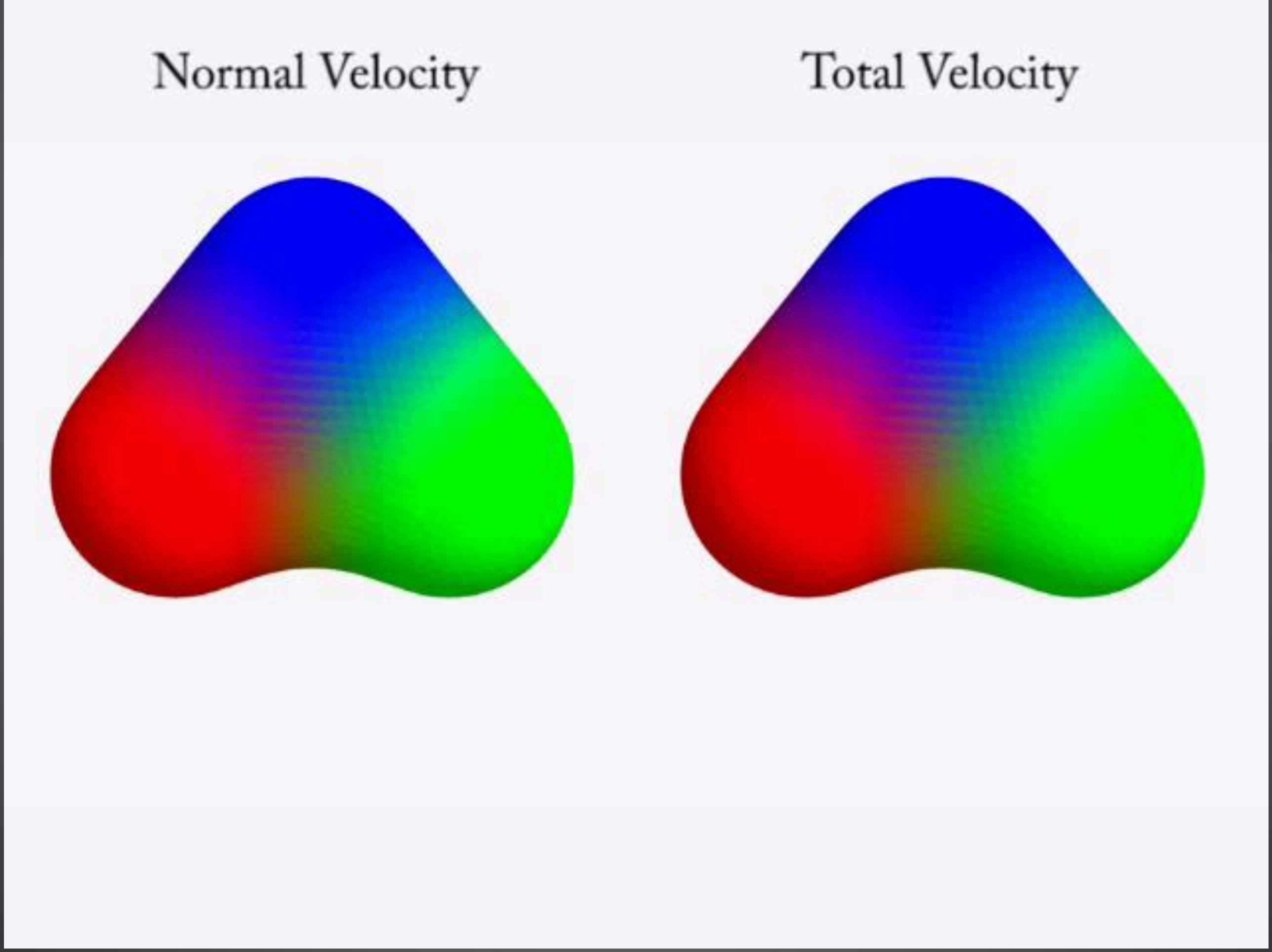
- Fairing is **Expensive**
- Total Velocity still needs a few rounds of fairing in more complex cases
- We can take bigger time steps and less fairing.

Tracking Stability

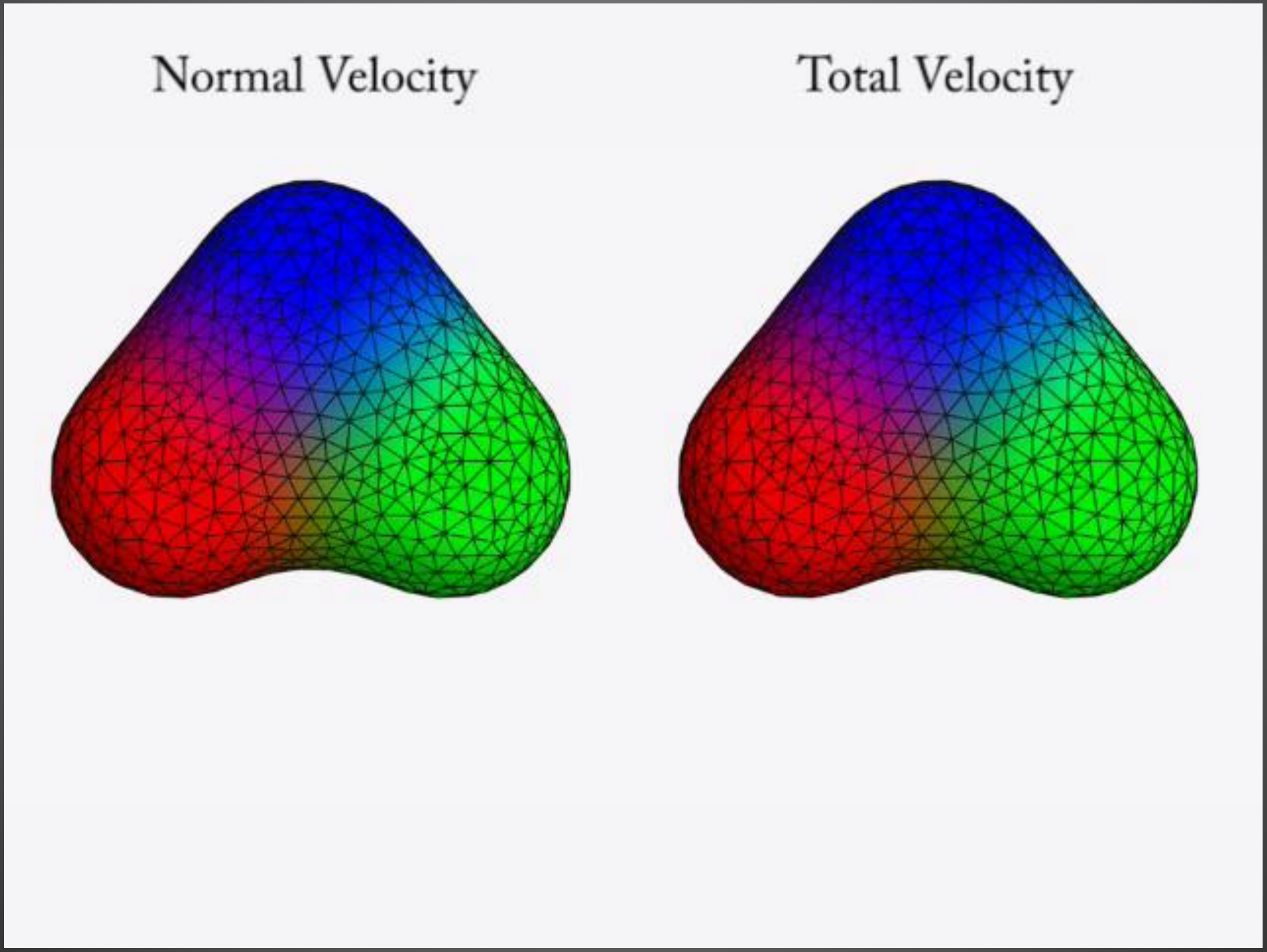
- Normal velocity exhibits much more ‘sliding’ around across moving surface



Tracking Stability

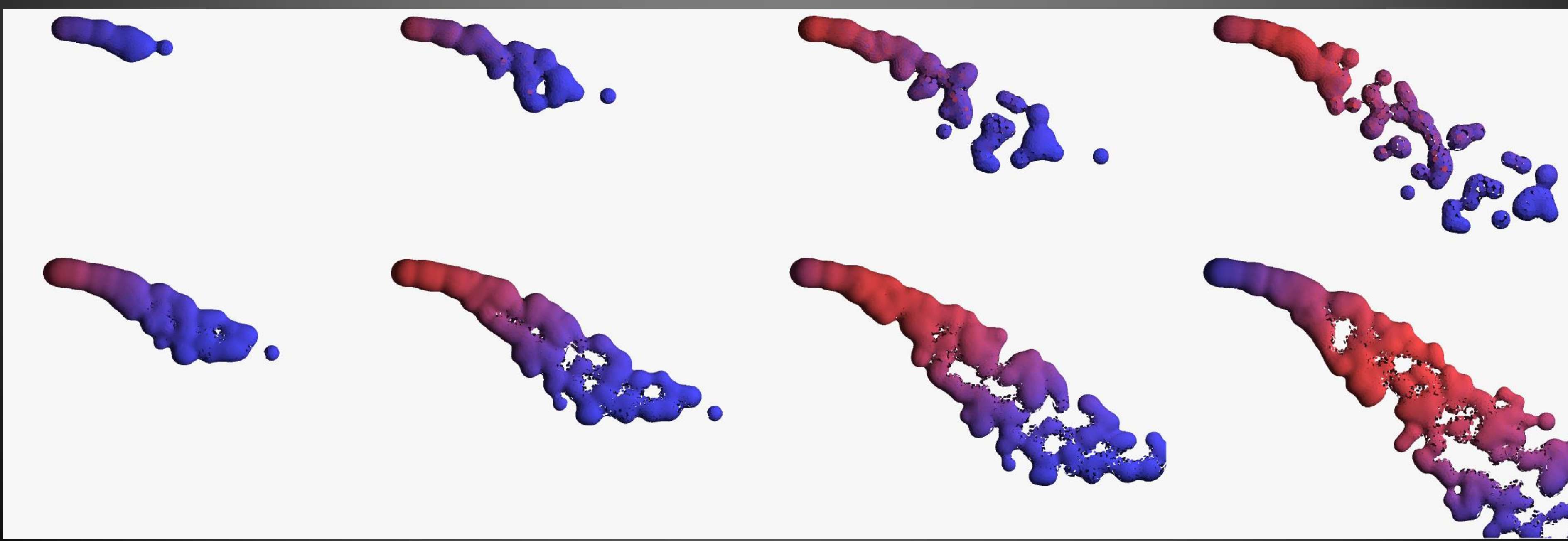


Tracking Stability

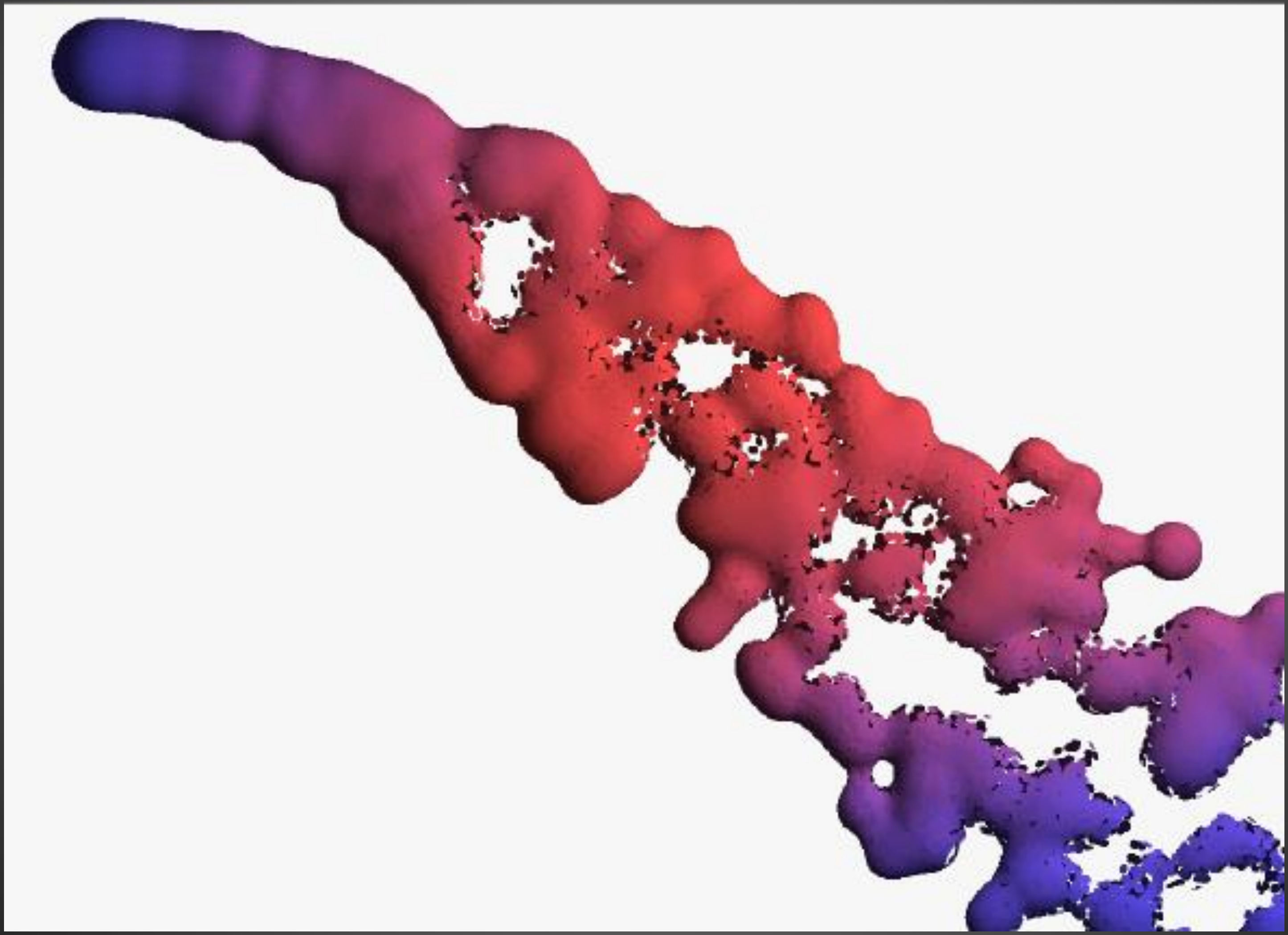


Complex Surface

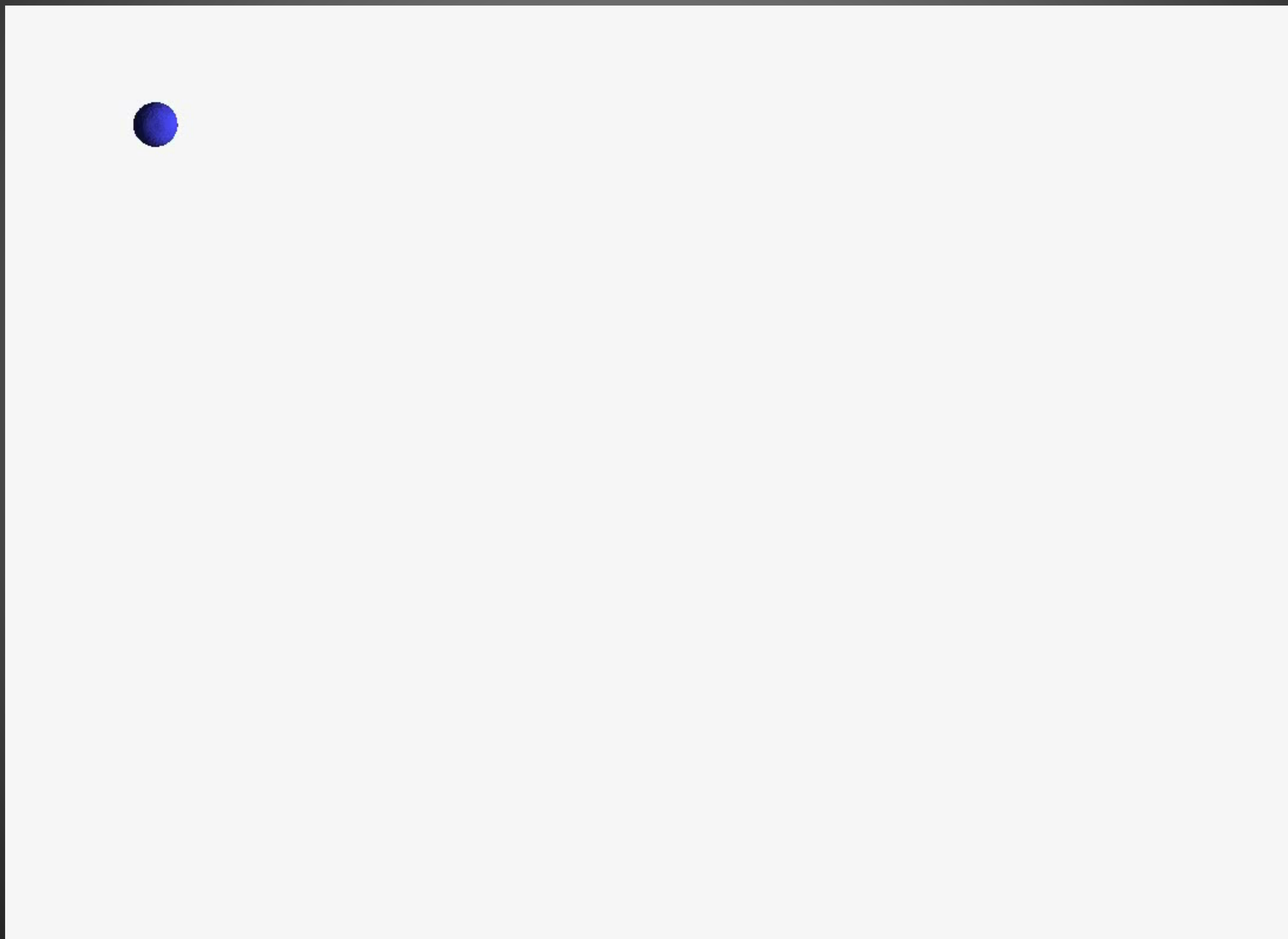
- Normal velocity immediately fails
- Total velocity maintains decent coverage, except where surface tears apart



Complex Surface



Complex Surface



Witkin-Heckbert Fairing

- Maintains better coverage
- Expensive
  - Better performance maybe available (eg Meyer05)

Witkin-Heckbert Fairing

Conclusion

“What is the velocity of an Implicit Surface?”

Normal velocity is uniquely defined.

Free to set tangential velocity
to meet your needs.

Future Work

Other tangent constraints? Theory? Just optimize?

Explore other functions than blobs

Stimulate more work in evolving implicit surfaces.

The End

Thank You

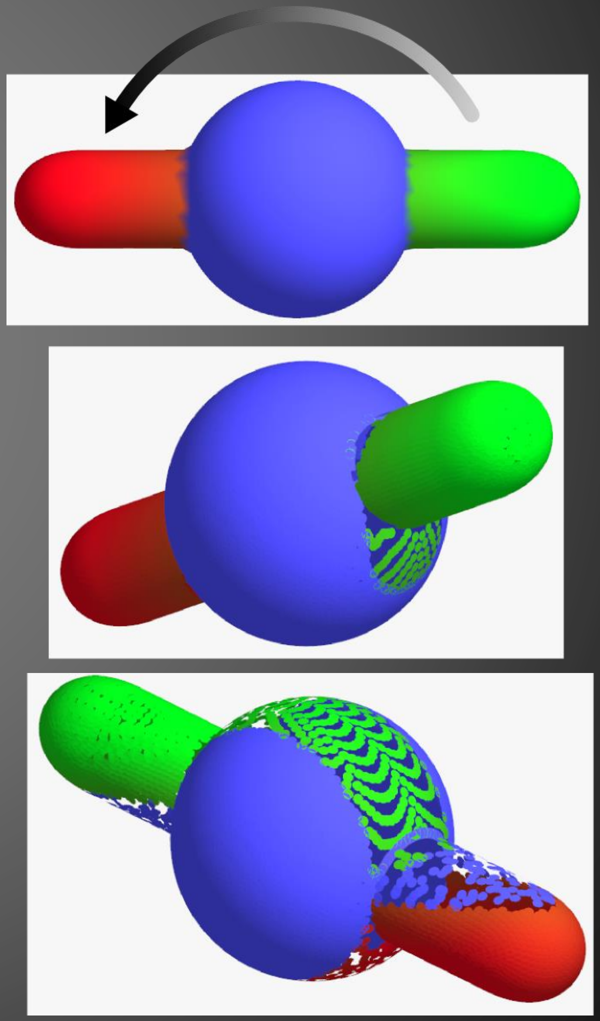
Merci

The End

Questions?

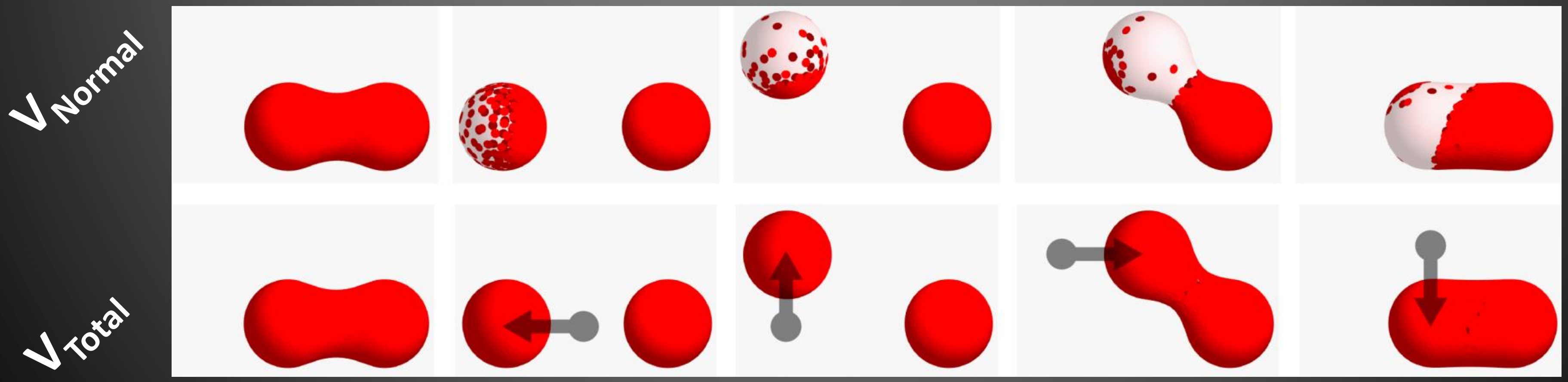
Field Discontinuities

- Gradient undefined
- Problem? Not really:
  - with operators like min/max CSG, always returning one of the well-defined gradients
  - Sampling-based numerical gradients are defined across discontinuities
- In practice, particles just tend to bunch up at creases

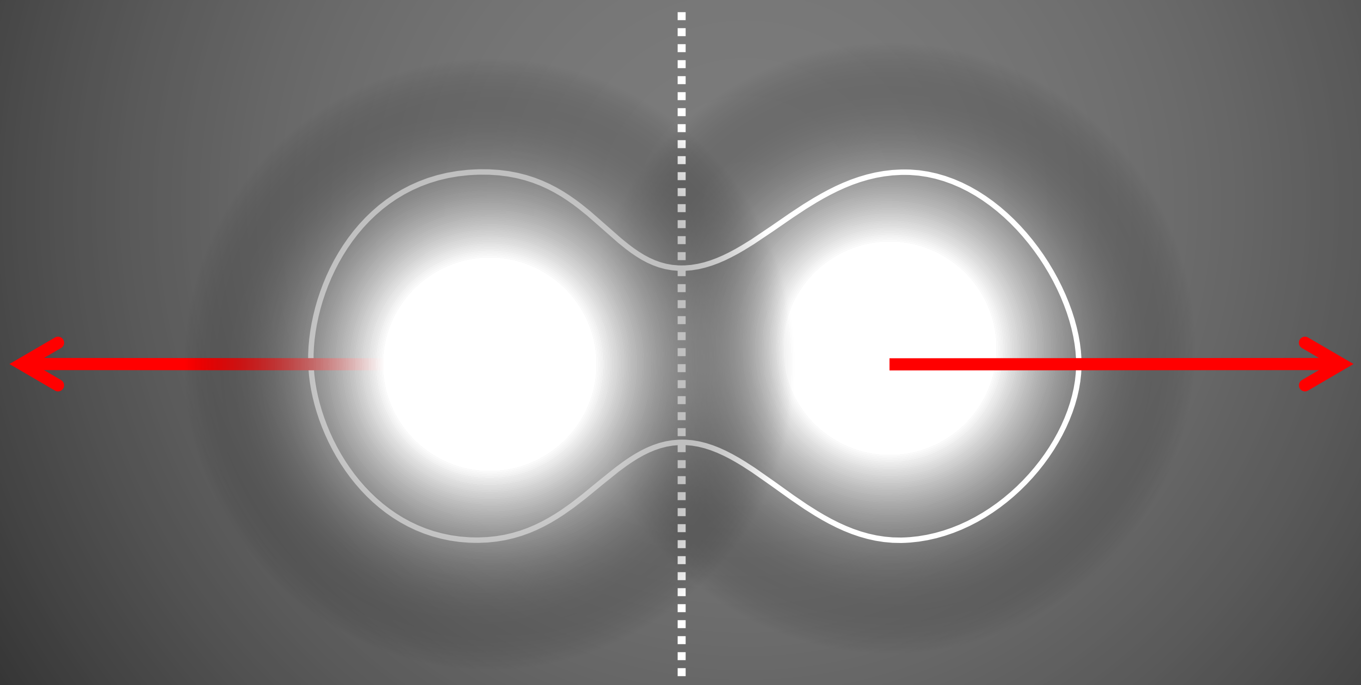


Interactive Visualization

- Infer velocity from interactive manipulations
- Total velocity generally maintains decent coverage during interaction
- Do Witkin-Heckbert redistribution during idle time

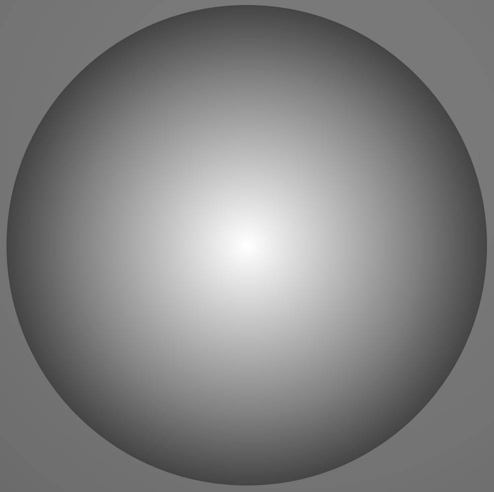


“Obvious Solution”



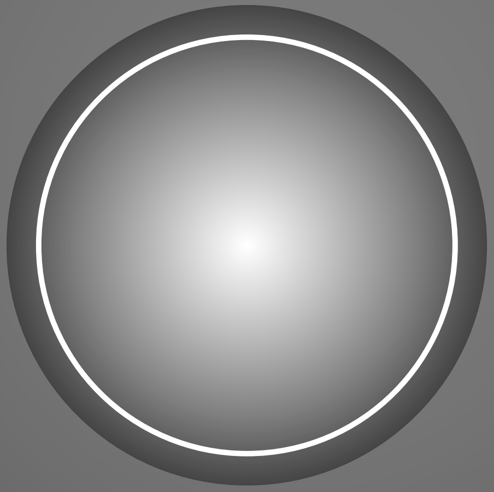
Zero velocity

Implicit Surface



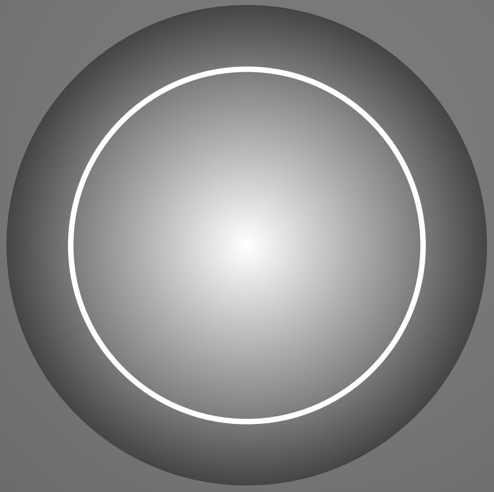
$$F(\mathbf{x})$$

Implicit Surface



$$F(\mathbf{x}) = T_1$$

Implicit Surface



$$F(\mathbf{x}) = T_2$$