Fluid Simulation for Computer Animation

Greg Turk

Why Simulate Fluids?

- Feature film special effects
- Computer games
- Medicine (e.g. blood flow in heart)
- Because it’s fun

Fluid Simulation

- Called *Computational Fluid Dynamics* (CFD)
- Many approaches from math and engineering
- Graphics favors *finite differences*
- Jos Stam introduced fast and stable methods to graphics [Stam 1999]

Navier-Stokes Equations

$$\nabla \cdot \mathbf{u} = 0$$ Incompressibility

$$\underbrace{\mathbf{u}_t}_{\text{Change in Velocity}} = \underbrace{\mathrm{k}\nabla^2\mathbf{u}}_{\text{Diffusion}} - \underbrace{(\mathbf{u}\cdot\nabla)\mathbf{u}}_{\text{Advection}} - \underbrace{\nabla p}_{\text{Pressure}} + \underbrace{\mathbf{f}}_{\text{Body Forces}}$$

Navier-Stokes Equations

$$\nabla \cdot \mathbf{u} = 0 \quad \text{Incompressibility}$$

$$\underbrace{\mathbf{u}_t}_{\text{Change in Velocity}} = \underbrace{k\nabla^2\mathbf{u}}_{\text{Diffusion}} - \underbrace{(\mathbf{u}\cdot\nabla)\mathbf{u}}_{\text{Advection}} - \underbrace{\nabla p}_{\text{Pressure}} + \underbrace{\mathbf{f}}_{\text{Body Forces}}$$

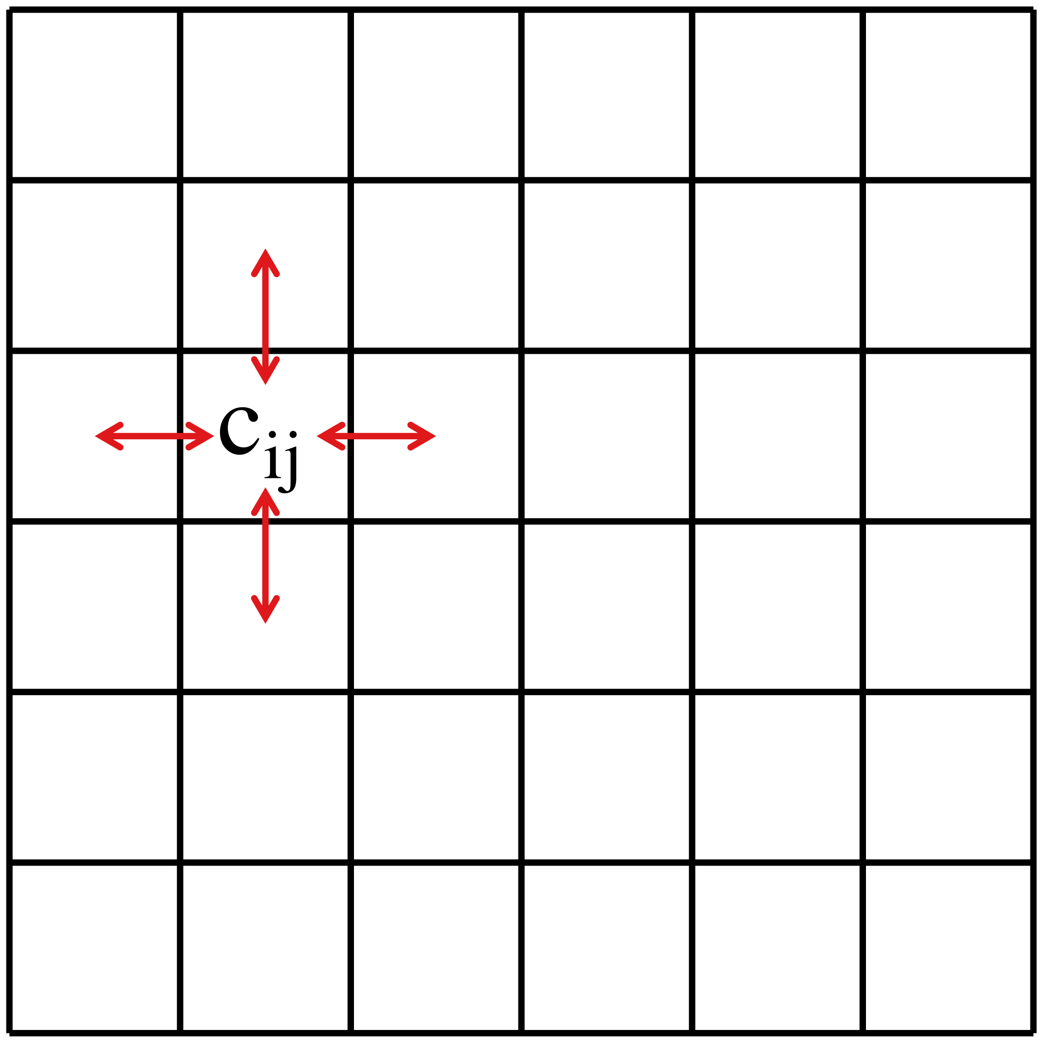
Finite Differences Grids

- All values live on regular grids
- Need *scalar* and *vector* fields
- Scalar fields: amount of smoke or dye
- Vector fields: fluid velocity
- Subtract adjacent quantities to approximate derivatives

Scalar Field (Smoke, Dye)

1.2	3.7	5.1	…		
	c_{ij}				

Diffusion



Diffusion

$$c_t = k\nabla^2 c$$

change in value

value relative to neighbors (Laplacian)

	1	
1	-4	1
	1	

$$c_{ij}^{new} = c_{ij} + k\,\Delta t\,(c_{i-1j} + c_{i+1j} + c_{ij-1} + c_{ij+1} - 4c_{ij})$$

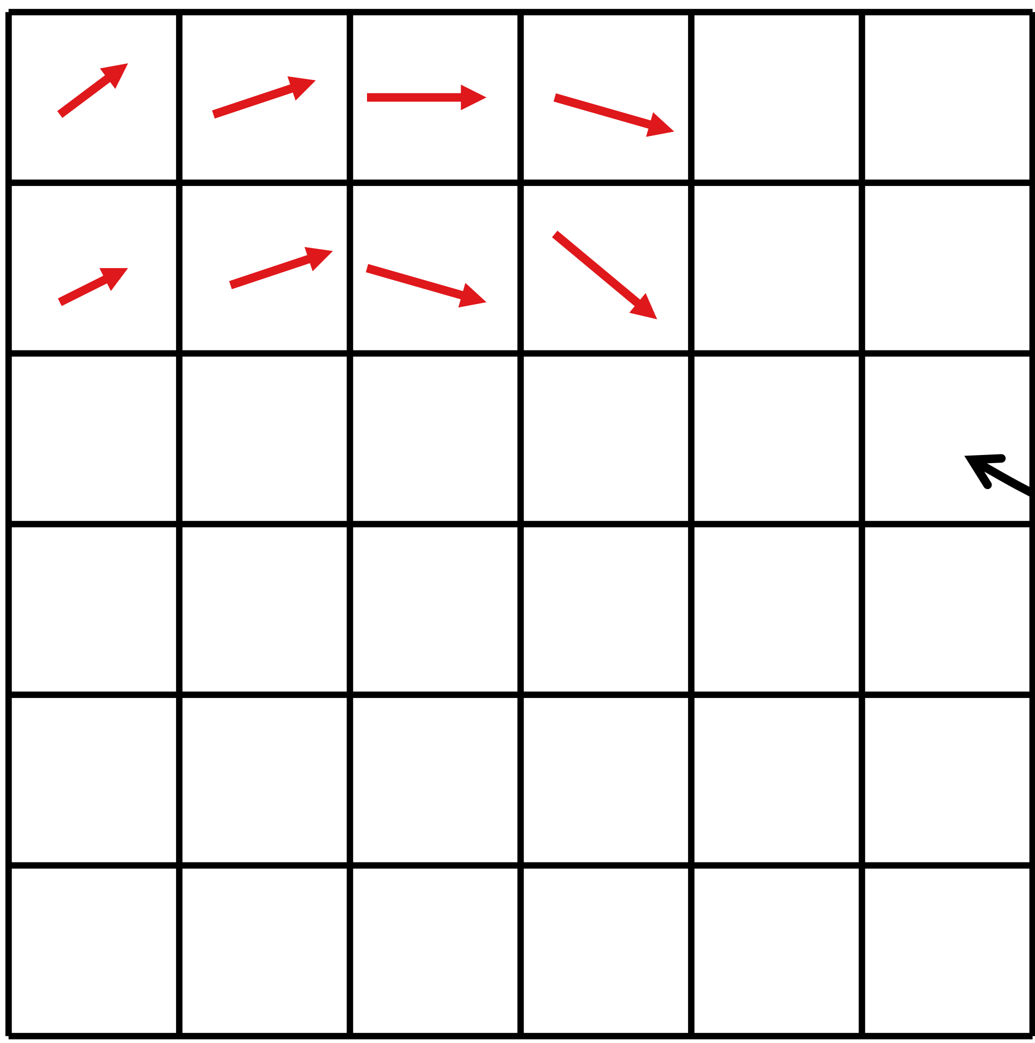
Diffusion = Blurring

Original

Some Diffusion

More Diffusion

Vector Fields (Fluid Velocity)



$\mathbf{u}_{ij} = (u^x, u^y)$

Vector Field Diffusion

$$\mathbf{u}_t = k\nabla^2\mathbf{u}$$

viscosity

Two separate diffusions:

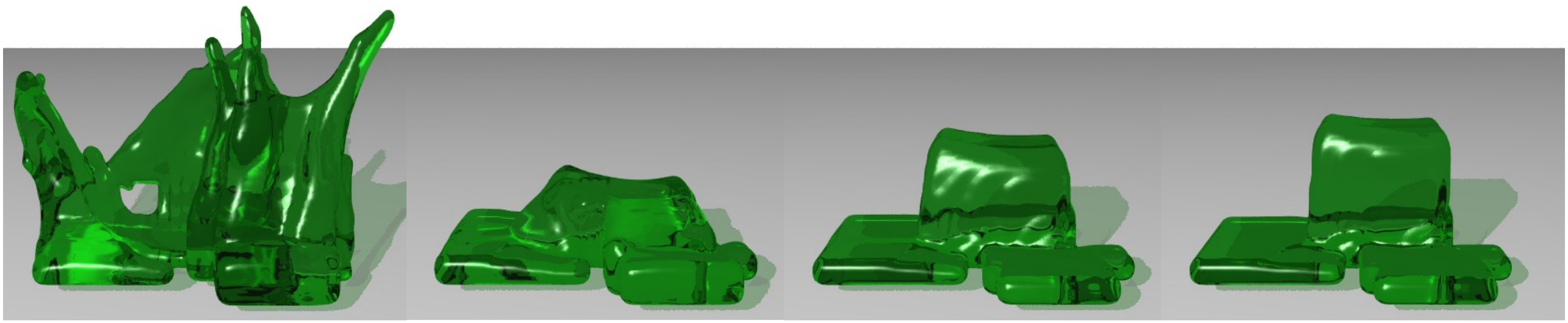
$$u^x{}_t = k\nabla^2 u^x$$

$$u^y{}_t = k\nabla^2 u^y$$

… blur the *x*-velocity and the *y*-velocity

Effect of Viscosity

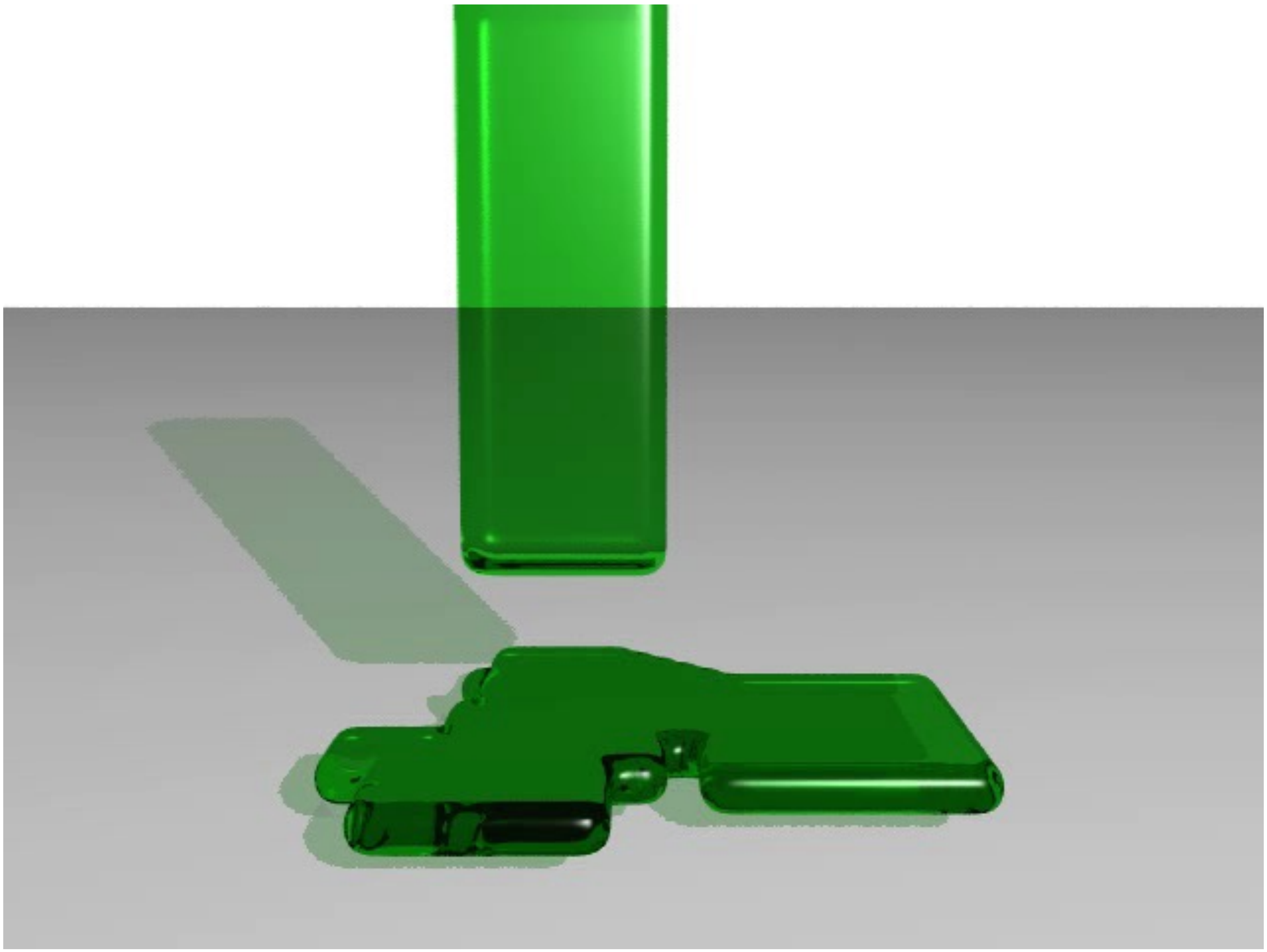
Low Medium High Very High



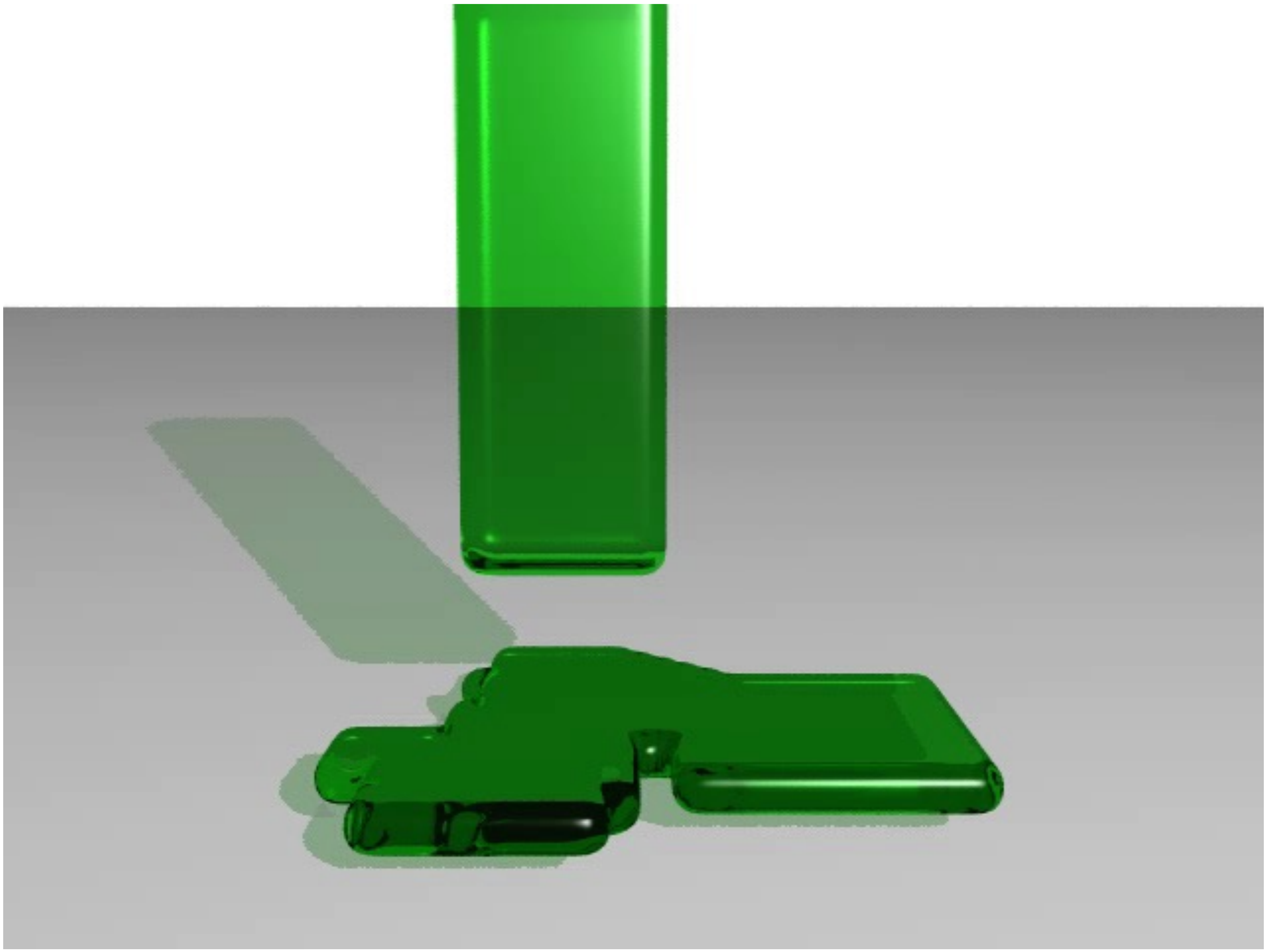
- Each one is ten times higher viscosity than the last

"Melting and Flowing"
Mark Carlson, Peter J. Mucha, Greg Turk
Symposium on Computer Animation 2002

Low Viscosity



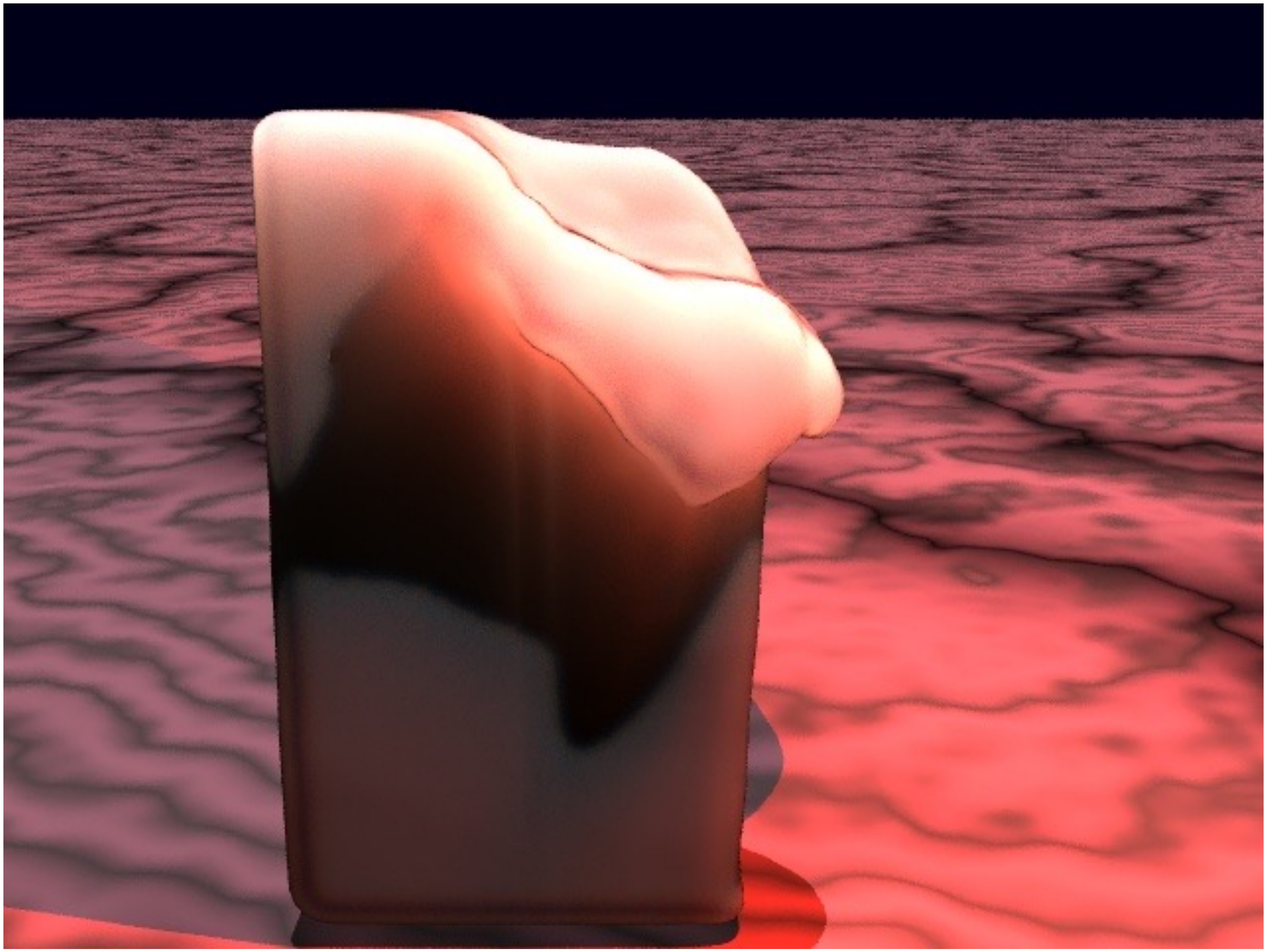
High Viscosity

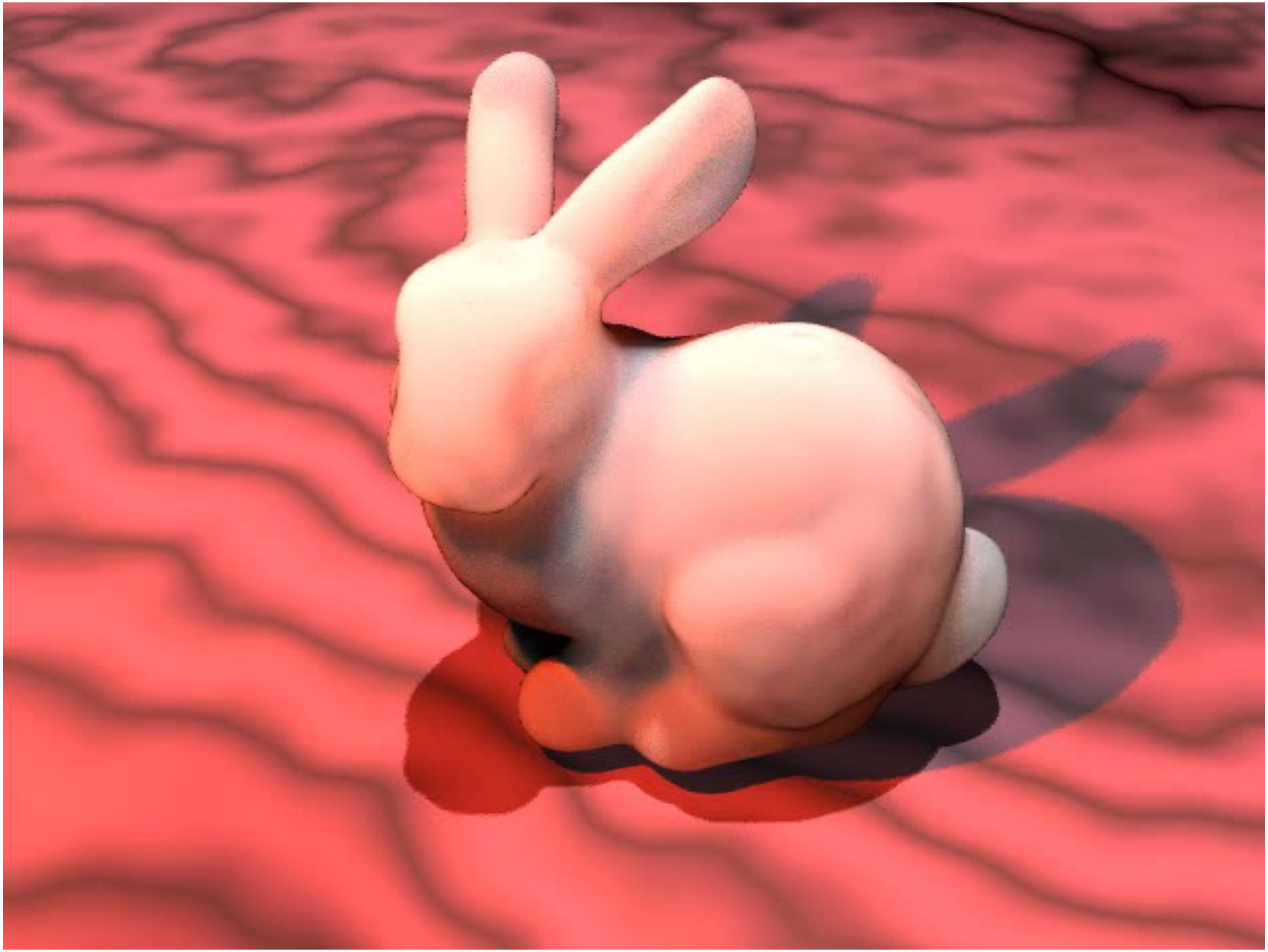


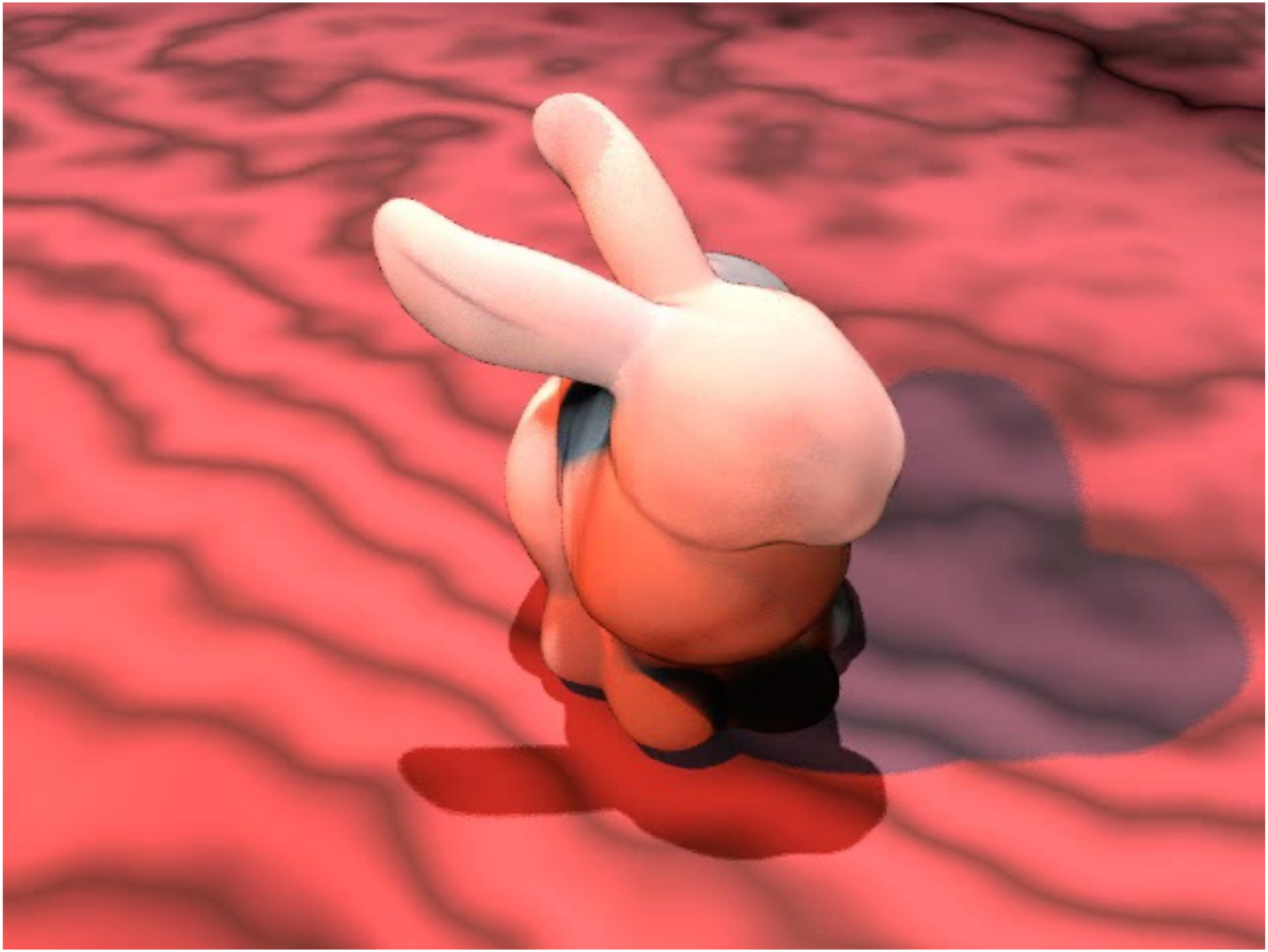
Variable Viscosity

- Viscosity can vary based on position
- Viscosity field *k* can change with temperature
- Need implicit solver for high viscosity

Wax





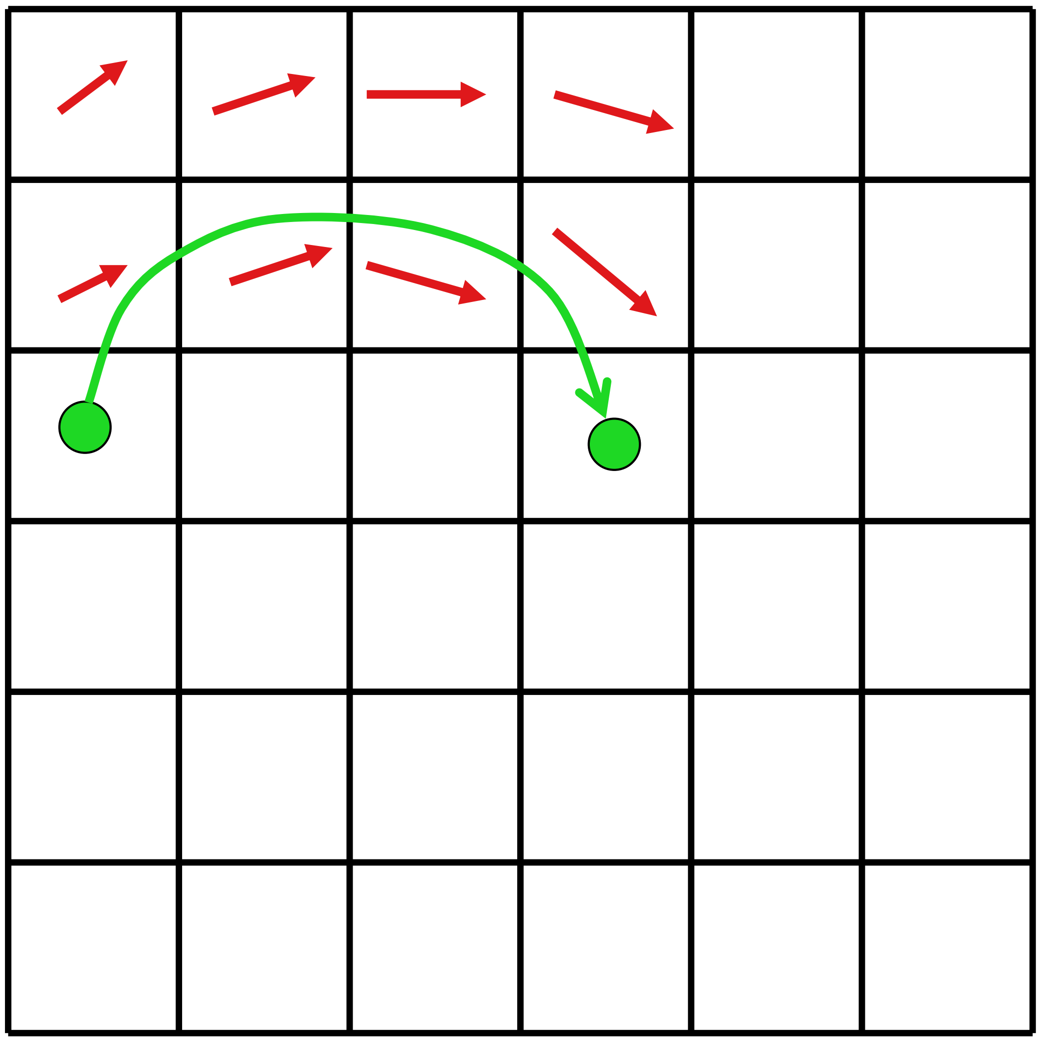


Navier-Stokes Equations

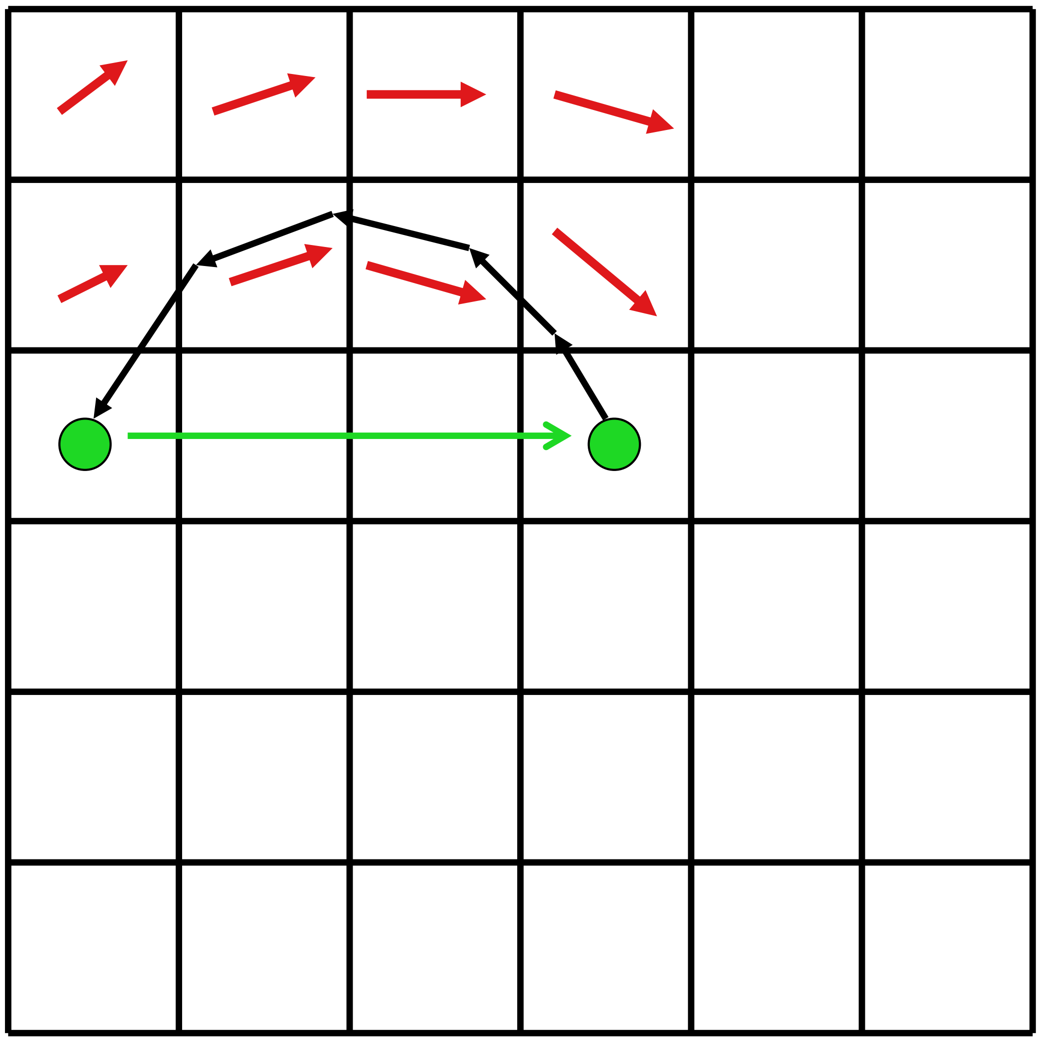
$$\nabla \cdot \mathbf{u} = 0$$ Incompressibility

$$\underbrace{\mathbf{u}_t}_{\text{Change in Velocity}} = \underbrace{k\nabla^2\mathbf{u}}_{\text{Diffusion}} \underbrace{-(\mathbf{u}\cdot\nabla)\mathbf{u}}_{\text{Advection}} \underbrace{-\nabla p}_{\text{Pressure}} + \underbrace{\mathbf{f}}_{\text{Body Forces}}$$

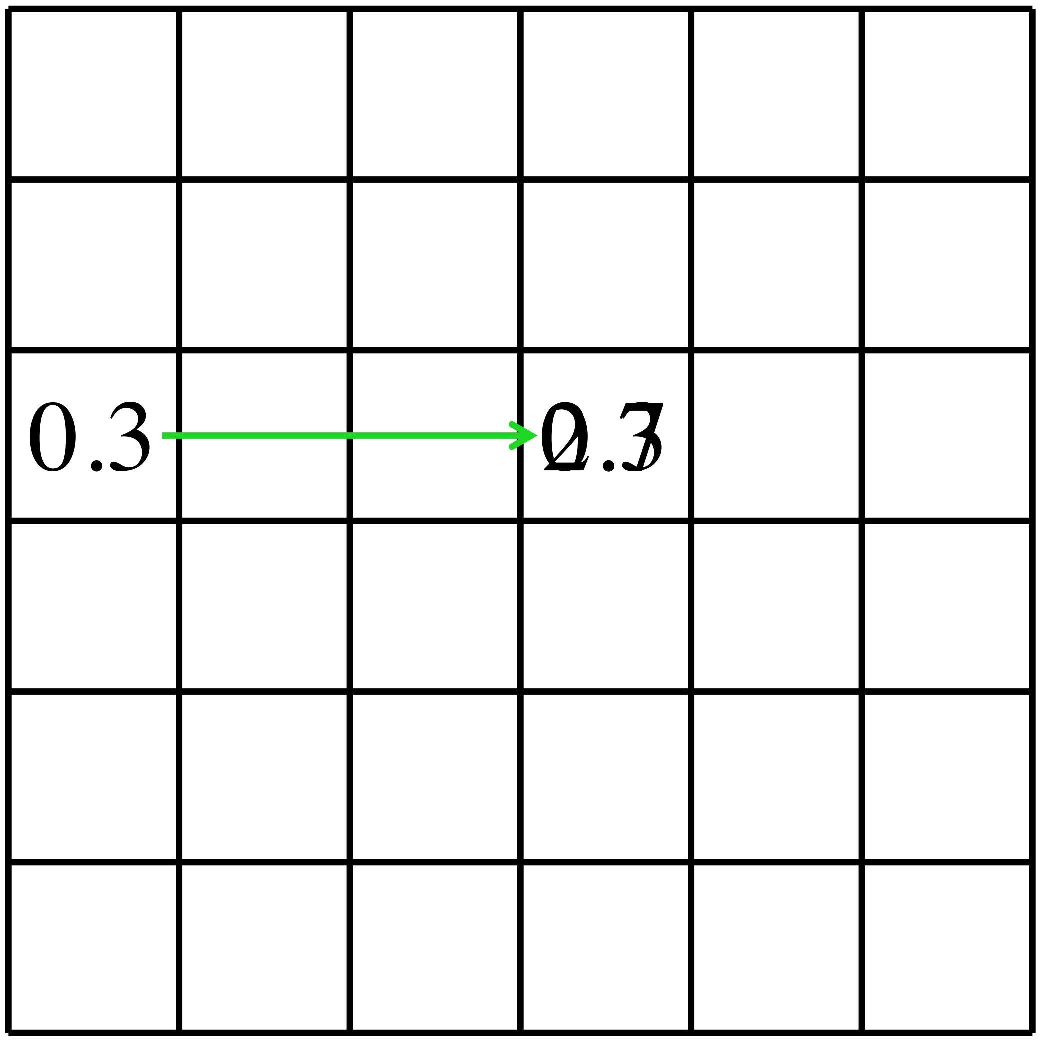
Advection = Pushing Stuff



Advection



Advection



Scalar Field Advection

$$\underbrace{c_t}_{\text{change in value}} = -\underbrace{(\mathbf{u} \cdot \nabla)}_{\text{advection}} \underbrace{c}_{\text{current values}}$$

Vector Field Advection

$$\mathbf{u}_t = -(\mathbf{u}\cdot\nabla)\mathbf{u}$$

Two separate advections:

$$\mathrm{u}^{\mathrm{x}}{}_t = -(\mathbf{u}\cdot\nabla)\mathrm{u}^{\mathrm{x}}$$

$$\mathrm{u}^{\mathrm{y}}{}_t = -(\mathbf{u}\cdot\nabla)\mathrm{u}^{\mathrm{y}}$$

… push around *x*-velocity and *y*-velocity

Advection

- Easy to code
- Method stable even at large time steps
- Important for water and smoke

[redacted]

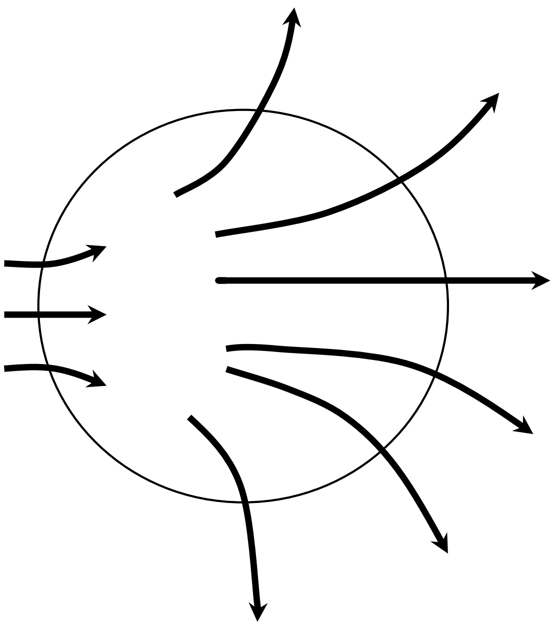
Navier-Stokes Equations

$$\nabla \cdot \mathbf{u} = 0$$

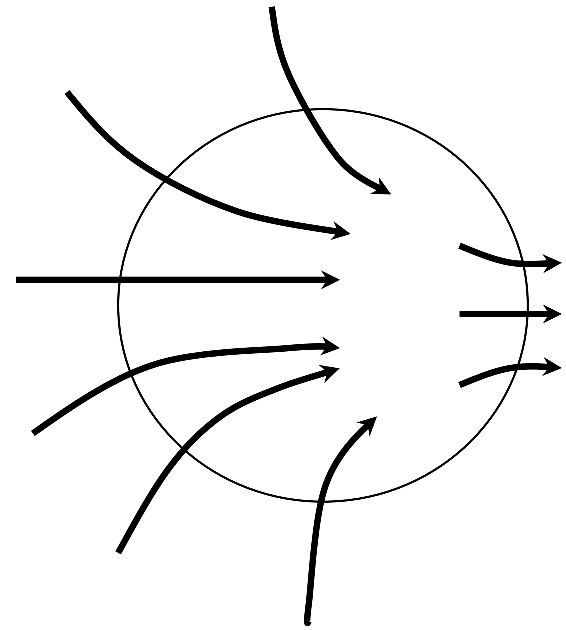
Incompressibility

$$\underbrace{\mathbf{u}_t}_{\text{Change in Velocity}} = \underbrace{\mathrm{k}\nabla^2\mathbf{u}}_{\text{Diffusion}} - \underbrace{(\mathbf{u}\cdot\nabla)\mathbf{u}}_{\text{Advection}} - \underbrace{\nabla p}_{\text{Pressure}} + \underbrace{\mathbf{f}}_{\text{Body Forces}}$$

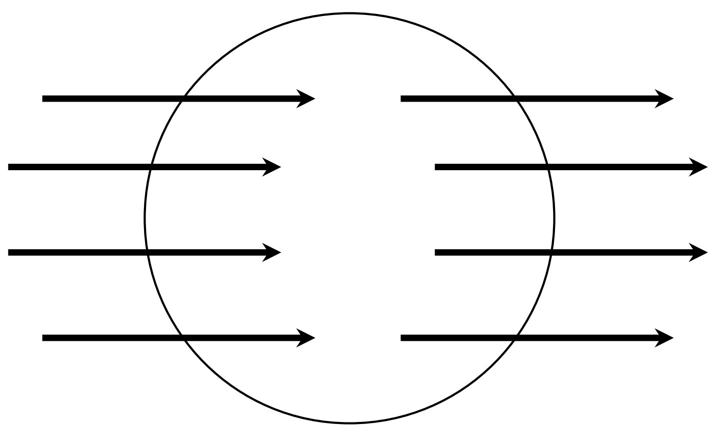
Divergence



High divergence



Low divergence



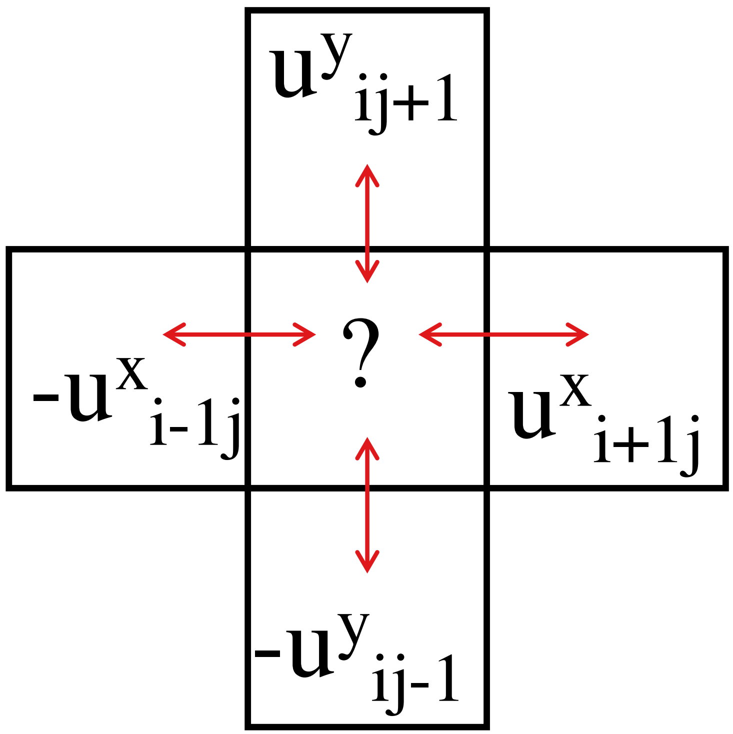
Zero divergence

Enforcing Incompressibility

- First do velocity diffusion and advection
- Find “closest” vector field that is divergence-free
  - Need to calculate divergence
  - Need to find and use pressure

Measuring Divergence

$\nabla \cdot \mathbf{u} = ?$



$$\nabla \cdot \mathbf{u}_{ij} = (u^x_{i+1j} - u^x_{i-1j}) + (u^y_{ij+1} - u^y_{ij-1})$$

Pressure Term

$$\mathbf{u}^{new} = \mathbf{u} - \nabla p$$

Take divergence of both sides…

$$\underbrace{\nabla \cdot \mathbf{u}^{new}}_{\text{zero}} = \nabla \cdot \mathbf{u} - \nabla \cdot \nabla p$$

$$\nabla \cdot \mathbf{u} = \nabla^2 p$$

Pressure Term

$$\underbrace{\nabla \cdot \mathbf{u}}_{\text{known}} = \underbrace{\nabla^2 p}_{\text{unknown}}$$

$$p^{\text{new}} = p + \varepsilon(\nabla \cdot \mathbf{u} - \nabla^2 p)$$

	1	
1	-4	1
	1	

Let $d_{ij} = \nabla \cdot \mathbf{u}_{ij}$

$$p^{\text{new}}_{ij} = p_{ij} + \varepsilon(d_{ij} - (p_{i-1j} + p_{i+1j} + p_{ij-1} + p_{ij+1} - 4p_{ij}))$$

Pressure Term

$$\mathbf{u}^{new} = \mathbf{u} - \nabla p$$

…and velocity is now divergence-free

Found “nearest” divergence-free vector field to original.

Fluid Simulator

1) Diffuse velocity
2) Advect velocity
3) Add body forces (e.g. gravity)
4) Pressure projection
5) Diffuse dye/smoke
6) Advect dye/smoke

“Real-Time Fluid Dynamics for Games”
Jos Stam, March 2003
(CDROM link is to source code)

www.dgp.toronto.edu/people/stam/reality/Research/pubs.html

Rigid Objects

- Want rigid objects in fluid
- Use approach similar to pressure projection

“Rigid Fluid: Animating the Interplay Between Rigid Bodies and Fluid”
Mark Carlson, Peter J. Mucha and Greg Turk
Siggraph 2004

Rigid Fluid Method

1) Solve Navier-Stokes on entire grid, treating solids *exactly as if they were fluid*

2) Calculate forces from collisions and relative density

3) Enforce rigid motion for cells inside rigid bodies

Rigid Fluid: Animating the Interplay Between Rigid Bodies and Fluid

Mark Carlson
Peter J. Mucha
Greg Turk

Georgia Institute of Technology

Sound FX by Andrew Lackey, M.P.S.E.

Small-scale liquid-solid Interactions

What makes large water and small water behave differently?

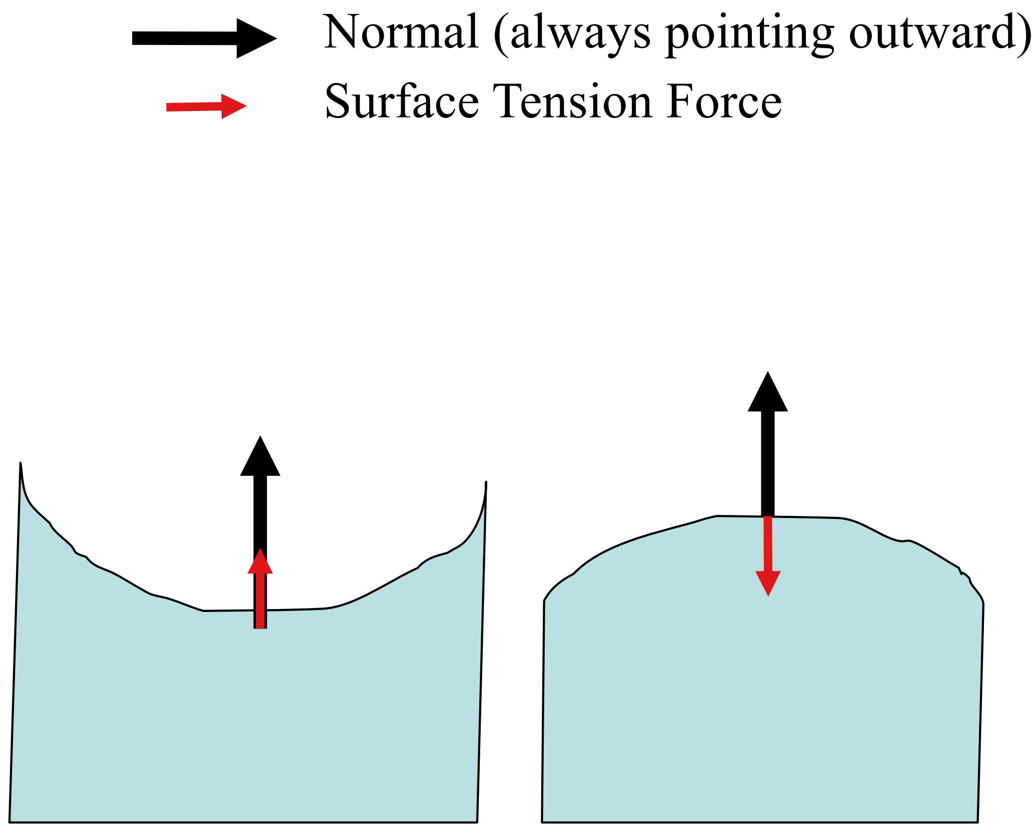
Surface Tension (water: 72 dynes/cm at 25º C)

Viscosity (water: 1.002×10^{-3} N·s/m^2 at 20º C)

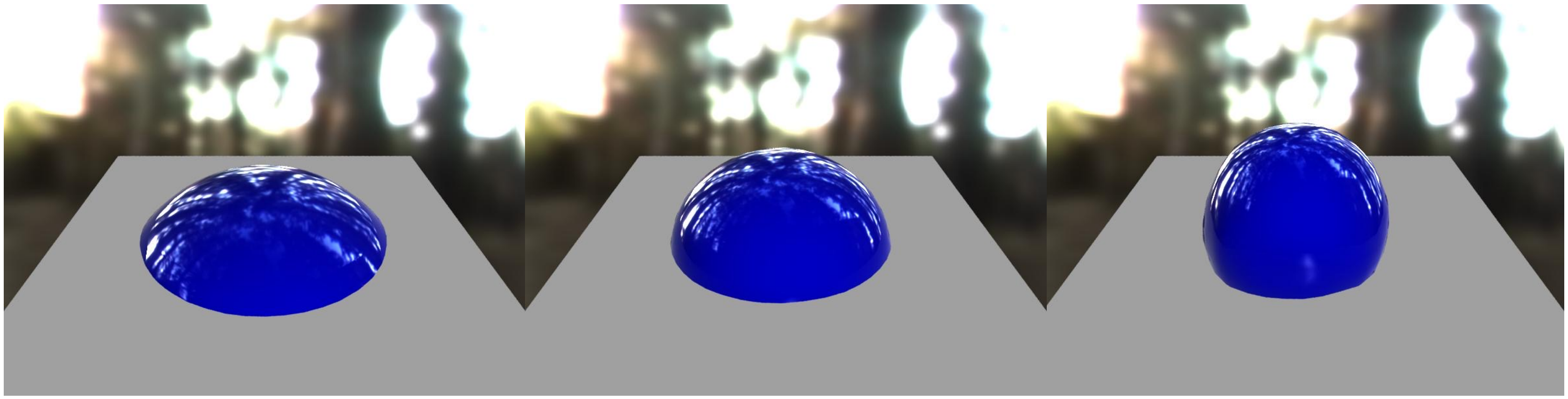
Lake (>1 meter)

Water drops (millimeters)

Surface Tension



Water/Surface Contact

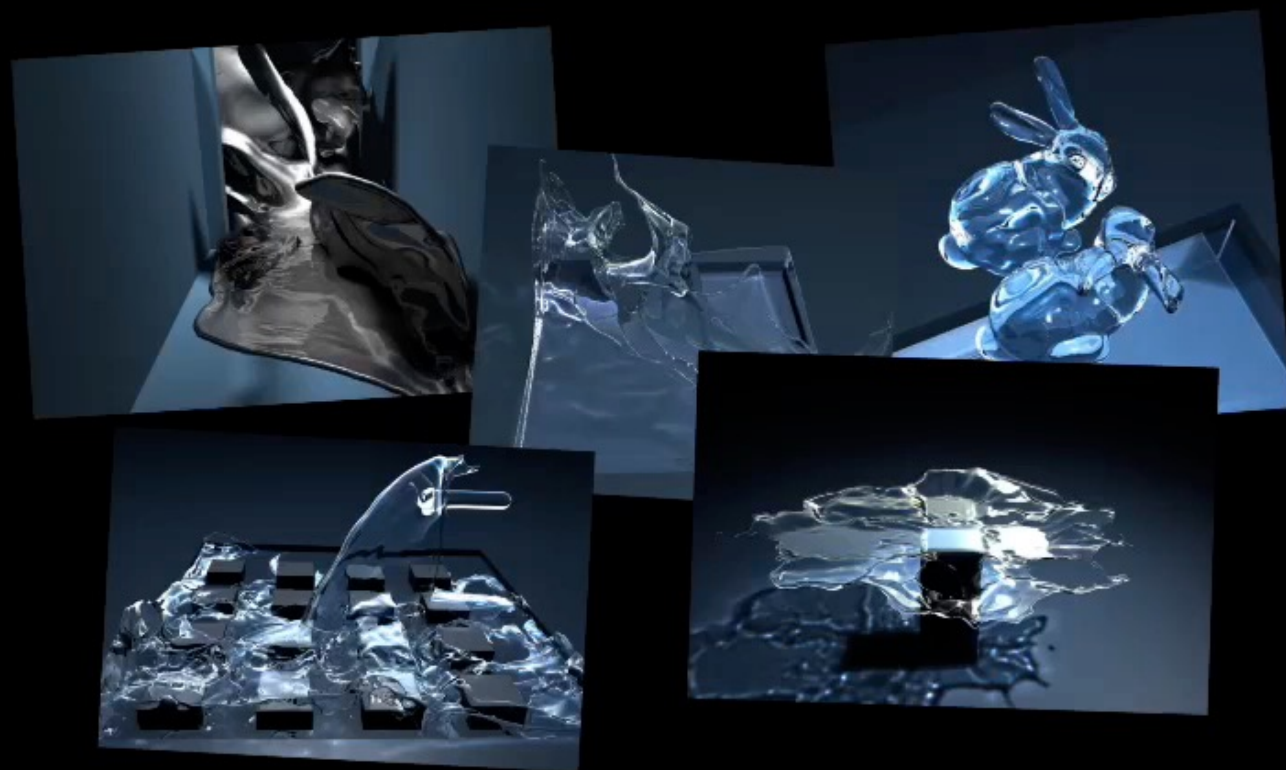


hydrophillic

hydrophobic

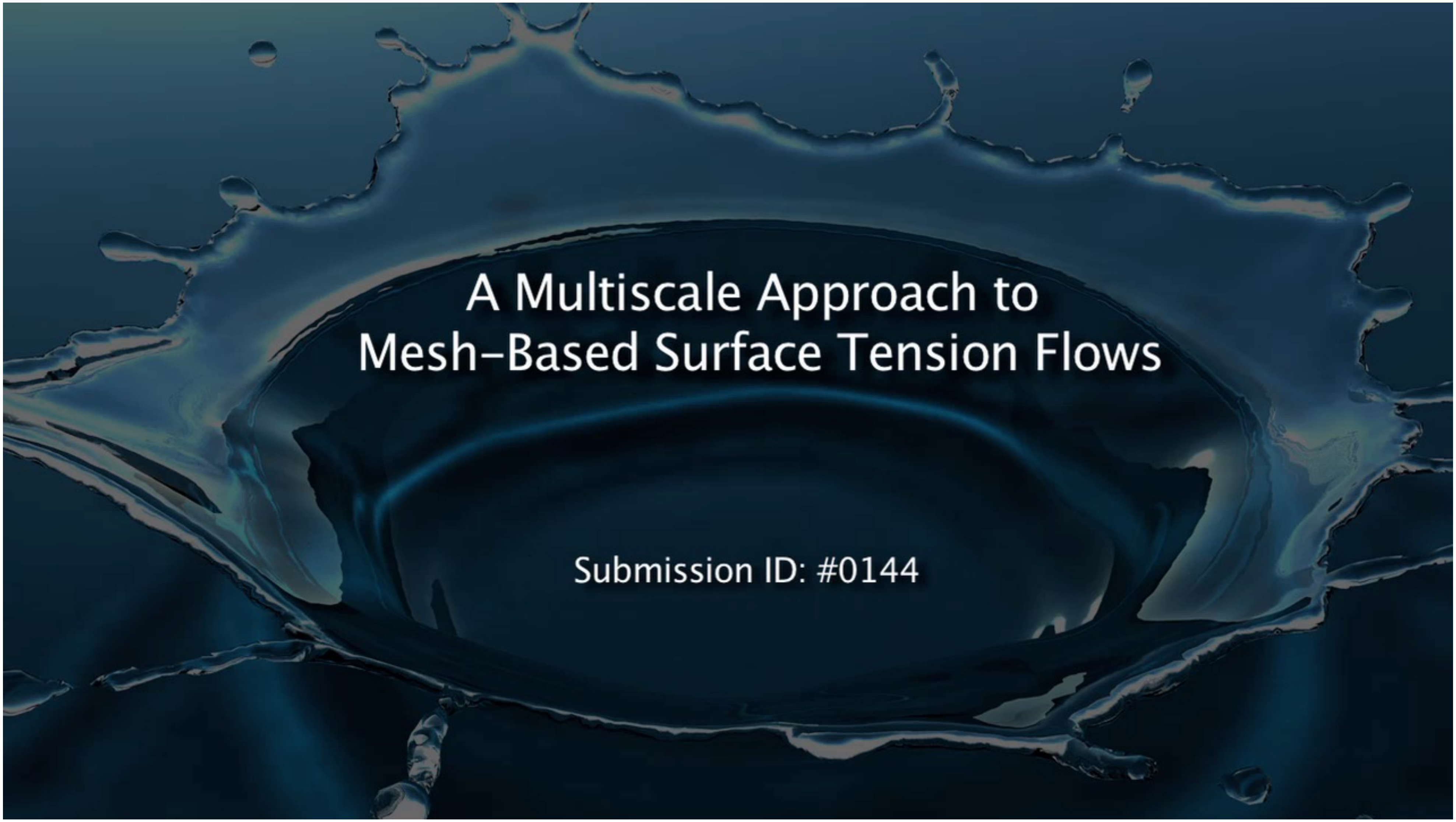
Water Drops on Surfaces

Huamin Wang, Peter J. Mucha, Greg Turk
Georgia Institute of Technology



Physically-Inspired Topology Changes for Thin Fluid Features

submission ID 0304


A Multiscale Approach to
Mesh-Based Surface Tension Flows
Submission ID: #0144

A Moving Eulerian-Lagrangian Particle Method for Thin Film and Foam Simulation

Yitong Deng Mengdi Wang Xiangxin Kong
Shiying Xiong Zangyueyang Xian Bo Zhu

DARTMOUTH

Dartmouth College

End