Physically-Based Simulation Final Project Presentation Waterwheel

Group 12

Ge Cao, Xiao Wu, Mingyang Song

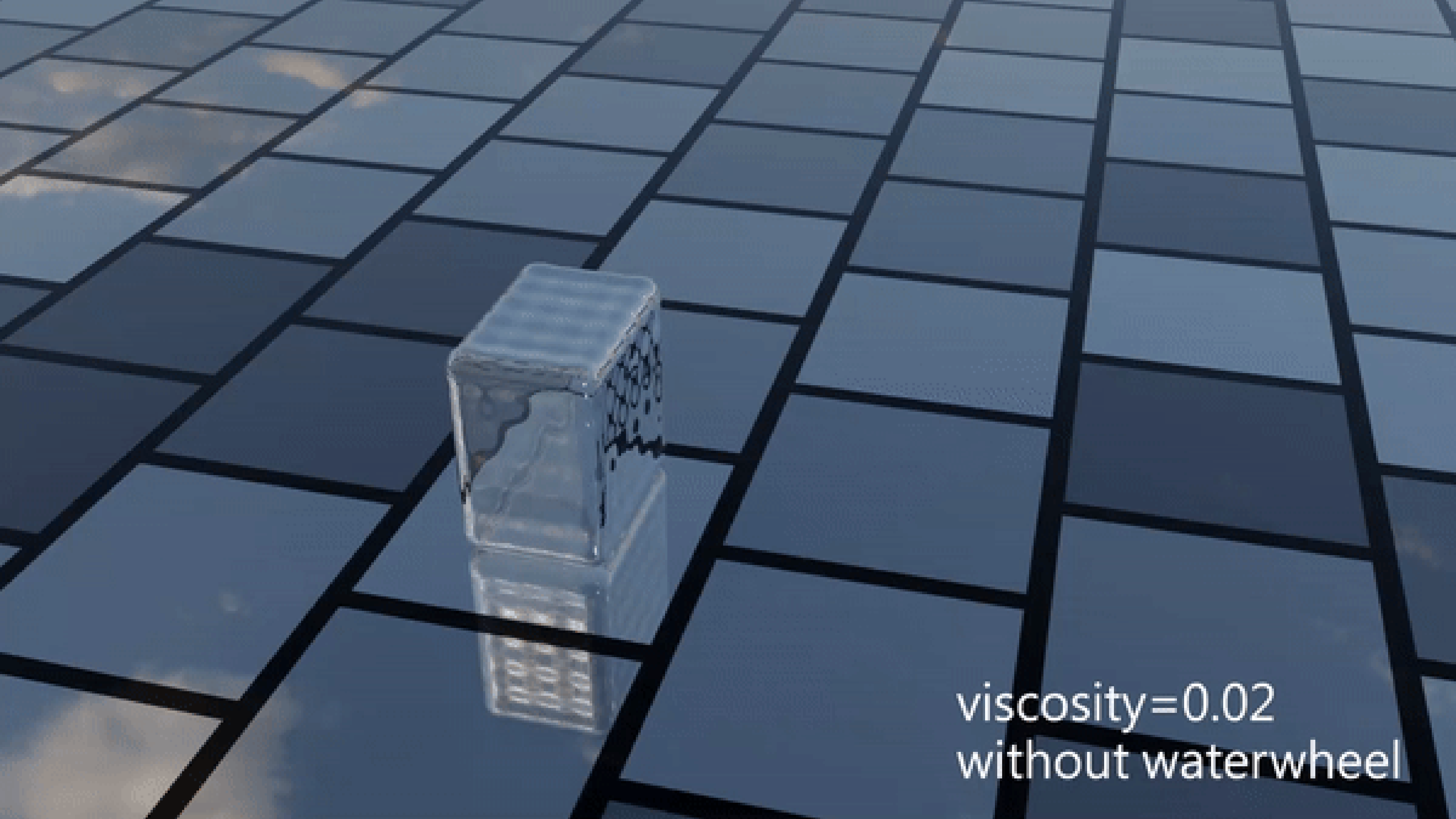
Introduction:

Implemented Features:

- Coupling with Rigid Bodies
- Iterative SESPH
- Particles Visualization (OVITO)
- Surface Reconstruction (splashsurf)
- Particles importing & exporting
- Rendering (Blender)
- Dambreak Scenario
- Waterwheel Scenario

Advanced Features:

- Multithreaded program
- Rendering with GPU


viscosity=0.02
without waterwheel

SPH Method:

- Coupling with Rigid Bodies

  Boundary Handling:
  Several Layers with Uniform Boundary Samples

- Incompressibility

  Iterative SESPH

SPH Pipeline:

Navier-Stokes equation: $\rho \frac{D_v}{D_t} = -\nabla p + \mu \nabla^2 v + f_{ext}$

Algorithm (basic pipeline):

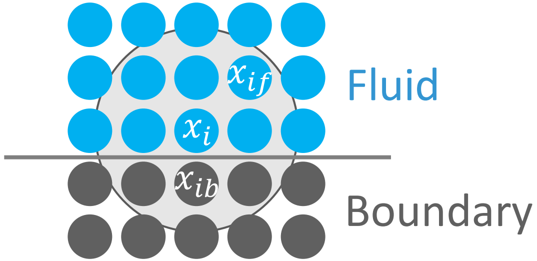
- Update v_i by non-pressure force: $v_i^* = v_i + \Delta t \left(\frac{\mu}{m_i} \sum_j \frac{m_j}{\rho_j} v_{ij} \frac{2\|\nabla_i W_{ij}\|}{\|r_{ij}\|} + \frac{1}{m_i} F_{ext} \right)$
- Determine pressure force F_i^p using state equation: $p_i = k(\rho_i - \rho_0)$

$$F_i^p = \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla_i W_{ij}$$

- Update v_i by solving: $v_i(t + \Delta t) = v_i^* - \frac{\Delta t}{m_i} F_i^p$
- Update x_i by solving: $x_i(t + \Delta t) = x_i(t) + \Delta t v_i(t + \Delta t)$

Boundary Handling:

https://interactivecomputergraphics.github.io/SPH-Tutorial/slides/03_boundary_handling.pdf



To compute F_i^p, boundaries are sampled with particles:

$$\rho_i = m_i \sum_{if} W_{iif} + m_i \sum_{ib} W_{iib}$$

Boundary neighbors contribute to the density

$$p_i = k \left(\frac{\rho_i}{\rho_0} - 1 \right)$$

Pressure at boundary samples: Mirroring

$$a_i^P = -m_i \sum_{if} \left(\frac{p_i}{\rho_i^2} + \frac{p_{if}}{\rho_{if}^2} \right) \nabla W_{iif} - m_i \sum_{ib} \left(\frac{p_i}{\rho_i^2} + \frac{p_{ib}}{\rho_{ib}^2} \right) \nabla W_{iib} \quad p_{ib} = p_i \quad \rho_{ib} = \rho_i$$

$$\Rightarrow a_i^P = -m_i \sum_{if} \left(\frac{p_i}{\rho_i^2} + \frac{p_{if}}{\rho_{if}^2} \right) \nabla W_{iif} - m_i \sum_{ib} \left(\frac{p_i}{\rho_i^2} + \frac{p_i}{\rho_i^2} \right) \nabla W_{iib}$$

Contributions from fluid neighbors

Contributions from boundary neighbors

Mirroring of pressure and density from fluid to boundary

Incompressibility: Iterative SESPH

https://interactivecomputergraphics.github.io/SPH-Tutorial/slides/02_incompressibility.pdf

for all *particle i* **do**
    find neighbors j
for all *particle i* **do**
    $\boldsymbol{a}_i^{nonp} = \nu \nabla^2 \boldsymbol{v}_i + \boldsymbol{g}\ ;\ \boldsymbol{v}_i^* = \boldsymbol{v}_i(t) + \Delta t \boldsymbol{a}_i^{nonp}$ — compute non-pressure acceleration & predict velocity
repeat
    for all *particle i* **do**
        $\rho_i^* = \sum_j m_j W_{ij} + \Delta t \sum_j m_j \left(\boldsymbol{v}_i^* - \boldsymbol{v}_j^*\right) \nabla W_{ij}$
        $p_i = k\left(\frac{\rho_i^*}{\rho_0} - 1\right)$
    for all *particle i* **do**
        $\boldsymbol{v}_i^* = \boldsymbol{v}_i^* - \Delta t \frac{1}{\rho_i^*} \nabla p_i$
until $\rho_i^* - \rho_0 < \eta$ (or iteration>max iteration)
for all *particle i* **do**
    $\boldsymbol{v}_i(t+\Delta t) = \boldsymbol{v}_i^*\ ;\ \boldsymbol{x}_i(t+\Delta t) = \boldsymbol{x}_i(t) + \Delta t \boldsymbol{v}_i(t+\Delta t)$

repeat the step 2 and step 3 in the basic pipeline

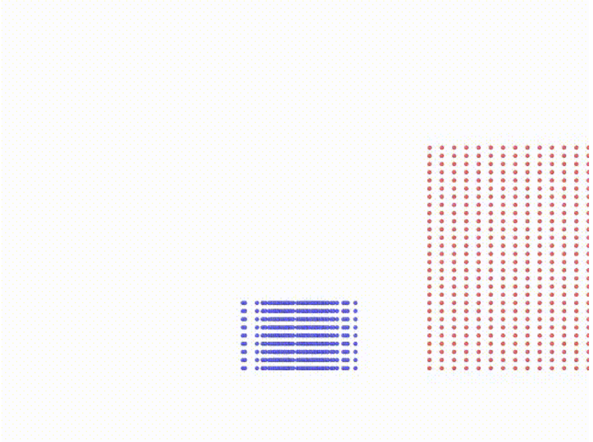
density from predicted position

pressure from predicted density

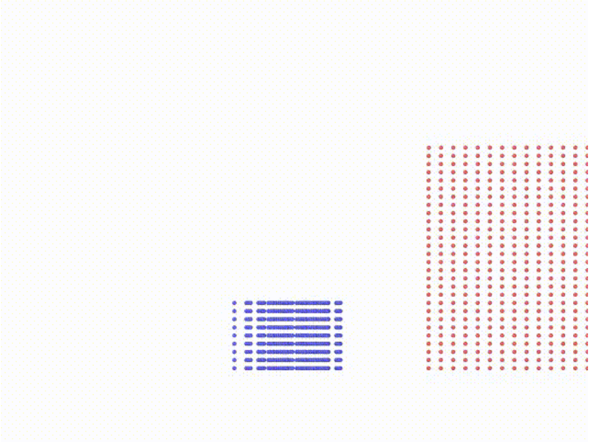
compute pressure acceleration & refine predicted velocity

Liquid with different viscosity

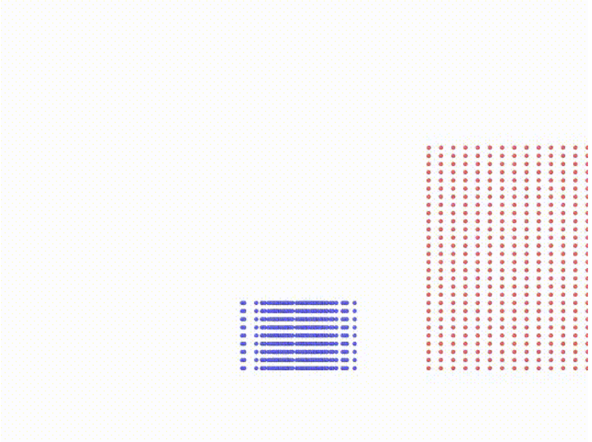
No viscosity can introduce a huge instability



$viscosity = 0$

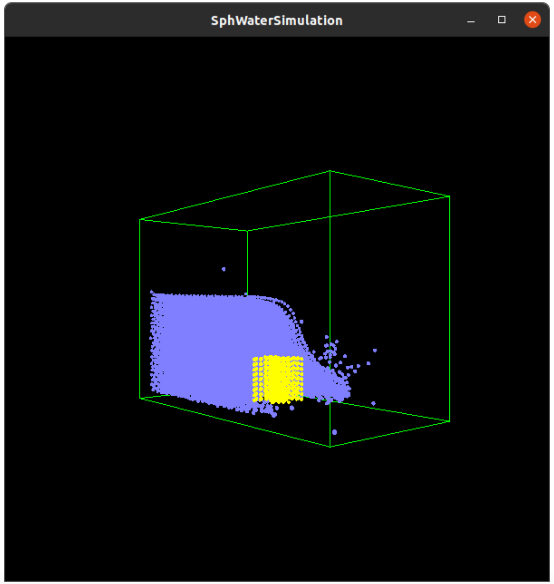


$viscosity = 0.002$

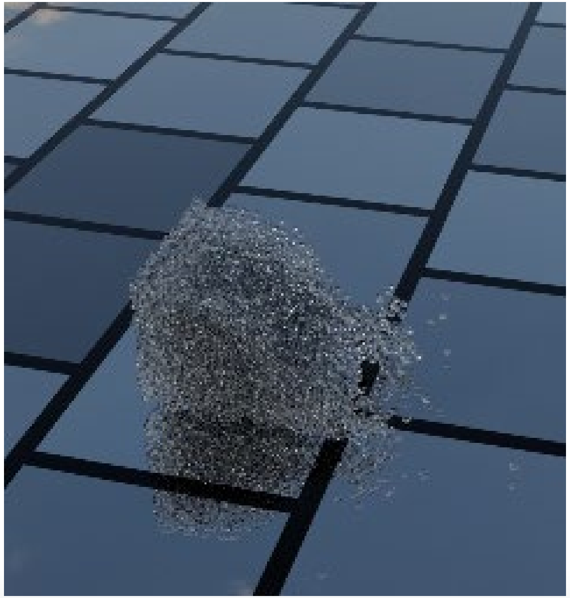


$viscosity = 0.02$

Rendering: rendered as particles

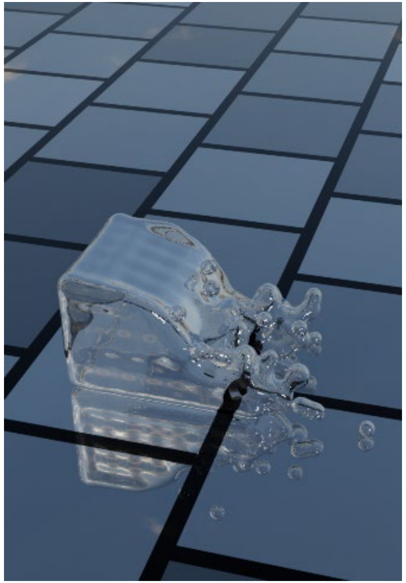


visualized in OVITO

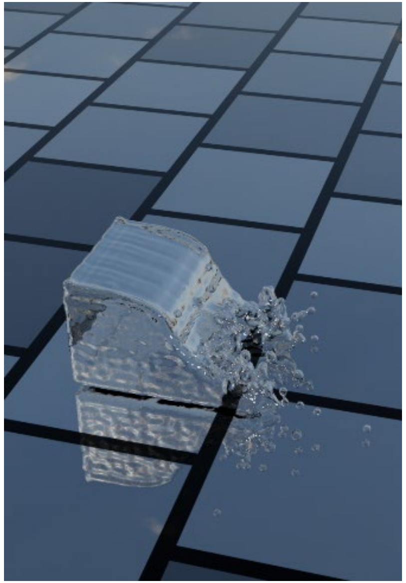


rendered as spheres in Blender

Rendering: rendered with surface reconstruction



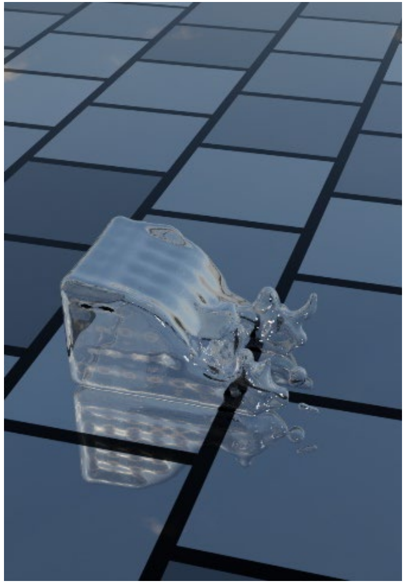
reconstructed surface



smaller kernel size

↓

sharp drops & uneven surface



higher surface threshold

↓

reduce drops number

Surface reconstruction: splashsurf
Rendering: Blender

Performance:

- Multi-threading computing
- Using GPU-based ray-tracing engine

For each frame, it takes 4 seconds to compute particles, 50 seconds to render image using ray-tracing

Thanks for your attention!

Any questions?