Fluid Simulation with Smoothed Particle Hydrodynamics

Katie Clark, Marco Ravelo

May 3, 2024

Intro to SPH

- **Navier-Stokes Equations**
  - Movement
  - Incompressibility
- **Components of fluid motion**
  - Density
  - Pressure
  - Viscosity

u: velocity, p: pressure, ρ: density,
ν : coefficient of viscosity

$$\rho\dot{\mathbf{u}} = -\rho(\nabla \cdot \mathbf{u})\mathbf{u} - \nabla \mathbf{p} + \nu\nabla^2\mathbf{u} + \mathbf{F}$$

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

[1] Pelfrey (2009)

Intro to SPH

- **Navier-Stokes Equations**
  - Movement
  - Incompressibility

- **Components of fluid motion**
  - Density
  - Pressure
  - Viscosity

Convective force

Pressure force

Viscous force

Fluid momentum

$$\rho\dot{\mathbf{u}} = -\rho(\nabla \cdot \mathbf{u})\mathbf{u} - \nabla \mathbf{p} + \nu \nabla^2 \mathbf{u} + \mathbf{F}$$

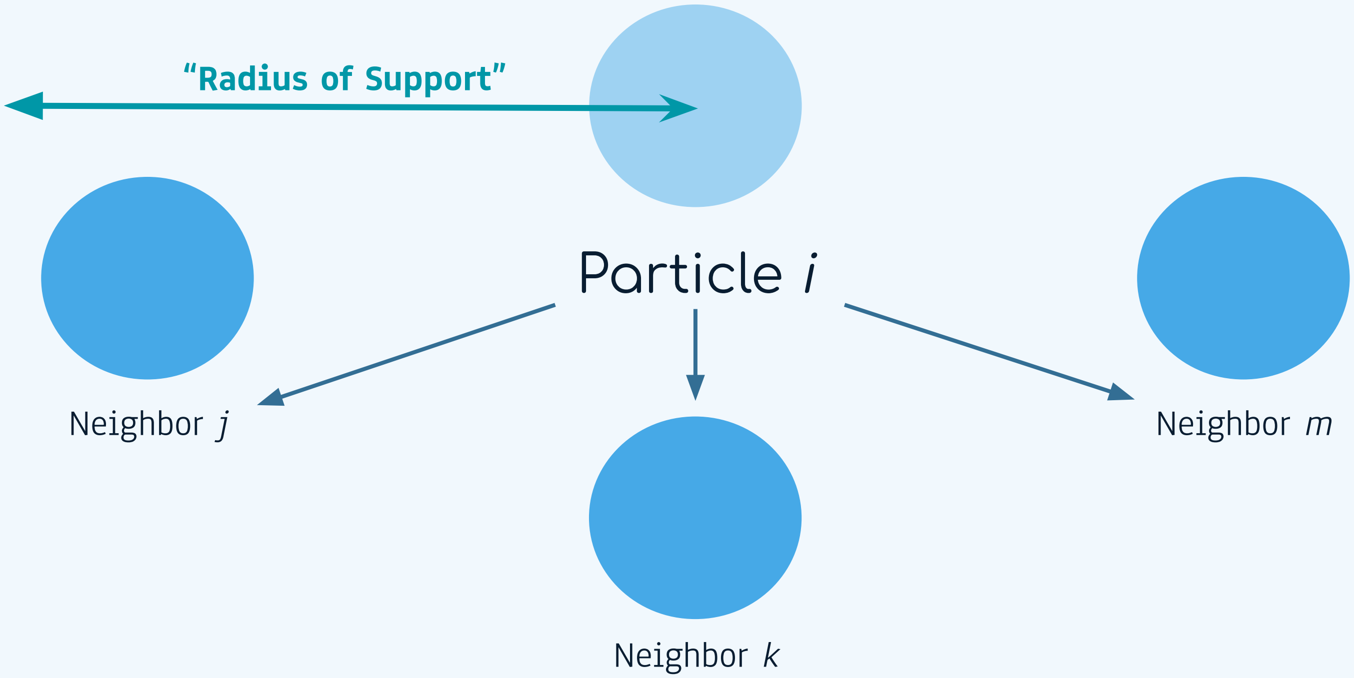
$$\nabla \cdot \mathbf{u} = 0$$

External forces

Incompressibility constraint

Implementation

Particles & Neighbors



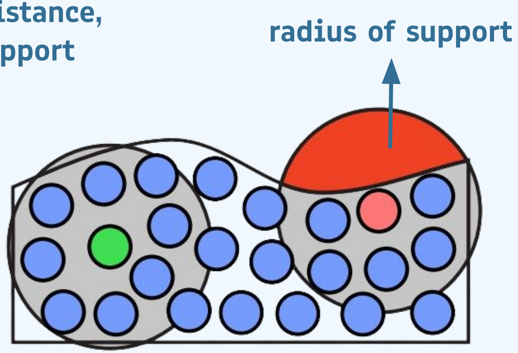
Density Calculation

- Approximate density using masses and kernel function *W*
  - Applies smoothing and defines influence of neighbors based on distance to *i*

$$\rho_i = \sum_j m_j \cdot W(|\mathbf{r}_{ij}|, h)$$

ρ: density, r: relative distance, m: mass, h: radius of support

- Iterate over each particle *i* and each of its neighbors *j*
  - Sum weighted distance from *i* to *j*
  - Add particles within *i*'s "radius of support" to its neighbor list and store weighted distances



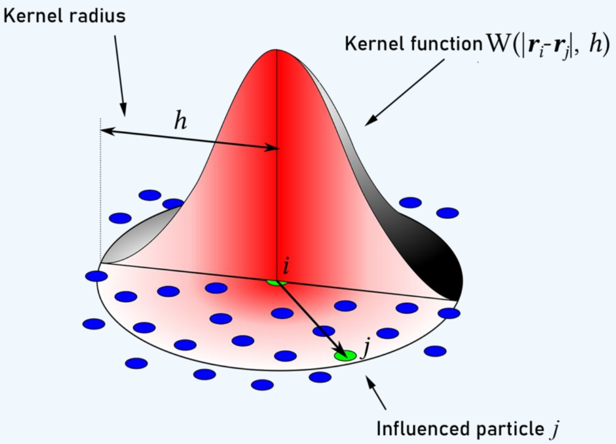
[2] Koschier, D., et al. (2019)

Pressure Calculation

- Determined based on difference between current density and rest density scaled by a fluid pressure constant *k* and weighted distance

$$P_i = k \sum_j (W(|\mathbf{r}_{ij}|, h) \cdot ((\rho_i - \rho_{\text{rest}}) + (\rho_j - \rho_{\text{rest}})))$$

- Iterate over each particle *i*
  - Compute pressure difference
  - For each neighbor *j*
    - Scale pressures by weighted distance
    - Sum pressure force between *i* and each of its neighbors to find the total pressure force at *i*



[4] Wikipedia

Viscosity Calculation

- Fluid resistance to flow and deformation
- Apply approximated viscosity impulse based on:
  - Difference in velocity between a particle and its neighbors
  - Weighted distance

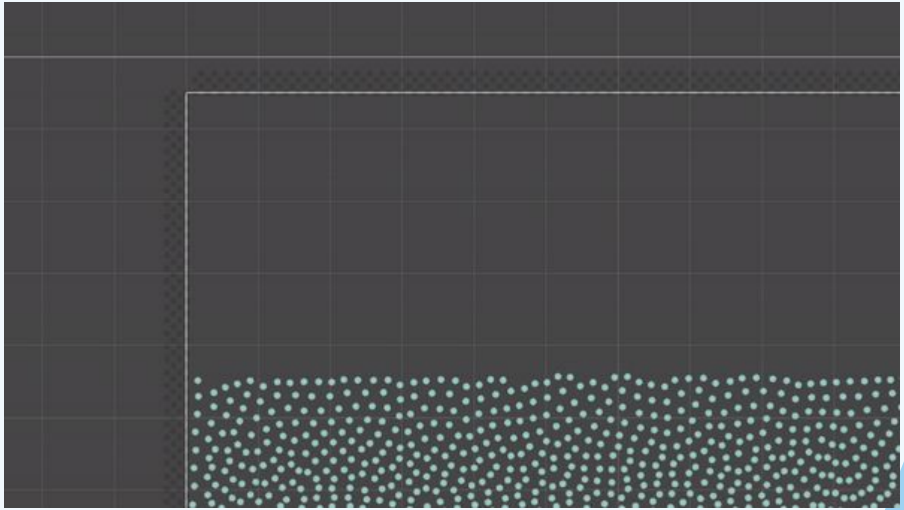
$$I = \sum_j \frac{m_j}{\rho_j} \cdot \sigma \left(\mathbf{v}_j - \mathbf{v}_i \right) \cdot \beta W(|\mathbf{r}_{ij}|, h)$$

- Constants σ and β allow users to tune the viscosity behavior
  - where ($0 \leq \sigma < 1$) and ($0 \leq \beta < 1$)
- Approximates:

$$I = \sum_j \frac{m_j}{\rho_j} \cdot (\mathbf{v}_j - \mathbf{v}_i) \cdot \nabla^2 W(|\mathbf{r}_{ij}|, h)$$

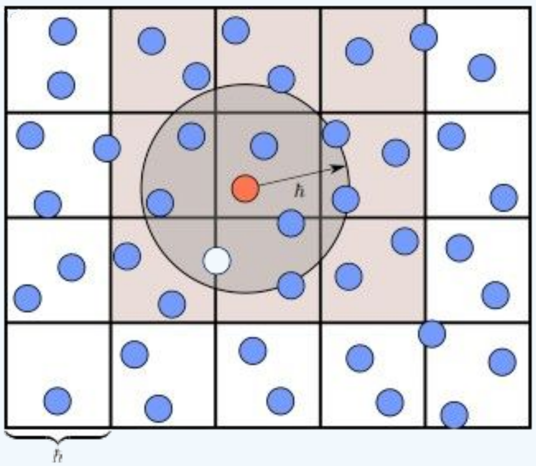
Boundary Conditions

- "Ghost" Particles
  - Multiple layers of fixed particles at the boundary
  - First line of defence
- Fall-back Constraint
  - After updating particle positions, check if any are past the boundary
  - Reposition particles within boundary



Spatial Partitioning

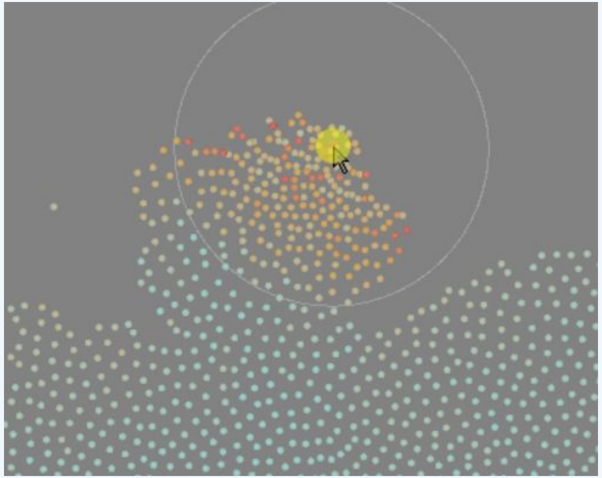
- Divides the space into square (or cube) containers of side length equal to the "radius of support"
- Significantly improves performance by reducing the number of computations needed to find neighbors
- Uses a dictionary with keys based on particle positions in order to place them in respective containers

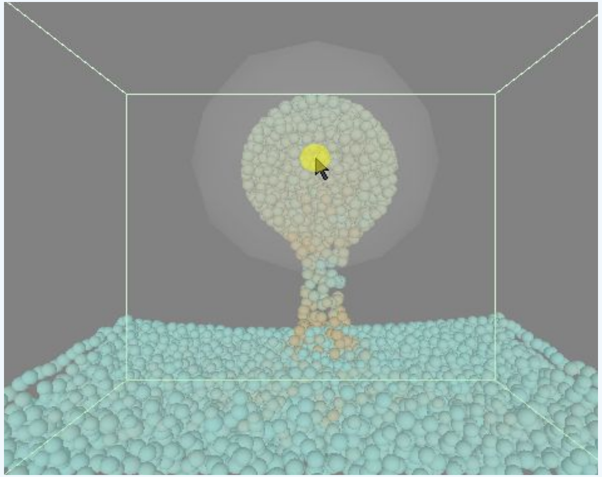


[2] Koschier, D., et al. (2019)

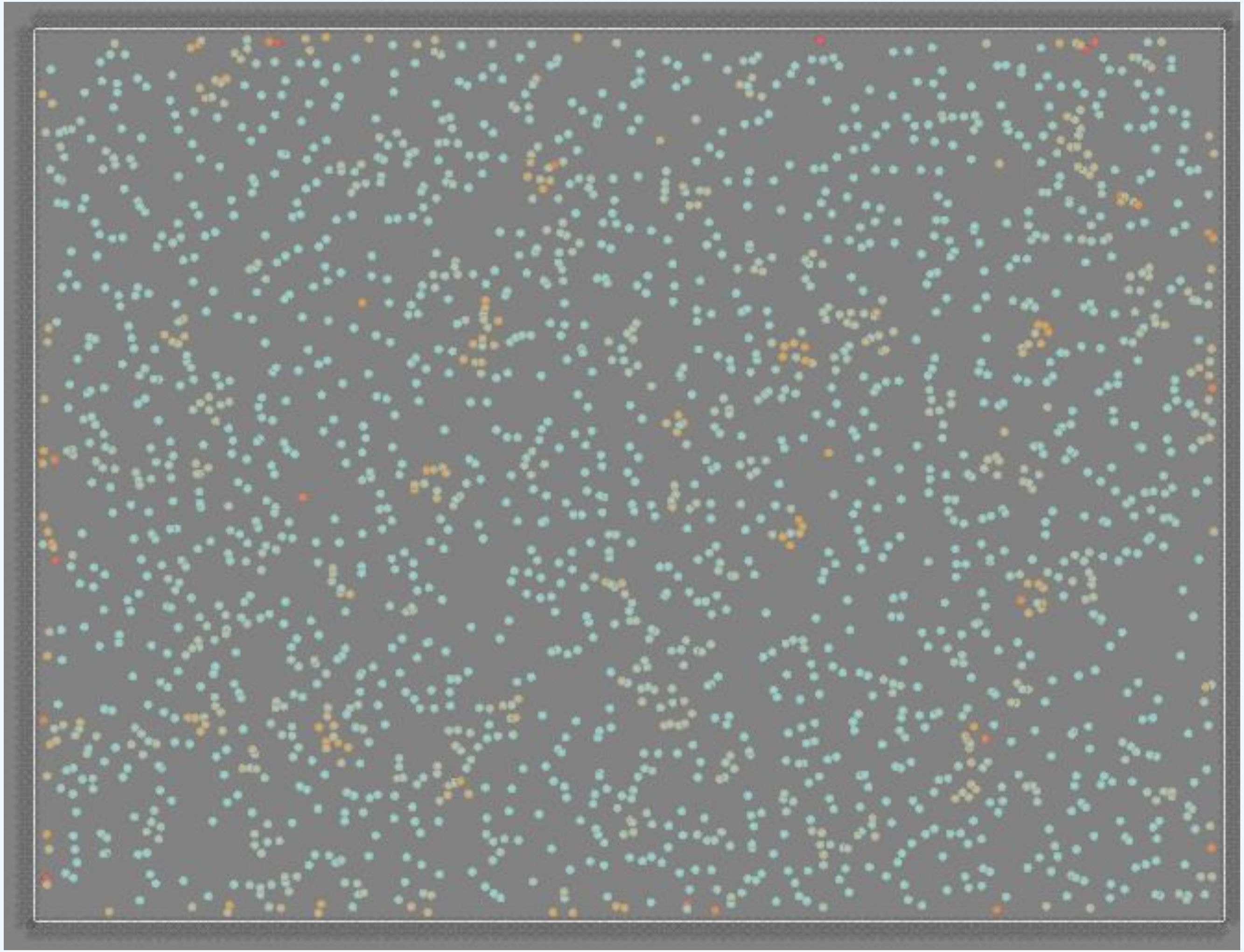
Mouse Interaction

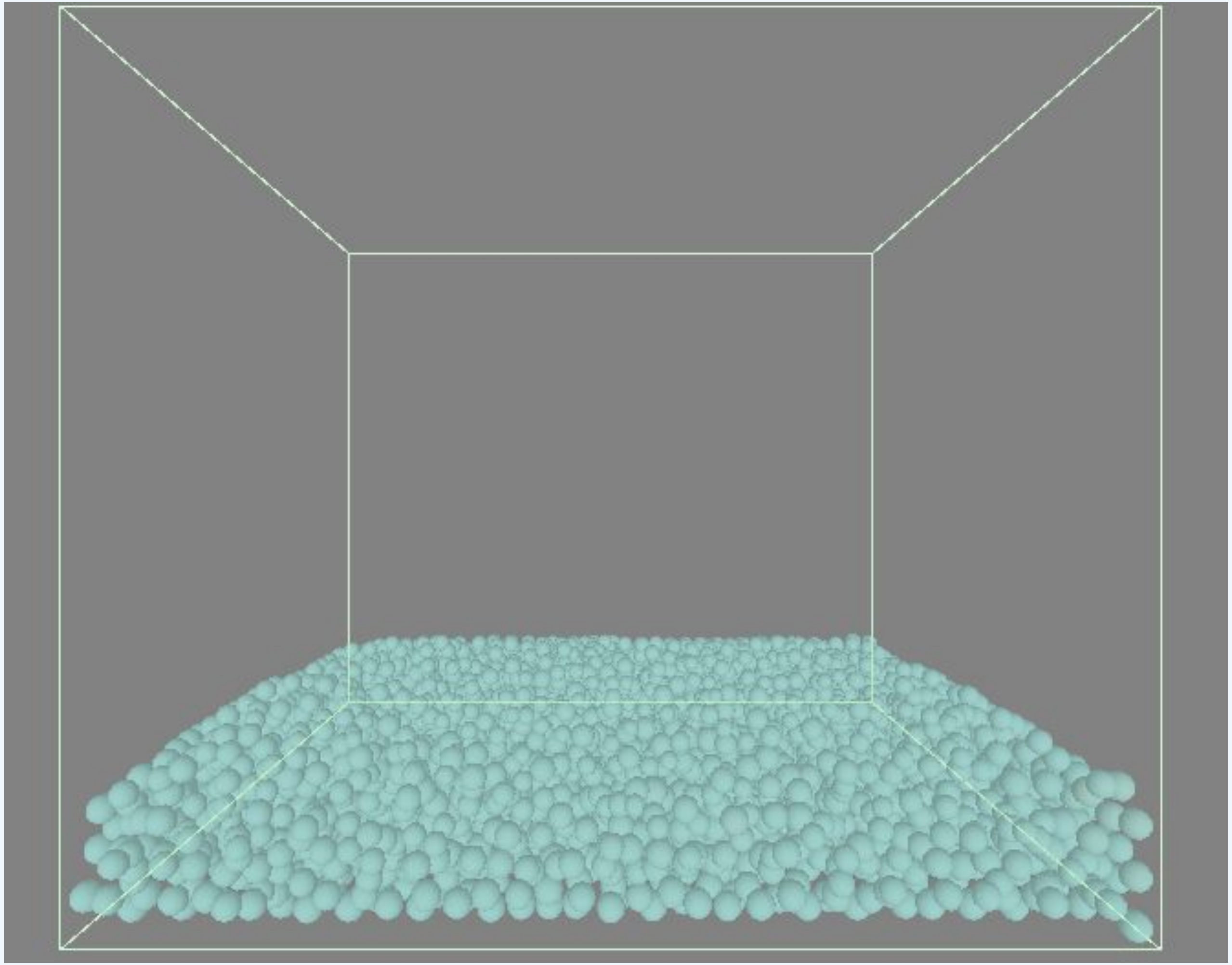
- Clicking creates an attractive force about a radius which allows for fluid manipulation in real-time
- In 3D, a raycast is used to determine the position of the force within the boundary





Results Demo





References

[1] Pelfrey, B. (2009). Real-Time Physics 103: Fluid Simulation in Games.

[2] Koschier, D., et al. (2019). Smoothed Particle Hydrodynamics Techniques for the Physics Based Simulation of Fluids and Solids. Eurographics.

[3] Lauge, S. "Coding Adventure: Simulating Fluids." YouTube, uploaded by Sebastian Lague, 8 October 2023. https://www.youtube.com/watch?v=rSKMYc1CQHE

[4] "Smoothed-particle Hydrodynamics" https://en.wikipedia.org/wiki/Smoothed-particle_hydrodynamics