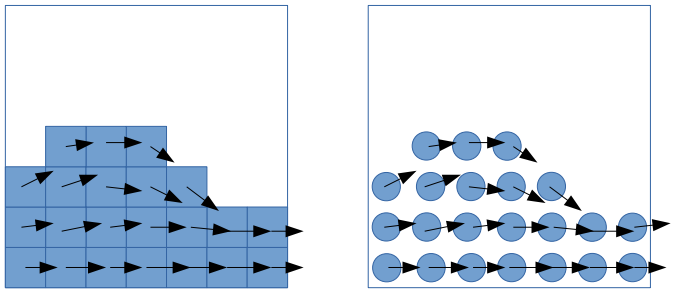
Chapter 14

Smoothed Particle Hydrodynamics (SPH)

In the beginning of this lecture, we shortly discussed the two points of view for modelling a fluid or, in general, a material: Lagrangian (observer moving with the fluid / material movement) or Eulerian (observer at fixed positions). When deriving the Navier-Stokes equations, we had decided that the Lagrangian approach was less suitable for fluid flow as the it would mess up our grid structure (intersecting grid lines, ill-formed elements, ...). However, these arguments hold if and only if we use a grid-based approach where we connect data points with grid lines and try to approximate differential operators with difference quotients. If we switch to a (virtual) particle-based approach with particles that have flexible positions (in contrast to Lattice-Boltzmann, where particle densities are assigned to grid positions), we can use a Lagrangian approach:



This is the basic idea of SPH: Each particle i has a mass, a position $\vec{x}_i$, and a velocity $\vec{v}_i$. Particles represent samples of continuous underlying scalar/vector fields (density, velocity, etc.), not actual fluid molecules. Each continuous function or field is represented as a linear combination of smooth kernel functions W centered at the particles' positions:

$$A(\vec{x}) \quad = \quad \sum_i A_i \frac{m_i}{\rho_i} W(\|\vec{x} - \vec{x}_i\|),$$

where $A(\vec{x})$ is the value at an arbitrary position in space, A_i the ith particle's value, $\frac{m_i}{\rho_i}$ the (virtual) particle volume, and W usually a Gaussian kernel, i.e.,

$$W(\|\vec{x} - \vec{x}_i\|) \quad = \quad e^{-\|\vec{x}-\vec{x}_i\|^2/\sigma^2}$$

with some $\sigma > 0$.

A time step of the SPH method executes the following steps:

1. compute the force acting on each particle,
2. update velocities and positions.

For the first step, we need to compute forces $\vec{f}_i$ acting on the particles. The second step requires a specialized time stepping method that ensures energy conservation in the overall system. In the following, we shortly sketch both steps:

14.1 Force Calculation

Forces acting on our virtual particles comprise the following 'components': gravity, pressure, and viscosity.

Gravity Gravity acting on a particle with density ρ_i is easy to describe with known physical laws as $\rho_i g$ with $g \approx 9,801 \frac{m}{s^2}$ representing the gravity acceleration on earth.

Pressure The pressure is the physical quantity that resists compression and volume change. The resulting force is (same as in our Navier-Stokes equations)

$$\vec{f}^p \quad = \quad -\nabla p.$$

In SPH, we'll assume that the pressure is proportional to the density:

$$p \quad = \quad k\left(\left(\frac{\rho}{\rho_0}\right)^7 - 1\right),$$

where k is the so-called gas constant, and ρ_0 denotes the rest density (at zero pressure).

Viscosity Similar as the influence of the pressure, we also know the influence of viscosity from our considerations for the Navier-Stokes equations. Viscosity describes internal friction or, in other words, the influence of mutual motion of fluid molecules:

$$\vec{f}^{vis} \quad = \quad \mu\Delta\vec{v},$$

where μ is the viscosity coefficient of the given fluid.

With this, we can calculate the total forces acting on particle i as

$$\vec{f}_i = \underbrace{\rho_i g}_{\text{gravity}} \underbrace{-\nabla p(\vec{x}_i)}_{\text{pressure}} + \underbrace{\mu \Delta \vec{v}(\vec{x}_i)}_{\text{viscosity}}.$$

In contrast to the discretization of our Navier-Stokes equations, evaluating derivatives in SPH is very simple, we only have to (analytically) compute the derivatives of the kernel function W:

$$\begin{aligned}
\nabla A(\vec{x}) &= \nabla \sum_i A_i \frac{m_i}{\rho_i} W(\|\vec{x} - \vec{x}_i\|) \\
&= \sum_i A_i \frac{m_i}{\rho_i} \nabla W(\|\vec{x} - \vec{x}_i\|), \\
\Delta A(\vec{x}) &= \Delta \sum_i A_i \frac{m_i}{\rho_i} W(\|\vec{x} - \vec{x}_i\|) \\
&= \sum_i A_i \frac{m_i}{\rho_i} \Delta W(\|\vec{x} - \vec{x}_i\|).
\end{aligned}$$

However, there is one physical law that we did not take into account yet: Newton's third law, which tells us that foces between two particles should be equal (in the absolute value) and opposite (in the direction). For pressure and viscosity forces, we have according to our derivations above:

$$\vec{f}_i^P = -\sum_j p_j \frac{m_j}{\rho_j} \nabla_{\vec{x}_i} W(\|\vec{x}_i - \vec{x}_j\|), \tag{14.1}$$

$$\vec{f}_i^{vis} = \mu \sum_j \vec{v}_j \frac{m_j}{\rho_j} \Delta_{\vec{x}_i} W(\|\vec{x}_i - \vec{x}_j\|). \tag{14.2}$$

We isolate the forces exerted from particle j on particle i

$$\begin{aligned}
\vec{f}_{i,j}^p &= -p_j \frac{m_j}{\rho_j} \nabla_{\vec{x}_i} W(\|\vec{x}_i - \vec{x}_j\|), \\
\vec{f}_{i,j}^{vis} &= \mu \vec{v}_j \frac{m_j}{\rho_j} \Delta_{\vec{x}_i} W(\|\vec{x}_i - \vec{x}_j\|),
\end{aligned}$$

and compare them to the forces exterted from particle i on particle j. To do so, we use

$$\nabla_{\vec{x}_i} W(\|\vec{x}_i - \vec{x}_j\|) = W'(\|\vec{x}_i - \vec{x}_j\|) \frac{\vec{x}_i - \vec{x}_j}{\|\vec{x}_i - \vec{x}_j\|} = -W'(\|\vec{x}_i - \vec{x}_j\|) \frac{\vec{x}_j - \vec{x}_i}{\|\vec{x}_i - \vec{x}_j\|} = -\nabla_{\vec{x}_j} W(\|\vec{x}_j - \vec{x}_i\|),$$

and

$$\begin{aligned}
\Delta_{\vec{x}_i} W(\|\vec{x}_i - \vec{x}_j\|) &= W''(\|\vec{x}_j - \vec{x}_i\|) \frac{(\vec{x}_i - \vec{x}_j)^T (\vec{x}_i - \vec{x}_j)}{\|\vec{x}_i - \vec{x}_j\|^2} + W'(\|\vec{x}_i - \vec{x}_j\|) \left(\frac{1}{\|\vec{x}_i - \vec{x}_j\|} - \frac{(\vec{x}_i - \vec{x}_j)^T (\vec{x}_i - \vec{x}_j)}{\|\vec{x}_i - \vec{x}_j\|^3} \right) \\
&= W''(\|\vec{x}_j - \vec{x}_i\|) = \Delta_{\vec{x}_j} W(\|\vec{x}_i - \vec{x}_j\|)
\end{aligned}$$

and get

$$\begin{aligned}
\vec{f}_{j,i}^p &= p_i \frac{m_i}{\rho_i} \nabla_{\vec{x}_i} W(\|\vec{x}_i - \vec{x}_j\|), \\
\vec{f}_{i,j}^{vis} &= \mu \vec{v}_i \frac{m_i}{\rho_i} \Delta_{\vec{x}_i} W(\|\vec{x}_i - \vec{x}_j\|).
\end{aligned}$$

We fulfil Newton's third law by averaging between the two version, i.e.,

$$\begin{aligned}\vec{f}_i^p &= -\sum_j \frac{1}{2}\left(p_i\frac{m_i}{\rho_i} + p_j\frac{m_j}{\rho_j}\right)\nabla_{\vec{x}_i} W(\|\vec{x}_i - \vec{x}_j\|),\\ \vec{f}_i^{vis} &= \sum_j \frac{\mu}{2}\left(\vec{v}_j\frac{m_j}{\rho_j} - \vec{v}_i\frac{m_i}{\rho_i}\right)\Delta_{\vec{x}_i} W(\|\vec{x}_i\vec{x}_j\|).\end{aligned}$$

Note that these force calculations are very expensive as, for each particle, we have to sum up contributions from the interaction with each other particle, resulting in an $\mathcal{O}(N^2)$ complexity if N is the total number of particles. We can borrow algorithms from moceluar dynamics to reduce this to $\mathcal{O}(N)$ by suitable approximations and algorithms. We don't go into detail here but refer to the respective literature.

14.2 Time Stepping

After knowing how to calculate the forces, we have to numerically solve the ODE

$$\ddot{\vec{x}}_i = \frac{\vec{f}_i}{m_i}$$

for each particle. To prepare this equation for time stepping, we first convert it to a system of first order ODEs:

$$\begin{aligned}\dot{\vec{x}}_i &= \vec{v}_i,\\ \dot{\vec{v}}_i &= \frac{\vec{f}_i}{m_i}.\end{aligned}$$

For such particle systems, energy conservation (sum of potential and kinetic energy) is crucial. Therefore, all our standard time stepping approchaes (Euler, Heun, Runge-Kutta & Co) are not suitable as they either generate or destroy energy. Methods conserving energy are called symplectic methods, a very simple and efficient one is the symplectic Euler that is formally explicit for the velocities and implicit for the positions:

$$\begin{aligned}\vec{v}_i^{(n+1)} &= \vec{v}_i^{(n)} + dt\cdot\frac{\vec{f}_i(\vec{x}^{(n)})}{m_i},\\ \vec{x}_i^{(n+1)} &= \vec{x}_i^{(n)} + dt\cdot\vec{v}_i^{(n+1)}.\end{aligned}$$

This symplectic Euler scheme is first order accuracte.

14.3 Summary of the Time Step Algorithm

Summarizing, the algorithm per time step comprises the following steps, using known values for $\vec{x}_i^{(n-1)}$ and $\vec{x}_i^{(n)}$.

1. Calculate the density

$$\rho_i^{(n)} = \rho(\vec{x}_i^{(n)}) = \sum_j m_j W(\|\vec{x}_i^{(n)} - \vec{x}_j^{(n)}\|),$$

2. Calculate the pressure coefficients

$$p_i^{(n)} = k\left(\left(\frac{\rho(\vec{x}_i^{(n)})}{\rho_0}\right)^7 - 1\right),$$

3. calculate the forces per particle from

$$\vec{f}_i^{(n)} = \rho_i^{(n)} g - \sum_j \frac{1}{2}\left(p_i^{(n)}\frac{m_i}{\rho_i^{(n)}} + p_j^{(n)}\frac{m_j}{\rho_j^{(n)}}\right)\nabla W(\|\vec{x}_i^{(n)} - \vec{x}_j^{(n)}\|) + \sum_j \frac{\mu}{2}\left(\vec{v}_j^{(n)}\frac{m_j}{\rho_j^{(n)}} - \vec{v}_i^{(n)}\frac{m_i}{\rho_i^{(n)}}\right)\Delta W(\|\vec{x}_i^{(n)} - \vec{x}_j^{(n)}\|),$$

4. calculate the new velocities as

$$\vec{v}_i^{(n+1)} = \vec{v}_i^{(n)} + dt\frac{\vec{f}_i^{(n)}}{m_i},$$

5. calculate the new positions as

$$\vec{x}_i^{(n+1)} = \vec{x}_i^{(n)} + dt\,\vec{v}_i^{(n+1)}.$$

6. calculate the velocity field

$$\vec{v}(\vec{x}_i) = \sum_j \vec{v}_j \frac{m_j}{\rho_j} W(\vec{x}_i - \vec{x}_j)$$

   and write its values on, e.g., a regular grid, to an output file if required (for visualization etc.).

We see that the SPH method is rather easy to implement, but also has several degrees of freedom that haven to be chosen a priori and are not directly derivable from physical laws: (i) particle masses and densities, (ii) kernel 'width' σ, and even the viscosity μ (depends on physical viscosity, but also the choices for W, ρ_i and m_i). Thus, SPH is challenging if we aim for physical correctness. On the other hand, producing beautiful images (e.g., for computer animation using isosurfaces of the density field $\rho(\vec{x})$) is easy and cheap, the same as simulations with free surfaces or transitions between solid and fluid states.