

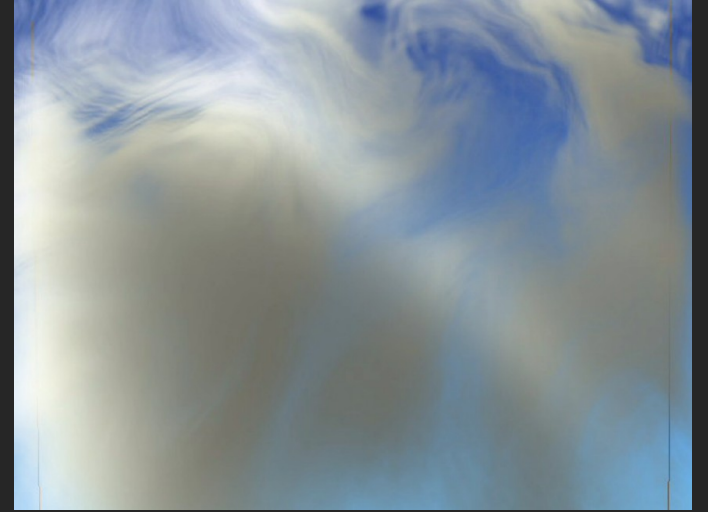
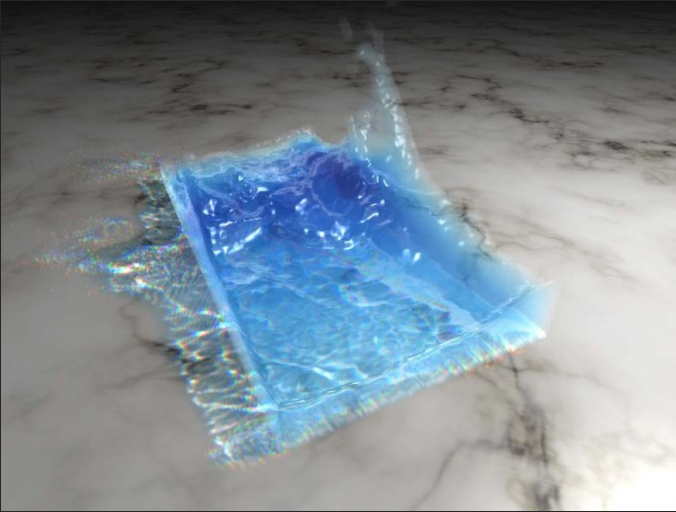
Fluid simulation

Jiří Chmelík, [Marek Trtík](#)

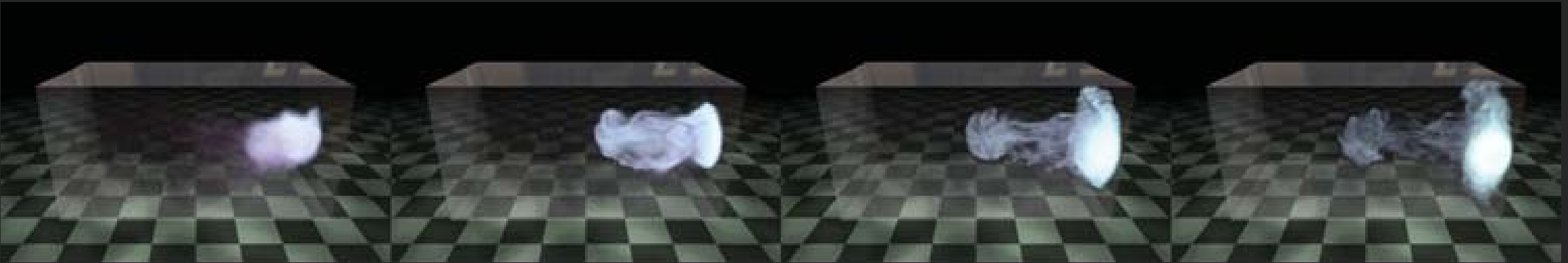
PA199

Motivation

Picture source: [2]



Picture source: [3]



Other pictures source: [5]

Outline

- ▶ **Euler approach:**

- ▶ Fluid is modelled by a vector field, representing the velocity of the fluid.

- ▶ **Lagrange approach:**

- ▶ Fluid is modelled by set of particles.

- ▶ **Smoothed Particle Hydrodynamics:**

- ▶ Fluid is modelled by set of particles moved via a velocity vector field.

- ▶ **Height-field surface approximation:**

- ▶ Suitable for simulation of only fluid's surface, e.g., lake or ocean surface.

Euler approach

Fluid Model

- ▶ Assumptions:
 - ▶ Incompressible fluid:
 - ▶ Volume of any subregion of the fluid is constant over time.
 - ▶ Represented by an incompressible constraint.
 - ▶ Homogeneous fluid:
 - ▶ The density of fluid is the same and constant in every region of the fluid and over time.
- ▶ Navier-Stokes equations model a fluid:
 - ▶ Fluid velocity (motion) represented by a vector field $u(x, t)$.
 - ▶ Fluid pressure represented by a scalar field $p(x, t)$.
 - ▶ Partial differential equations define changes in the vector field u over time.

Navier-Stokes Equations

- ▶ The momentum equations (for each coordinate one):

$$\frac{\partial u}{\partial t} = - \underbrace{(u \cdot \nabla)u}_{\text{advection}} - \frac{1}{\rho} \nabla p + \underbrace{\nu \nabla^2 u}_{\text{viscosity}} + \underbrace{g}_{\text{gravity}}$$

- ▶ The incompressibility constraint:

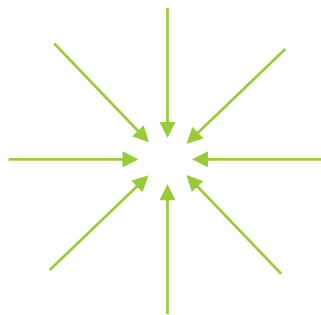
$$\nabla \cdot u = 0$$

- ▶ Where (let $x = (x, y, z)^T$ be a position in space and t be simulation time):

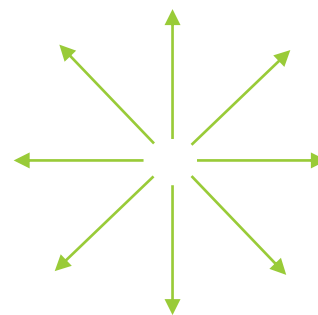
- ▶ $u(x, t) = (u(x, t), v(x, t), w(x, t))^T$ is the velocity vector field of the fluid. **(computed)**
- ▶ $p(x, t)$ is a pressure scalar field of the fluid; used to preserve incompressibility. **(computed)**
- ▶ ρ is the density of the fluid, e.g., water $\rho = 10^3 \frac{\text{kg}}{\text{m}^3}$
- ▶ ν is the viscosity (resistance to deformation) of the fluid, e.g., honey - high viscosity, water - low viscosity.
- ▶ $g(x, t)$ is the acceleration vector field of forces acting on the fluid, e.g., gravity $g(x, t) = (0, 0, -10)^T \frac{\text{m}}{\text{s}^2}$
- ▶ \cdot is the dot product.

Gradient, Divergence and Laplacian

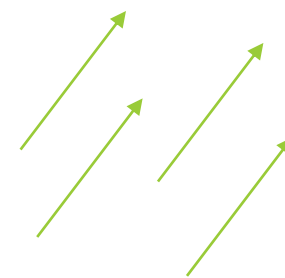
- ▶ Operator of total partial derivatives: $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)$
 - ▶ identifies a direction of a maximum increase of a function at a given time.
 - ▶ Example: $\nabla p = \left(\frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}, \frac{\partial p}{\partial z} \right)$
- ▶ Divergence operator: $\nabla \cdot \mathbf{u} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right) \cdot (u, v, w) = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}$
 - ▶ Can only be applied to a vector field.



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



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



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

Gradient, Divergence and Laplacian

► Directional derivative: $\mathbf{u} \cdot \nabla = (u, v, w)^T \cdot \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right) = u \frac{\partial}{\partial x} + v \frac{\partial}{\partial y} + w \frac{\partial}{\partial z}$

► Therefore, $(\mathbf{u} \cdot \nabla) \mathbf{u} = \left(u \frac{\partial}{\partial x} + v \frac{\partial}{\partial y} + w \frac{\partial}{\partial z} \right) (u, v, w)^T = \begin{pmatrix} u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \\ u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \\ u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \end{pmatrix}$

► Laplacian operator: $\nabla^2 = \nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$

► Example: $\nabla^2 \mathbf{u} = \nabla \cdot \nabla \mathbf{u} = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) (u, v, w)^T = \begin{pmatrix} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \\ \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \\ \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \end{pmatrix}$

Adding Custom Quantities

▶ Beside the fluid we also simulate other quantities, e.g., smoke density, temperature.

▶ Represent any such quantity q as another scalar/vector field.

▶ Add a related equation, how q changes in time:

$$\frac{\partial q}{\partial t} = -(\mathbf{u} \cdot \nabla)q + \nu \nabla^2 q + S$$

▶ Observe the similarity with the momentum equation:

▶ Advection: $-(\mathbf{u} \cdot \nabla)q$

▶ Diffusion: $\nu \nabla^2 q$

▶ We do not have pressure term.

▶ S can be used to simulate constant inflow of q into the fluid.

⇒ Methods for solving the momentum equation can be also applied for q equation.

Boundary Conditions

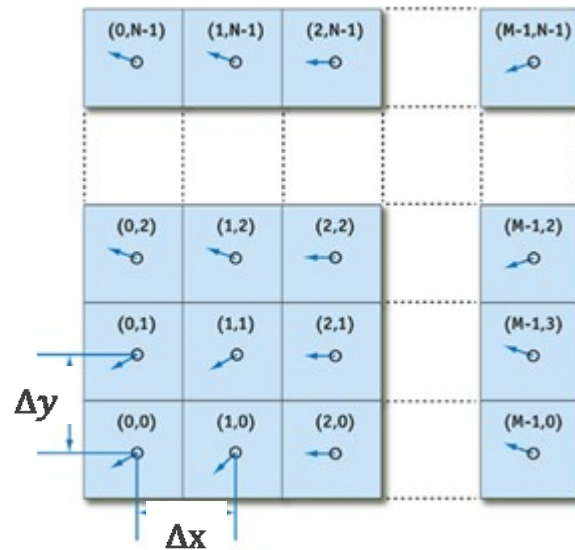
- ▶ The fluid can collide with:
 - ▶ Static solid objects, like walls.
 - ▶ Freely moveable solid objects, like piece of wood in water.
 - ▶ Another fluid, like oil stain on water surface. (not covered in this lecture)
- ▶ Our goal is to prevent the fluid to flow into the solid objects.
- ▶ Let $n(x, t), u_s(x, t)$ be the normal and velocity of the solid surface.
- ▶ The boundary constraint for:
 - ▶ Low viscosity fluid: $u(x, t) \cdot n(x, t) = u_s(x, t) \cdot n(x, t)$
 - ▶ High viscosity fluid: $u(x, t) = u_s(x, t)$
- ▶ We can use boundary condition to model fluid source and/or sink.

Discretize fields

- ▶ Discretize the space into a regular grid.

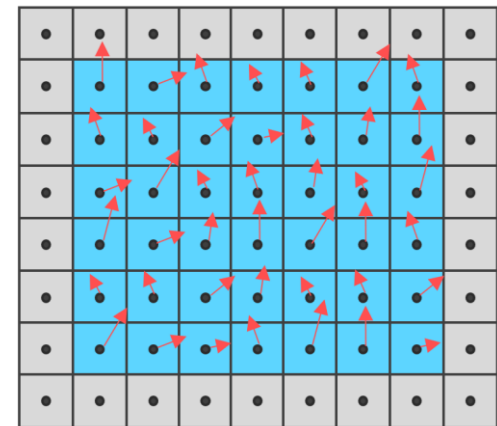
for each cell (i, j) , we store:

- ▶ Fluid velocity: $u_{i,j}$
- ▶ Pressure: $p_{i,j}$
- ▶ Any other field: $q_{i,j}$



- ▶ Discretize boundary conditions:

- ▶ Mark cells filled by solid objects, e.g., walls.



Discretize derivatives

- ▶
- ▶ Use finite differences to approximate partial derivatives.
- ▶ Examples:

$$\nabla \phi_{ijk} = \left(\frac{\phi_{i+1,j,k} - \phi_{i-1,j,k}}{2\Delta x}, \frac{\phi_{i,j+1,k} - \phi_{i,j-1,k}}{2\Delta y}, \frac{\phi_{i,j,k+1} - \phi_{i,j,k-1}}{2\Delta z} \right)$$

$$\nabla \cdot \mathbf{u}_{ijk} = \frac{u_{i+1,j,k} - u_{i-1,j,k}}{2\Delta x} + \frac{v_{i,j+1,k} - v_{i,j-1,k}}{2\Delta y} + \frac{w_{i,j,k+1} - w_{i,j,k-1}}{2\Delta z}$$

$$\nabla^2 \phi_{ijk} = \frac{\phi_{i+1,j,k} - 2\phi_{i,j,k} + \phi_{i-1,j,k}}{\Delta x^2} + \frac{\phi_{i,j+1,k} - 2\phi_{i,j,k} + \phi_{i,j-1,k}}{\Delta y^2} + \frac{\phi_{i,j,k+1} - 2\phi_{i,j,k} + \phi_{i,j,k-1}}{\Delta z^2}$$

Solving Equations

► Method of splitting:

- Solve a complex equation by a sequence numerical integrations.

$$\frac{dq}{dt} = f(q) + g(q) \quad \rightarrow \quad \begin{aligned} \tilde{q} &= q^t + \Delta t f(q^t) \\ q^{t+\Delta t} &= \tilde{q} + \Delta t g(\tilde{q}) \end{aligned}$$

- The result is equivalent to a single integration:

$$\begin{aligned} q^{t+\Delta t} &= \tilde{q} + \Delta t g(\tilde{q}) \\ &= q^t + \Delta t f(q^t) + \Delta t g(q^t + \Delta t f(q^t)) \\ &= q^t + \Delta t f(q^t) + \Delta t (g(q^t) + O(\Delta t)) \\ &= q^t + \Delta t (f(q^t) + g(q^t)) + O(\Delta t^2) \\ &= q^t + \Delta t \frac{dq}{dt} + O(\Delta t^2) \end{aligned}$$

Solving Equations

- ▶ We solve the momentum equation using the splitting method:

$$\frac{\partial u}{\partial t} = -(u-v)u - \frac{1}{\rho} \nabla p + \nu \nabla^2 u + g_x$$

- ▶ Start in the current state:

$$w_1(x) = u(x,t)$$

- ▶ Apply external accelerations g_x :

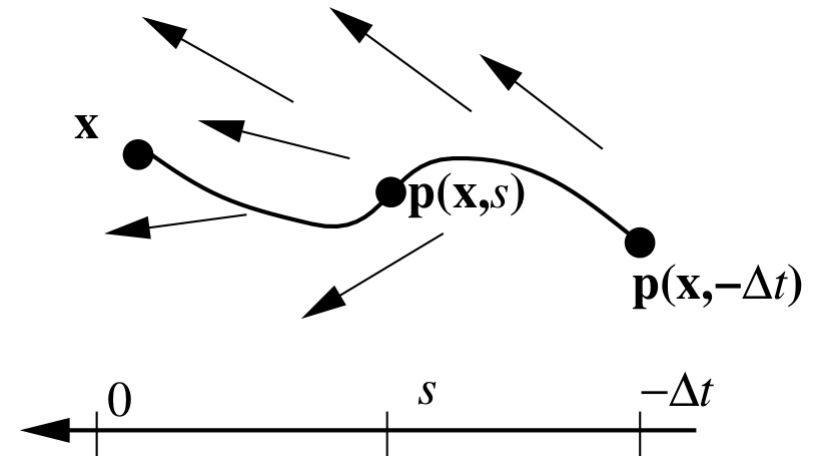
$$w_2(x) = w_1(x) + \Delta t g_x$$

[forward Euler]

- ▶ Apply fluid advection $-(u-v)u$:

$$w_2(x) = w_1(p(x, -\Delta t))$$

[method of characteristics]



The new velocity $w_2(x)$ is now the velocity at the location $p(x, -\Delta t)$ at the previous time step.

$w_2(x) = w_1(p(x, -\Delta t))$

Solving Equations

- ▶ We solve the momentum equation using the splitting method:

$$\frac{\partial u}{\partial t} = -(u \cdot \nabla)u - \frac{1}{\rho} \nabla p + \nu \nabla^2 u + g$$

- ▶ Apply fluid viscosity $\nu \nabla^2 u$

$$w_2(x) \leftarrow w_2(x) + \Delta t \nu \nabla^2 w_2(x)$$

(backward Euler)

- ▶ Lastly, we must compute the pressure $-\frac{1}{\rho} \nabla p$ acceleration s.t. we remove divergence from w_3 , i.e., to satisfy the incompressibility:

$$\nabla \cdot w_3 = 0$$

Solving Equations

- ▶ Helmholtz-Hodge Decomposition: Any vector field w can be uniquely decomposed to a vector field u and a scalar field p satisfying:

$$w = u + \nabla p$$

where u is a divergence free, i.e., $\nabla \cdot u = 0$.

- ▶ When we apply divergence operator to both sides of the equation:

$$\nabla \cdot w = \nabla^2 p$$

we get a Poisson equation.

- ▶ Due to discretization, we get a sparse system of linear equations

=> Use, for example, Jacobi method.

- ▶ We use the computed pressure field to get the resulting fluid velocity:

$$u(x, t + \Delta t) = w_3(x) - \nabla p$$

Euler approach

DEMO!

<https://paveldogreat.github.io/WebGL-Fluid-Simulation/>
<http://haxiomic.github.io/projects/webgl-fluid-and-particles/>

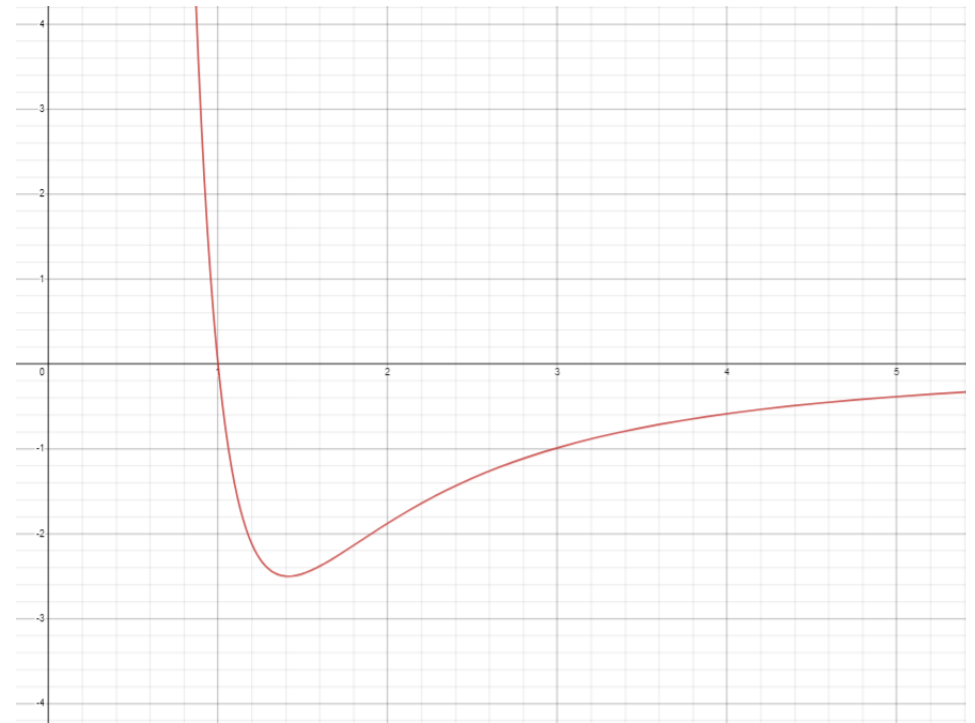
Lagrange approach

Particles Simulation

- ▶ The fluid is represented by n particles $\{P_0, \dots, P_{n-1}\}$.
- ▶ Each particle P_i is defined by:
 - ▶ Mass: m_i
 - ▶ Position vector: p_i
 - ▶ Velocity vector: w_i
 - ▶ Total external force: f_i
- ▶ Newton's equations of motion for moving particles:
 - ▶ $\frac{dp_i}{dt} = w_i$ (3 equations in 3D space)
 - ▶ $\frac{dw_i}{dt} = \frac{f_i}{m_i}$ (3 equations in 3D space)

External Forces

- ▶ The attribute f_i of a particle p_i is a sum of all forces acting on the particle.
- ▶ We usually want Farth's gravity to act on particles:
 - ▶ Force of a homogenous field: $m_i g$
 - ▶ Typically: $g = (0, 0, -10)^T$
- ▶ Interaction between particles P_i and P_j via Lennard-Jones forces:
 - ▶ Let $d_{ij} = |p_i - p_j|$ and $d_{ij} = \frac{r_0 - d_{ij}}{d_{ij}}$
 - ▶ $F_{ij} = \left(\frac{k_1}{d_{ij}^{12}} - \frac{k_2}{d_{ij}^6} \right) d_{ij}$, $F_{ji} = -F_{ij}$
 - ▶ where typically $k_1 = k_2$, $m = 4$ and $n = 2$.



Plotted at <https://www.desmos.com/calculator>

Lagrange approach

DEMO!

Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics

- ▶ Simulate fluid using a set of n particles, i.e., Lagrange approach.
- ▶ Compute forces acting on the particles by Euler approach. How?
- ▶ Smooth properties of particles into continuous fields.

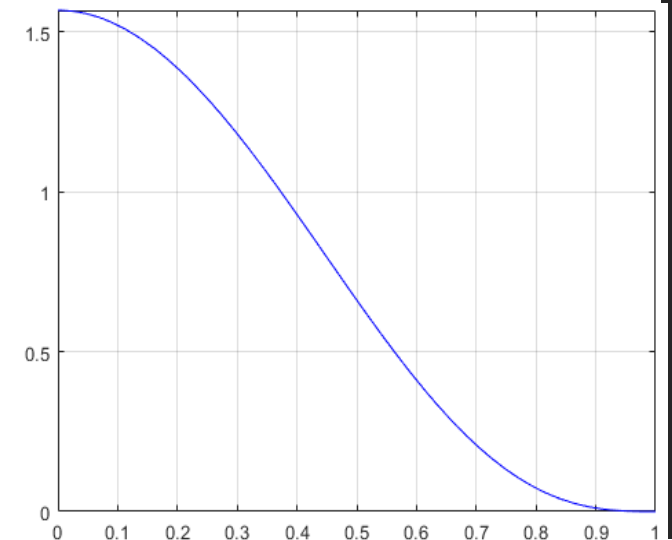
- ▶ Use a smoothing kernel $W(x)$, e.g., poly6:

$$W(x) = \frac{315}{64\pi d^3} \begin{cases} (d^2 - x^2)^3 & \text{if } 0 \leq x \leq d \\ 0 & \text{otherwise} \end{cases}$$

- ▶ Let A be a property of particle. Then continuous field $A(x)$ is:

$$A(x) = \sum_{j=0}^{N-1} m_j \frac{A_j}{\rho_j} W(|x_j - x|)$$

- ▶ Example: $\rho(x) = \sum_{j=0}^{N-1} m_j \frac{\rho_j}{\rho_j} W(|x_j - x|) = \sum_{j=0}^{N-1} m_j W(|x_j - x|)$.



Smoothed Particle Hydrodynamics

- ▶ With the fields defined we can use momentum and incompressibility equations:

$$\frac{\partial u}{\partial t} = -(\mathbf{u} \cdot \nabla)u - \frac{1}{\rho} \nabla p + \nu \nabla^2 u + g, \quad \nabla \cdot \mathbf{u} = 0.$$

- ▶ We simulate particles \Rightarrow mass is conserved $\Rightarrow \nabla \cdot \mathbf{u} = 0$ is not needed.
- ▶ Particles automatically move with the fluid $\Rightarrow -(\mathbf{u} \cdot \nabla)u$ is not needed.

- ▶ So, we only solve: $\frac{\partial u}{\partial t} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 u + g.$

- ▶ Recall second Newton's equation of motion: $\frac{d\mathbf{u}_i}{dt} = \frac{\mathbf{f}_i}{m_i}$

- ▶ Therefore, $\frac{\mathbf{f}_i}{m_i} = -\frac{1}{\rho(x_i)} \nabla p(x_i) + \nu \nabla^2 u(x_i) + g.$

Smoothed Particle Hydrodynamics

- ▶ The pressure field p can be obtained from density field ρ by law of ideal gas
 - ▶ $p(x) = k(\rho(x) - \rho_0)$, where k is a gas constant and ρ_0 is the environment pressure.
- ▶ Derivatives of any field $A(x)$:

$$\nabla A(x) = \sum_{j=0}^{n-1} m_j \frac{A_j}{P_j} \nabla W(x_j - x), \quad \nabla^2 A(x) = \sum_{j=0}^{n-1} m_j \frac{A_j}{P_j} \nabla^2 W(x_j - x)$$

where $\nabla W(x_j - x) = W'(|x_j - x|) \frac{x_j - x}{|x_j - x|}$, $\nabla^2 W(x_j - x) = W''(|x_j - x|) + \frac{2W'(|x_j - x|)}{|x_j - x|^3}$

- ▶ Forces between two particles generated by fields $\nabla p, \nabla^2 w$ should be symmetric => we usually modify their computation:

$$\nabla p(x) = \sum_{j=0}^{n-1} m_j \frac{P_j + P_i}{2P_j} \nabla W(|x_j - x_i|), \quad \nabla^2 u(x) = \sum_{j=0}^{n-1} m_j \frac{u_j}{P_j} \nabla^2 W(x_j - x_i)$$

Height-field surface approximation

Fluid Surface Model

- ▶ We model a fluid surface by a function $h(x, y, t)$.
 - ▶ At a point (x, y) in the XY plane and at time t the function defines fluid height $z = h(x, y, t)$.
- ▶ Change of h in time is given by:
$$\frac{\partial^2 h}{\partial t^2} = v^2 \nabla^2 h$$
where v is the speed of waves in the fluid.
- ▶ How to solve the equation?
 - ▶ Introduce an auxiliary function $q = \frac{\partial h}{\partial t}$.
 - ▶ Rewrite the equation into this system:
$$\frac{\partial q}{\partial t} = v^2 \nabla^2 h, \quad \frac{\partial h}{\partial t} = q.$$
 - ▶ Discretize (next slide).

Discretize Model

- ▶ We discretize the functions h, q by 2D arrays:

$$h(x_0 + i\Delta x, y_0 + j\Delta y, t_0 + k\Delta t) \Rightarrow h_{ij}^k$$

$$q(x_0 + i\Delta x, y_0 + j\Delta y, t_0 + k\Delta t) \Rightarrow q_{ij}^k$$

where

- ▶ i, j are indices to the arrays.
 - ▶ $\Delta x, \Delta y$ are distances between grid cells in X,Y axes.
 - ▶ k simulation step number.
 - ▶ Δt simulation time step.
 - ▶ NOTE: Usually, $x_0 = y_0 = t_0 = 0$.
- ▶ We solve the discretized system numerically, e.g., using forward Euler method:

$$q_{ij}^{k+1} = q_{ij}^k + \Delta t \nu^2 \left(\frac{h_{i+1,j}^k - 2h_{ij}^k + h_{i-1,j}^k}{\Delta x^2} + \frac{h_{i,j+1}^k - 2h_{ij}^k + h_{i,j-1}^k}{\Delta y^2} \right)$$

$$h_{ij}^{k+1} = h_{ij}^k + \Delta t c q_{ij}^{k+1}$$

Hight-field surface approximation

DEMO!

References

- [1] W.J. Laan, S. Green, M. Sainz; Screen Space Fluid Rendering with Curvature Flow; I3D 2009.
- [2] S. Green; Screen Space Fluid Rendering for Games; GDC 2010.
- [3] Jos Stam; Stable Fluids; ACM Transactions on Graphics, 2001.
- [4] R.Bridson, M.Müller; Fluid simulation; SIGGRAPH 2007 course notes.
- [5] GPU Gems 3; Chapter 30: Real-Time Simulation and Rendering of 3D Fluids.
- [6] GPU Gems; Chapter 38: Fast Fluid Dynamics Simulation on the GPU; https://developer.download.nvidia.com/books/HTML/gpugems/gpugems_ch38.html
- [7] C. Johanson; Real-time water rendering; Master thesis, Lund University, 2004.