SPH simulation

Oshiro laboratory, Division of Bioengineering

Graduate School of Engineering Science, Osaka University

Shunsuke Yoshimoto

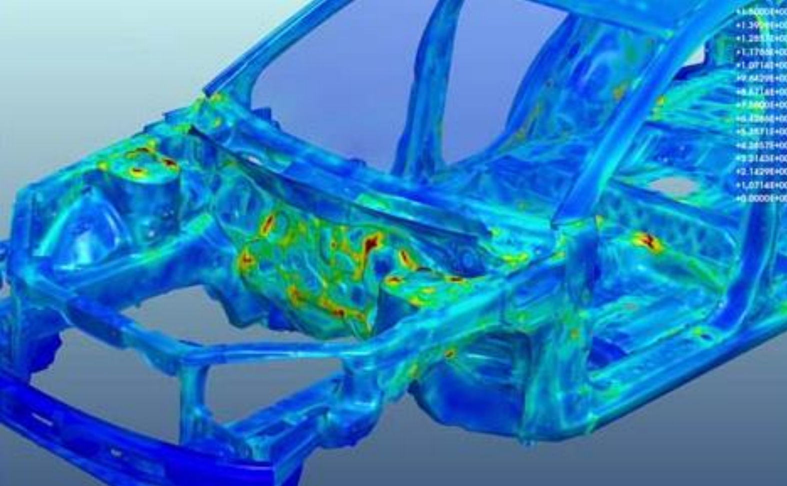
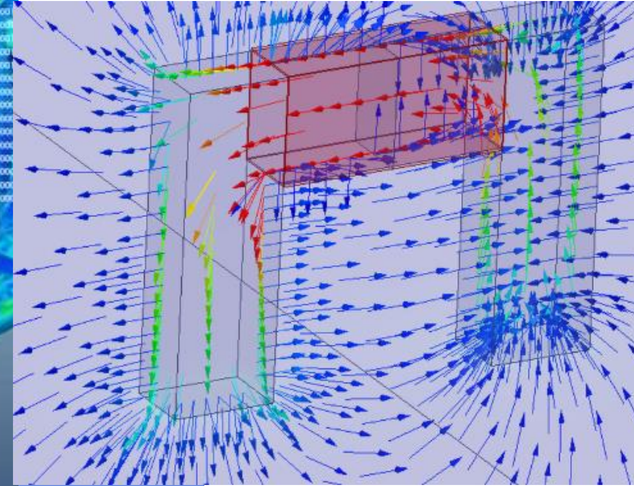
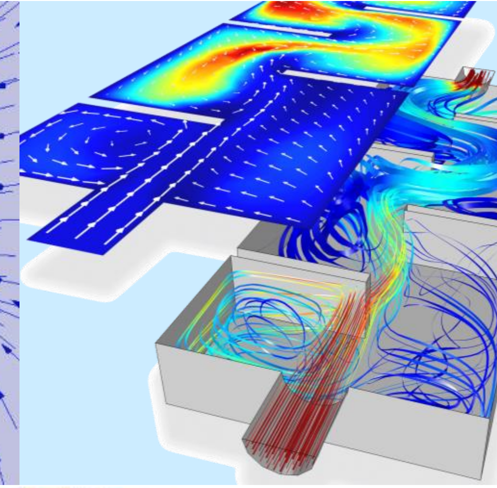
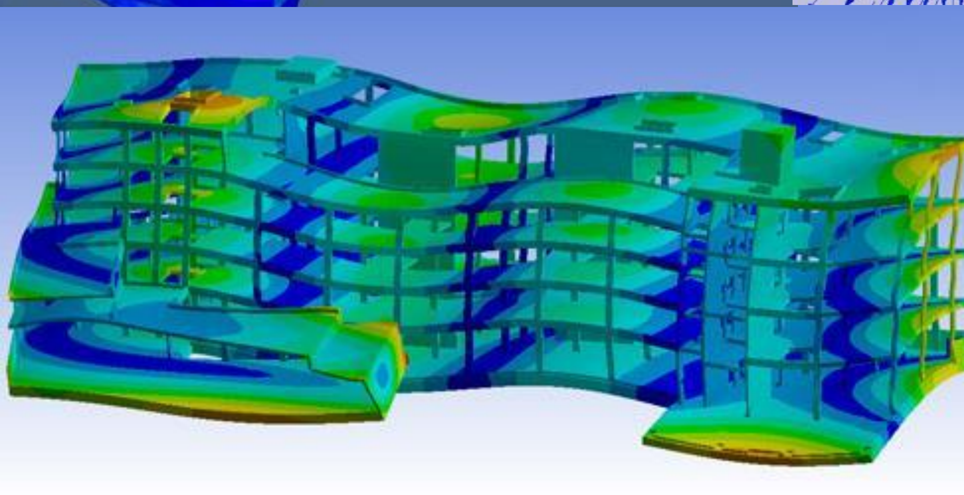
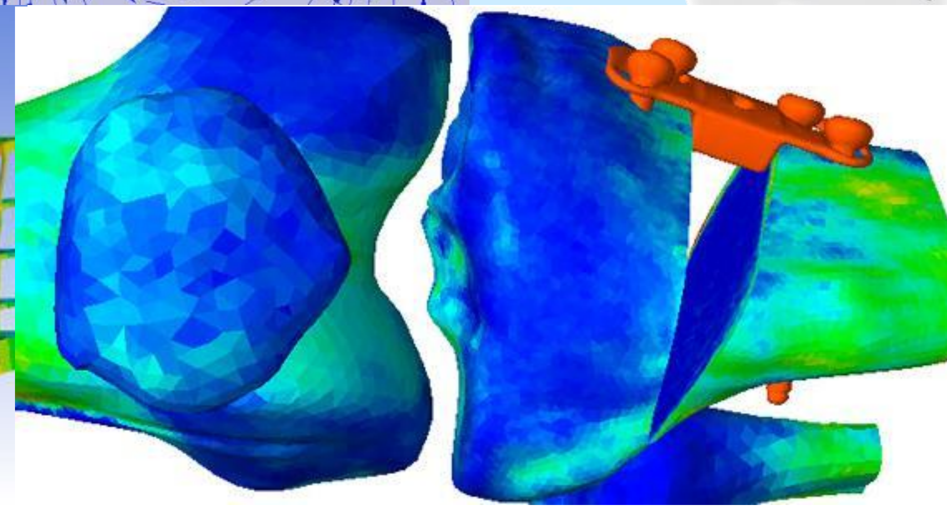
April 22, 2016

Physics based simulation

Demonstration

Physics based simulation

Fundamentals

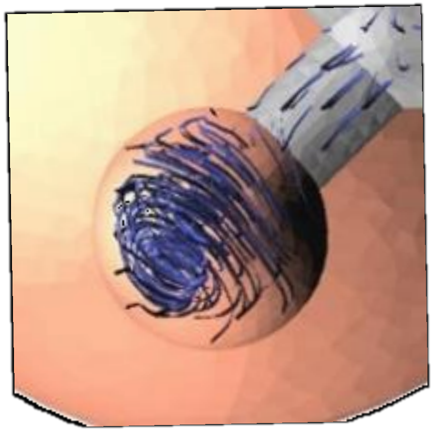
Physics based simulation is an effective technology for various fields.

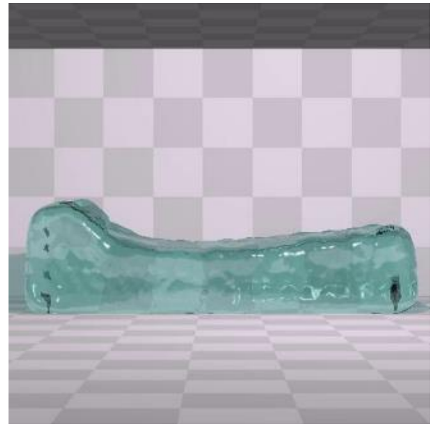
What is simulation?

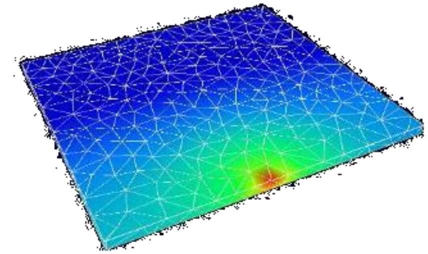
Meaning of simulation is imitation of real-world phenomenon.

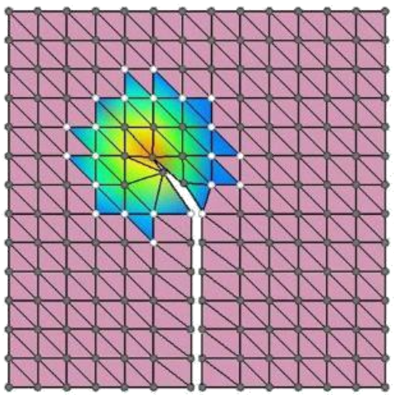
> *“Investigation of objective signal and profile by imitating real-world phenomenon under controllable condition to reveal the mechanism”* from WIKI

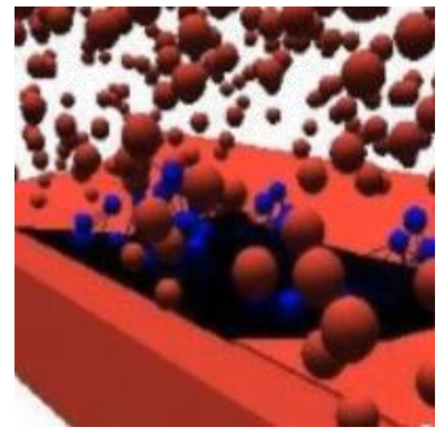
- Prediction of phenomenon for preparation
- Costless and convenient approach requiring no real resource
- Quantitative result is available.
- It is possible to Investigate the phenomenon of difficult to monitor.

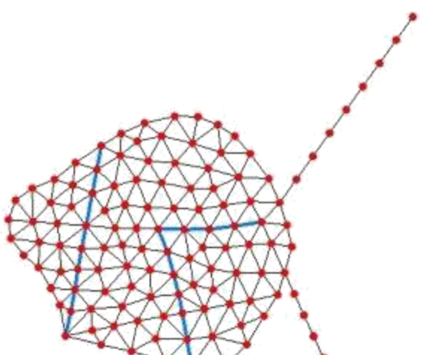


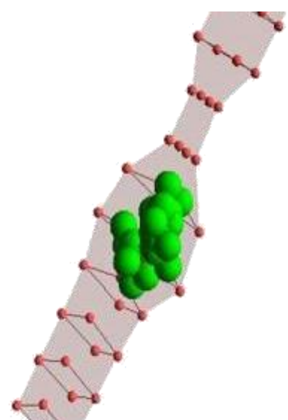


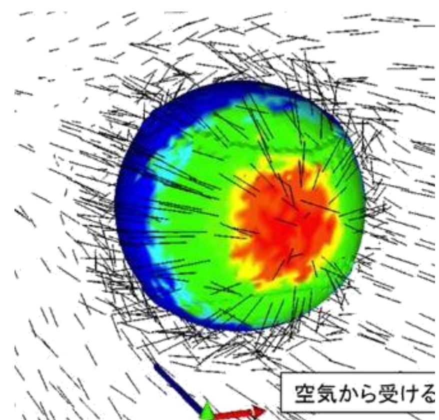











空気から受ける

Physical simulation: Obtaining numerical solution for governing equation

$$\frac{\partial \boldsymbol{v}}{\partial t} = -\frac{1}{\rho}\nabla p + \mu \nabla^2 \boldsymbol{v} + \boldsymbol{F}$$

$$\nabla \cdot \boldsymbol{B} = 0$$

$$C_v \frac{\partial T}{\partial t} = -\lambda \frac{\partial^2 T}{\partial x^2}$$

$$\nabla \times \boldsymbol{E} + \frac{\partial \boldsymbol{B}}{\partial t} = 0$$

$$\frac{1}{s^2}\frac{\partial^2 u}{\partial t^2} = \nabla^2 u$$

$$\nabla \cdot \boldsymbol{D} = \rho$$

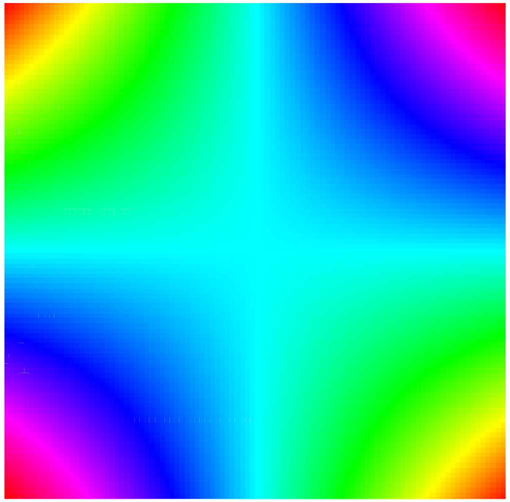
$$\boldsymbol{F} = m\frac{d^2\boldsymbol{x}}{dt^2} + c\frac{d\boldsymbol{x}}{dt} + k\boldsymbol{x}$$

$$\nabla \times \boldsymbol{H} - \frac{\partial \boldsymbol{D}}{\partial t} = j$$

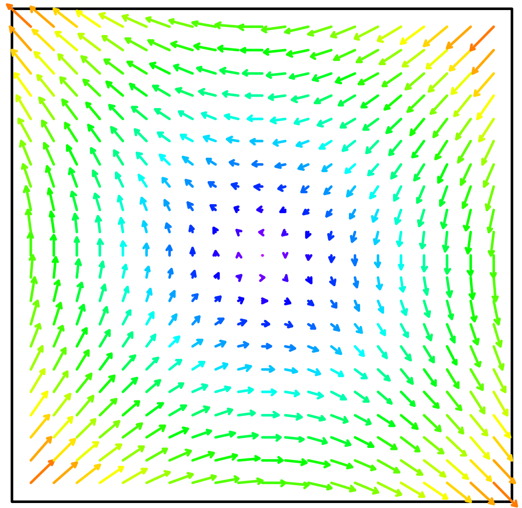
$$\frac{\partial \phi}{\partial t} = D\nabla^2 \phi$$

$$\int_v \{\epsilon\}^{\mathrm{T}}\{\sigma\}dv - \int_v \{U\}^{\mathrm{T}}\{\bar{G}\}dv - \int_{S_\sigma} \{U\}^{\mathrm{T}}\{\bar{T}\}ds = 0$$

Scalar field, vector field and derivative operator



Scalar field



Vector field

$$\nabla = \left\{ \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right\}$$

Spatial derivative

$$\nabla f = \left\{ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right\} = \mathrm{grad} f$$

$$\nabla \times \boldsymbol{v} = \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ v_x & v_y & v_z \end{vmatrix} = \mathrm{rot} \boldsymbol{v}$$

$$\nabla \cdot \boldsymbol{v} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} = \mathrm{div} \boldsymbol{v}$$

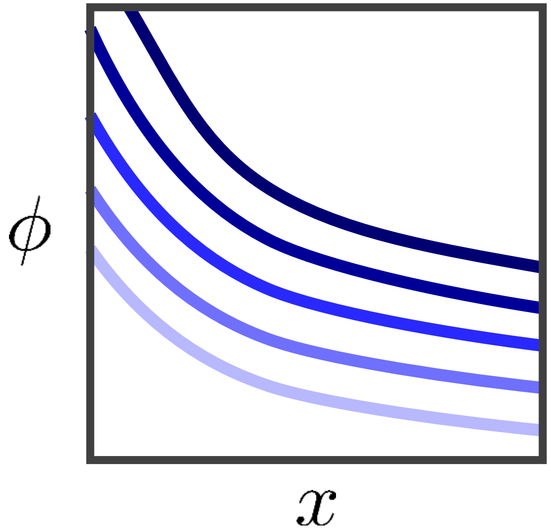
$$\Delta f = \nabla^2 f$$

Solution with initial and boundary conditions

Without boundary condition

$$\frac{d\phi(x)}{dx} = -\phi(x)$$

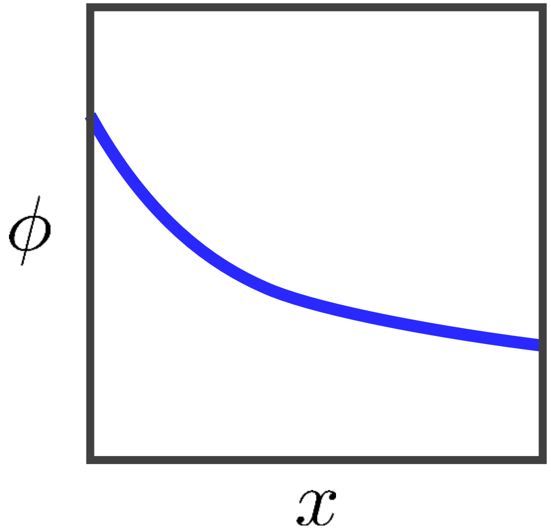
$$\phi(x) = A\exp(-x)$$



With boundary condition

$$\frac{d\phi(x)}{dx} = -\phi(x) \qquad \boxed{\phi(0) = 1}$$

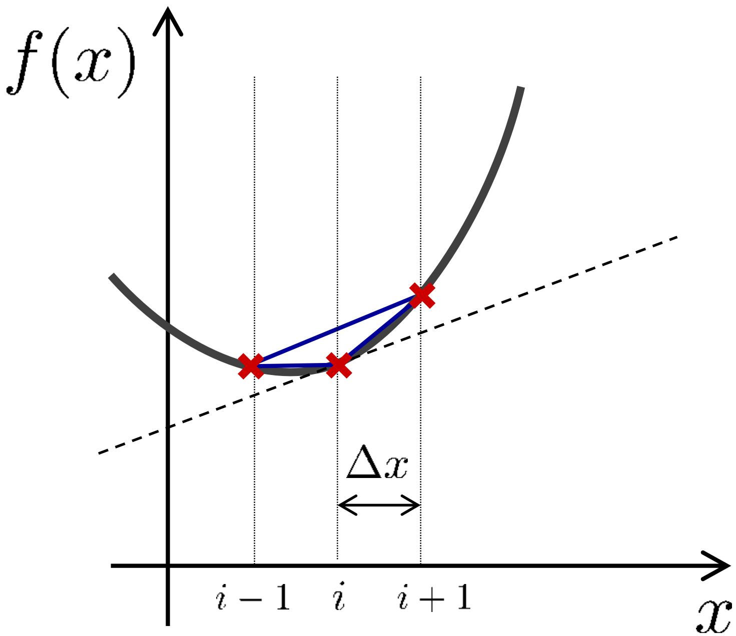
$$\phi(x) = \exp(-x)$$



Discretization for calculation in a computer

$$f(x) \xrightarrow{\text{discretization}} f[i]$$

$$\frac{df(x)}{dx} \xrightarrow{\text{discretization}}$$



Forward difference $\dfrac{f[i+1]-f[i]}{\Delta x}$

Central difference $\dfrac{f[i+1]-f[i-1]}{2\Delta x}$

Backward difference $\dfrac{f[i]-f[i-1]}{\Delta x}$

Keywords in numerical analysis

- Error, Accuracy, Stability
- Newton method, Bisection method
- Gaussian elimination, iterative method
- Polynomial interpolation, Least-squares method
- Quadrature by parts, Trapezoidal rule, Simpson's rule
- Euler method, Crank-Nicholson scheme, Runge-Kutta method
- Method of Lagrange multiplier

Fluid dynamics and numerical calculation

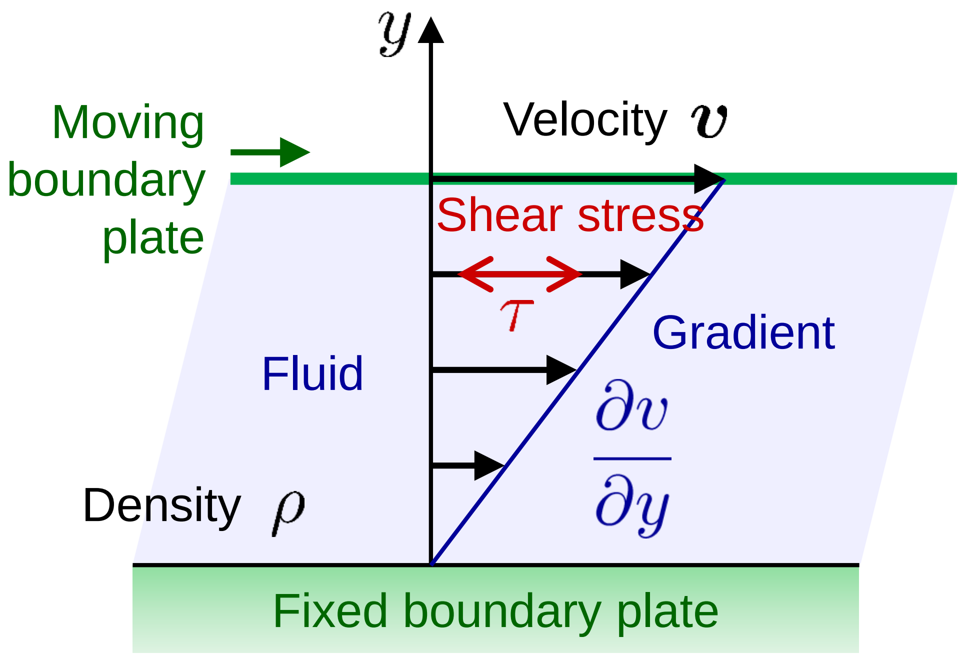
Warming-up

State of flow

- A flow in which the velocity of the fluid at a particular fixed point
  - does not change with time: Steady flow
  - changes with time: Unsteady flow
- Properties
  - compressible / incompressible
  - viscid / inviscid

Viscosity

: Resistance to gradual deformation by shear stress



$$\tau = \mu \frac{\partial v}{\partial y}$$

$$\nu = \frac{\mu}{\rho}$$

– Dynamic (shear) viscosity μ

– Kinematic viscosity ν

Types of fluid

According to shear velocity, the dynamic viscosity of the fluid

- will not change: Newtonian fluid
  - water, air, silicon oil, alcoholic
- will change: Non-Newtonian fluid
  - Pseudo plastic fluid
    - Paint, polymer solution, mayonnaise and chocolate
  - Binbham plastic fluid
    - Pigment, protein solution and cream
  - Dilatant fluid
    - Slurry of starch and water and sand containing water

Two equations of fluid dynamics

- Equation of continuity

  (law of conservation of mass)

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0$$

t : Time ρ : Density $\boldsymbol{v}$: Velocity

- Navier–Stokes equations

  (law of conservation of momentum)

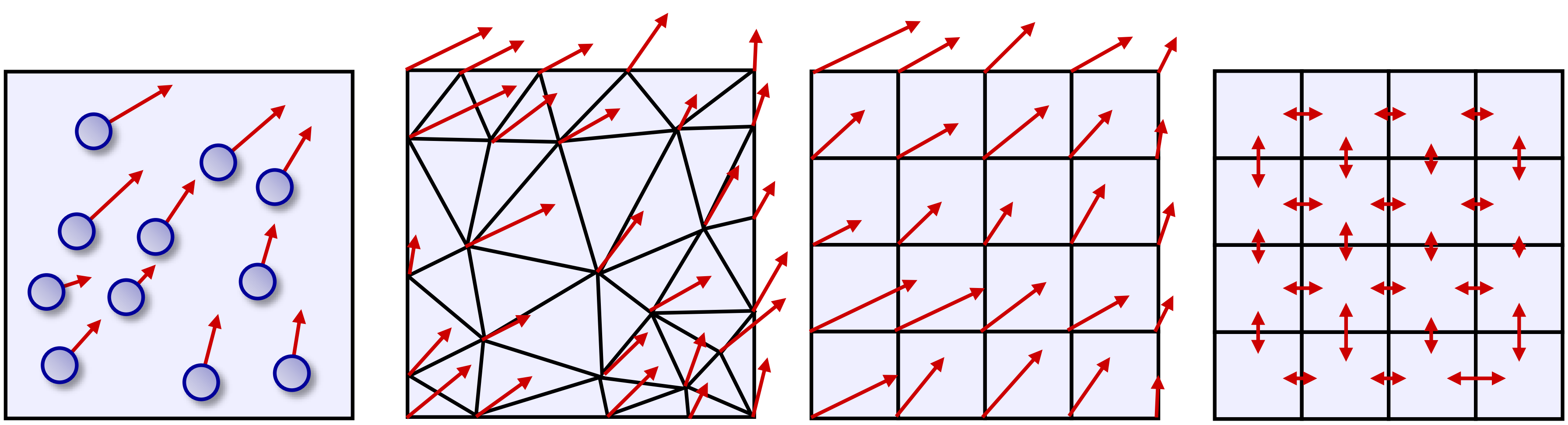
$$\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{v} = -\frac{1}{\rho}\nabla p + \mu \nabla^2 \boldsymbol{v} + \boldsymbol{f}_{\mathrm{ext}}$$

p : Pressure μ : Kinematic viscosity $\boldsymbol{f}_{\mathrm{ext}}$: External force

Methodology of computational fluid dynamics

- Particle method: Fluid is represented as a set of particles.
  - Smoothed Particle Hydrodynamics, Moving Particle Semi-implicit
- Mesh method: The space is represented a mesh.
  - Finite element method, Finite volume method,

    and Finite difference methods

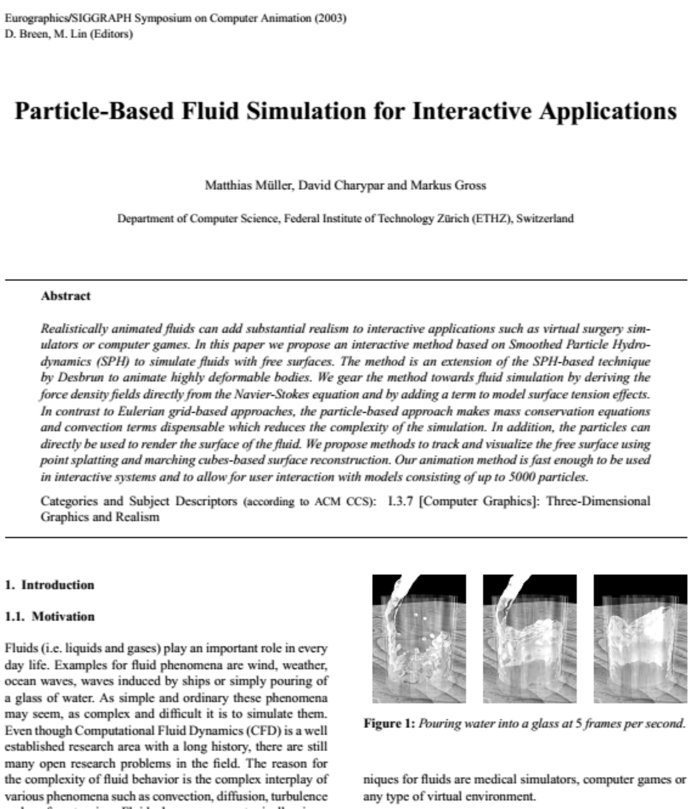


Smoothed Particle Hydrodynamics (SPH)

Theory

Smoothed Particle Hydrodynamics (SPH)

Physical quantity is represented as weighted average of the physical quantity of the neighbor particles using a kernel.



Eurographics/SIGGRAPH Symposium on Computer Animation (2003)
D. Breen, M. Lin (Editors)

Particle-Based Fluid Simulation for Interactive Applications

Matthias Müller, David Charypar and Markus Gross

Department of Computer Science, Federal Institute of Technology Zürich (ETHZ), Switzerland

Abstract

Realistically animated fluids can add substantial realism to interactive applications such as virtual surgery simulators or computer games. In this paper we propose an interactive method based on Smoothed Particle Hydrodynamics (SPH) to simulate fluids with free surfaces. The method is an extension of the SPH-based technique by Desbrun to animate highly deformable bodies. We gear the method towards fluid simulation by deriving the force density fields directly from the Navier-Stokes equation and by adding a term to model surface tension effects. In contrast to Eulerian grid-based approaches, the particle-based approach makes mass conservation equations and convection terms dispensable which reduces the complexity of the simulation. In addition, the particles can directly be used to render the surface of the fluid. We propose methods to track and visualize the free surface using point splatting and marching cubes-based surface reconstruction. Our animation method is fast enough to be used in interactive systems and to allow for user interaction with models consisting of up to 5000 particles.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

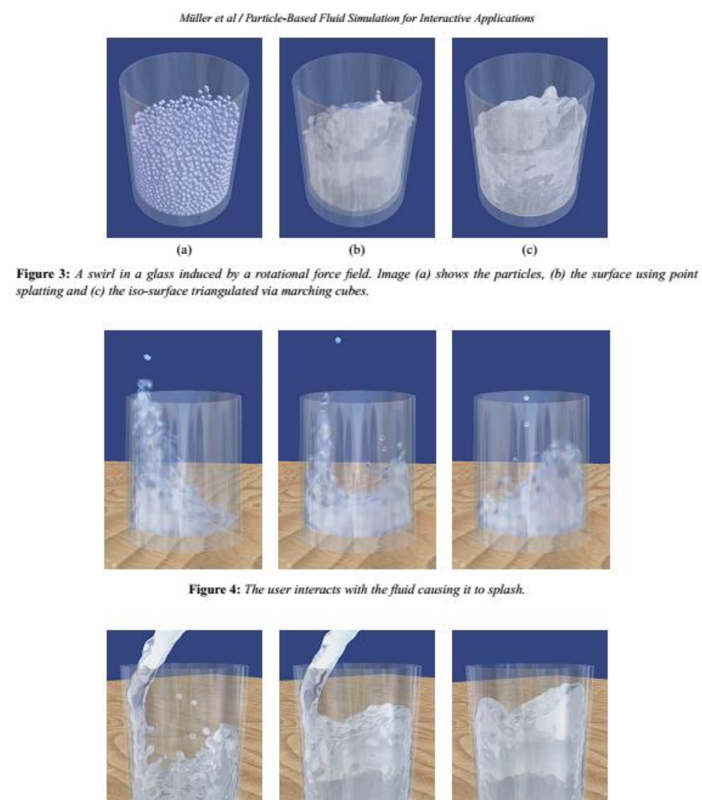
1. Introduction

1.1. Motivation

Fluids (i.e. liquids and gases) play an important role in every day life. Examples for fluid phenomena are wind, weather, ocean waves, waves induced by ships or simply pouring of a glass of water. As simple and ordinary these phenomena may seem, as complex and difficult it is to simulate them. Even though Computational Fluid Dynamics (CFD) is a well established research area with a long history, there are still many open research problems in the field. The reason for the complexity of fluid behavior is the complex interplay of various phenomena such as convection, diffusion, turbulence

Figure 1: *Pouring water into a glass at 5 frames per second.*

niques for fluids are medical simulators, computer games or any type of virtual environment.



Müller et al / Particle-Based Fluid Simulation for Interactive Applications

(a) (b) (c)

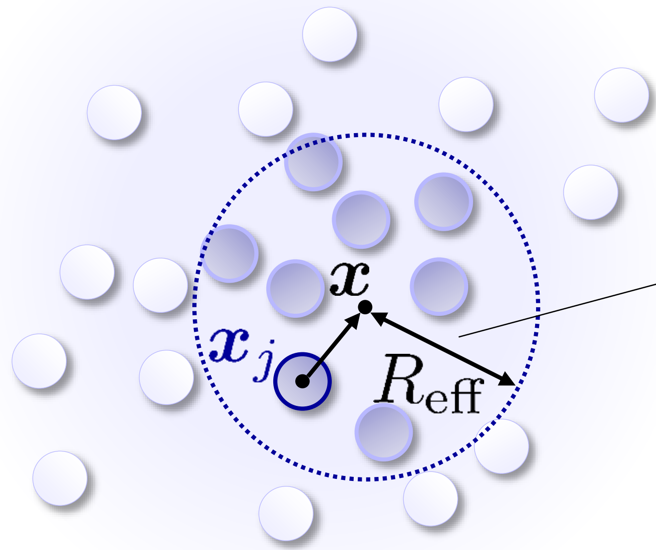
Figure 3: *A swirl in a glass induced by a rotational force field. Image (a) shows the particles, (b) the surface using point splatting and (c) the iso-surface triangulated via marching cubes.*

Figure 4: *The user interacts with the fluid causing it to splash.*

Interpolation of physical quantity

Relative position

$$\boldsymbol{r}_j = \boldsymbol{x} - \boldsymbol{x}_j$$



Effective radius of interaction

Physical quantity at position $\boldsymbol{x}$

Physical quantity of j-th particle

$$A(\boldsymbol{x}) = \sum_j m_j \frac{A_j}{\rho_j} W(\boldsymbol{r}_j, R_{\text{eff}})$$

Kernel function

Mass of j-th particle

density of j-th particle


ξ = 0.2

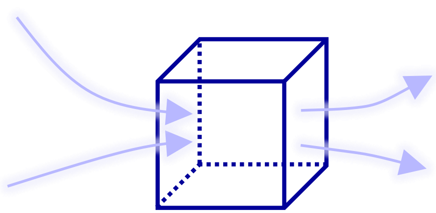
Equation to solve in SPH

- Law of conservation of mass $\nabla \cdot (\rho \boldsymbol{v}) = 0$ (incompressible)

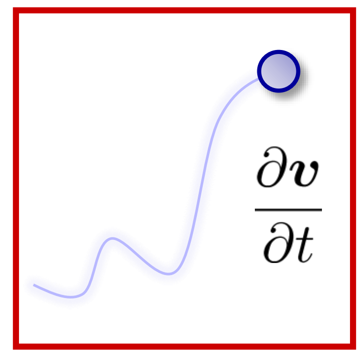
  SPH simulation with **a constant number of particle**

  and **a constant particle mass** satisfies this formula

- Law of conservation of momentum



$$\frac{\partial \boldsymbol{v}}{\partial t} + \underbrace{(\boldsymbol{v} \cdot \nabla)\boldsymbol{v}}_{\text{Advection}}$$



$$\frac{\partial \boldsymbol{v}}{\partial t}$$

$$\frac{\partial \boldsymbol{v}}{\partial t} = \underbrace{-\frac{1}{\rho}\nabla p}_{\text{Pressure term}} + \underbrace{\mu \nabla^2 \boldsymbol{v}}_{\text{Viscosity term}} + \underbrace{\boldsymbol{f}_{\text{ext}}}_{\text{External force}}$$

5 steps to simulate fluid behavior using SPH

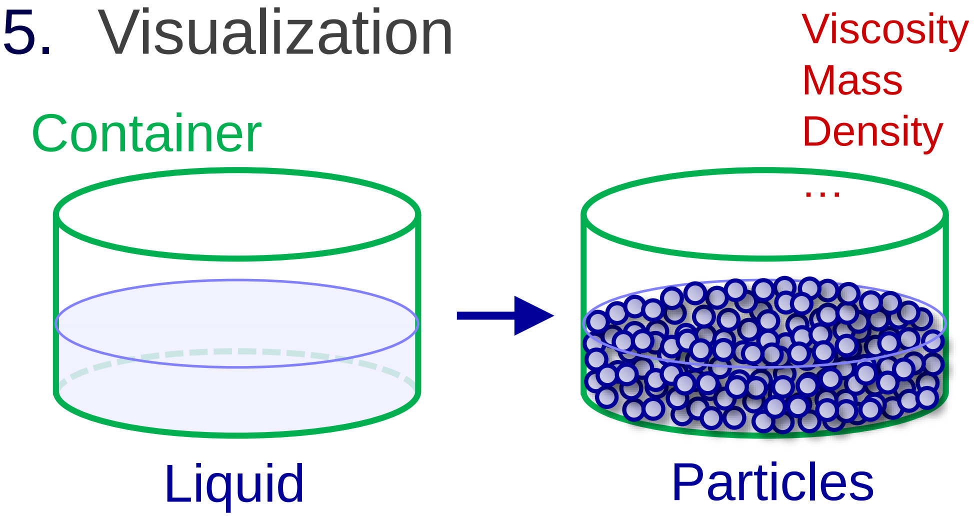
1. Particle generation
2. Parameter setting
3. Solving Navier-Stokes Equation
4. Update particle position
5. Visualization

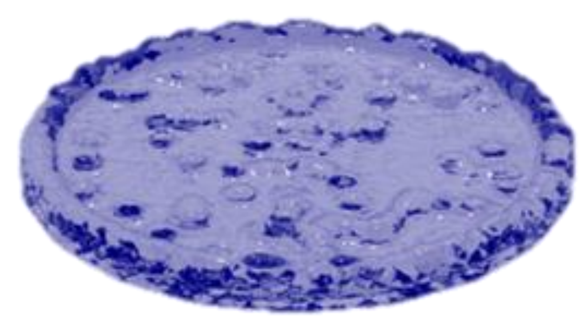
$$\frac{\partial \boldsymbol{v}}{\partial t} = -\frac{1}{\rho}\nabla p + \mu \nabla^2 \boldsymbol{v} + \boldsymbol{f}_{\text{ext}}$$

\+ Boundary condition

$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_n + \left(-\frac{1}{\rho}\nabla p_n + \mu\nabla^2 \boldsymbol{v}_n + \boldsymbol{f}_{\text{ext,n}}\right)\Delta t$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{v}_{n+1}\Delta t$$

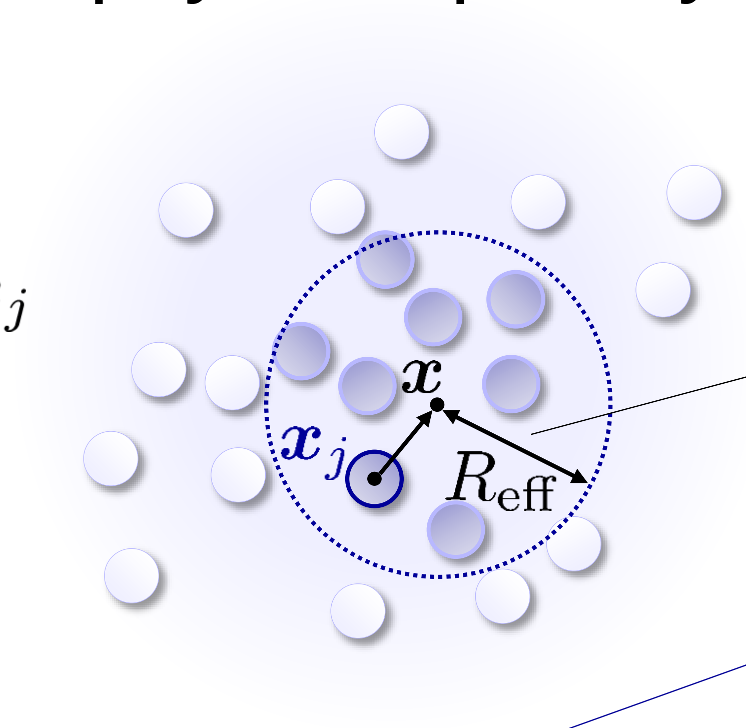




Interpolation of physical quantity

Relative position

$$\boldsymbol{r}_j = \boldsymbol{x} - \boldsymbol{x}_j$$



Effective radius of interaction

Physical quantity at position $\boldsymbol{x}$

Physical quantity of j-th particle

$$A(\boldsymbol{x}) = \sum_j m_j \frac{A_j}{\rho_j} W(\boldsymbol{r}_j, R_{\mathrm{eff}})$$

Kernel function

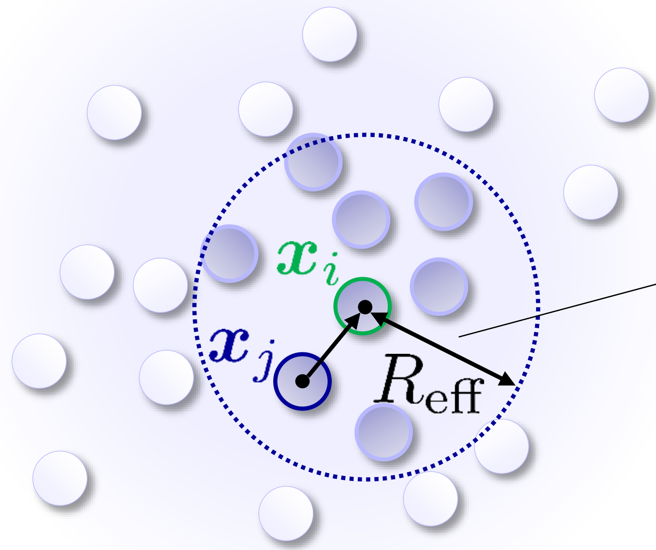
Mass of j-th particle

density of j-th particle

Interpolation of physical quantity

Relative position

$$\boldsymbol{r}_{ji} = \boldsymbol{x}_i - \boldsymbol{x}_j$$



Effective radius of interaction

Physical quantity of j -th particle

Physical quantity of j -th particle

$$A_i = \sum_j m_j \frac{A_j}{\rho_j} \underline{W(\boldsymbol{r}_{ji}, R_{\mathrm{eff}})}$$

Kernel function

Mass of j-th particle

density of j -th particle

Density, pressure and viscosity in SPH

- Density:

M. Muller et al., 2003

$$\rho_i = \sum_j m_j W_{\text{poly6}}(\boldsymbol{r}_{ji}, R_{\text{eff}})$$

- Pressure term:

$$-\frac{1}{\rho_i}\nabla p_i = -\sum_j m_j \frac{p_i + p_j}{2\rho_i\rho_j}\nabla W_{\text{spiky}}(\boldsymbol{r}_{ji}, R_{\text{eff}})$$

$$p_i = C(\rho_i - \rho_0)$$

Gas constant C

Rest density ρ_0

- Viscosity term:

$$\mu\nabla^2 v_i = \mu\sum_j m_j \frac{\boldsymbol{v}_j - \boldsymbol{v}_i}{\rho_j}\nabla^2 W_{\text{viscosity}}(\boldsymbol{r}_{ji}, R_{\text{eff}})$$

Properties of kernel function

- The value depends on distance $R_{ji} = |\boldsymbol{r}_{ji}| = |\boldsymbol{x}_i - \boldsymbol{x}_j|$
- The value becomes large as the decrease of the distance
- The function value is zero outside of the effective radius
- Omnidirectional integration of kernel function becomes 1

Kernel functions

- Poly6 kernel to interpolate density

$$W_{\text{poly6}}(\boldsymbol{r}_{ji}, R_{\text{eff}}) = \frac{315}{64\pi R_{\text{eff}}^9} \begin{cases} (R_{\text{eff}}^2 - R_{ji}^2)^3 & 0 \le R_{ji} \le R_{\text{eff}} \\ 0 & otherwise \end{cases}$$

- Spiky kernel to interpolate pressure term

$$W_{\text{spiky}}(\boldsymbol{r}_{ji}, R_{\text{eff}}) = \frac{15}{\pi R_{\text{eff}}^6} \begin{cases} (R_{\text{eff}} - R_{ji})^3 & 0 \le R_{ji} \le R_{\text{eff}} \\ 0 & otherwise \end{cases}$$

$$\nabla W_{\text{spiky}}(\boldsymbol{r}_{ji}, R_{\text{eff}}) = -\frac{45}{\pi R_{\text{eff}}^6} \begin{cases} (R_{\text{eff}} - R_{ji})^2 \frac{\boldsymbol{r}_{ji}}{R_{ji}} & 0 \le R_{ji} \le R_{\text{eff}} \\ 0 & otherwise \end{cases}$$

- Viscosity kernel to interpolate viscosity term

$$W_{\text{viscosity}}(\boldsymbol{r}_{ji}, R_{\text{eff}}) = \frac{15}{2\pi R_{\text{eff}}^3} \begin{cases} -\frac{R_{ji}^3}{2R_{\text{eff}}^3} + \frac{R_{ji}^2}{R_{\text{eff}}^2} + \frac{R_{\text{eff}}}{2R_{ji}} - 1 & 0 \le R_{ji} \le R_{\text{eff}} \\ 0 & otherwise \end{cases}$$

$$\nabla^2 W_{\text{viscosity}}(\boldsymbol{r}_{ji}, R_{\text{eff}}) = \frac{45}{\pi R_{\text{eff}}^6} \begin{cases} R_{\text{eff}} - R_{ji} & 0 \le R_{ji} \le R_{\text{eff}} \\ 0 & otherwise \end{cases}$$

Calculation of new particle position

- Discrete time:

$$t \to n\Delta t \equiv t_n$$

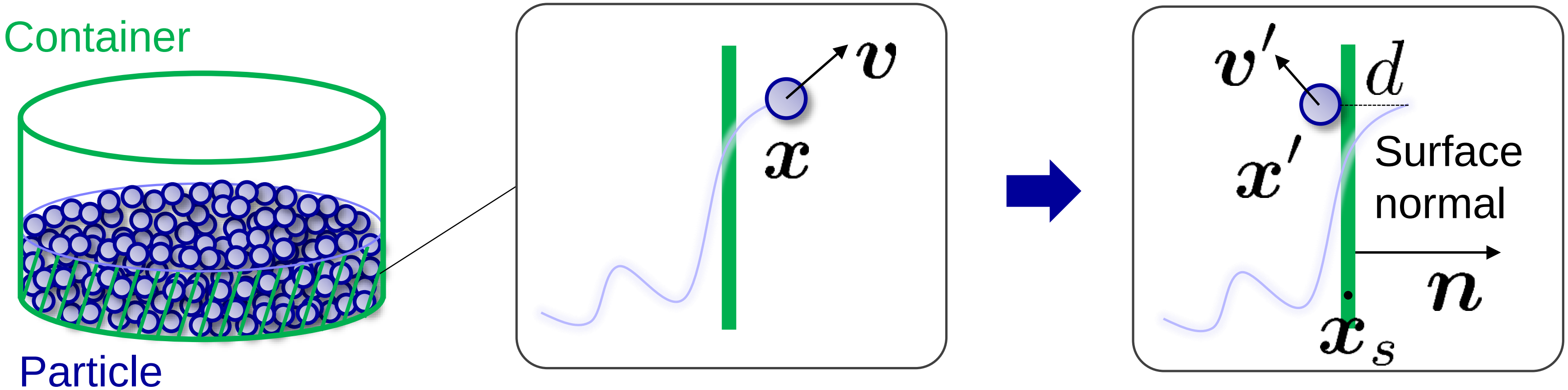
- Differential equation:

$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_n + \left(-\frac{1}{\rho}\nabla p_n + \mu\nabla^2\boldsymbol{v}_n + \boldsymbol{f}_{\mathrm{ext,n}}\right)\Delta t$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{v}_{n+1}\Delta t$$

$$t_{n+1} = (n+1)\Delta t \qquad\qquad t_0 = 0$$

Boundary condition



$$\text{if} \quad \boldsymbol{n} \cdot (\boldsymbol{x} - \boldsymbol{x}_s) > 0$$

Friction   Restitution

$$\Rightarrow \quad \boldsymbol{v}' = (1 - \mu_{\text{fric}})(\boldsymbol{v} - (1 + e_{\text{wall}})(\boldsymbol{n} \cdot \boldsymbol{v})\boldsymbol{n})$$

$$\boldsymbol{x}' = \boldsymbol{x} - (\boldsymbol{n} \cdot (\boldsymbol{x} - \boldsymbol{x}_s))\boldsymbol{n}$$

List of variables

- Particle
  - position
  - velocity
  - force
  - density
  - pressure

```
Vec3d position;
Vec3d velocity;
Vec3d force;
double rho;
double pressure;
```

- Neighbor
  - number of particles
  - index list

```
int num_particles;
int index[MAX_NUM_INDEX];
```

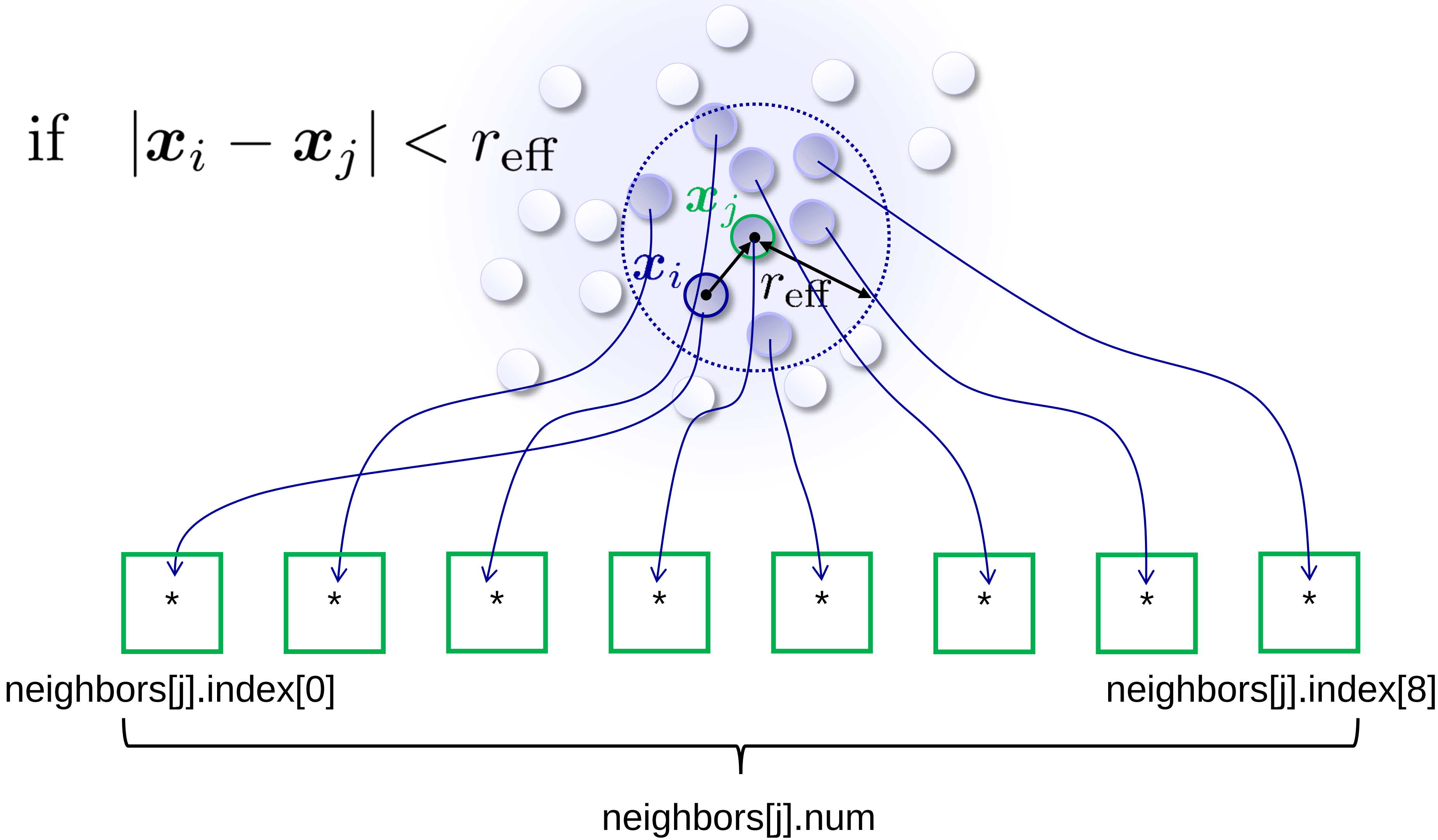
- SPH simulation
  - delta time, digitalized step, simulation time
  - gravity
  - gas constant
  - mass, density, viscosity
  - effective radius, particle radius
  - restitution, friction
  - coefficient of Poly6 kernel
  - coefficient of Spiky kernel
  - coefficient of Viscosity Kernel
  - boundary condition
  - number of particles
  - particles
  - neighbors

List of functions

- Initialization
- Load SPH parameters
- Set initial position of liquid particles and container based on mesh data
- Generate index lists of the neighbor particles
- Interactive gravity setting
- Run simulation
- Calculate density and pressure
- Calculate force
- Update particle position
- Release memory for SPH pointer
- Rendering function of the particles and container
- Save current scene as Pov-Ray file

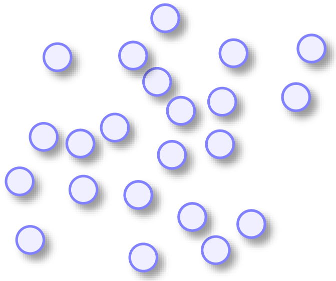
Neighbor map

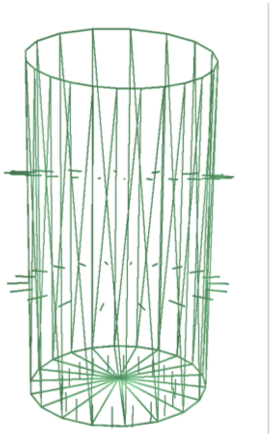
$$\text{if} \quad |\boldsymbol{x}_i - \boldsymbol{x}_j| < r_{\text{eff}}$$



Use mesh data as boundary condition

```
for i = 0 to np-1 do
    for j = 0 to nf-1 do
        if f[j].n*(p[i].p-f[j].p0)>0 then
            set p[i].v = (1-fric)(p[i].v-(1+restit)(f[j].n*p[i].v) f[j].n);
            set p[i].p=p[i].p-(f[j].n*(p[i].p-f[j].p0));
        end if
    end do
end do
```





np	:num_particles	nf	:container.num_facet
p[i].p	:particles[i].position	f[j].p0	:container.facet[j].position[0]
p[i].v	:particles[i].velocity	f[j].n	:container.facet[j].normal

Parallelization using OpenMP

- Include header

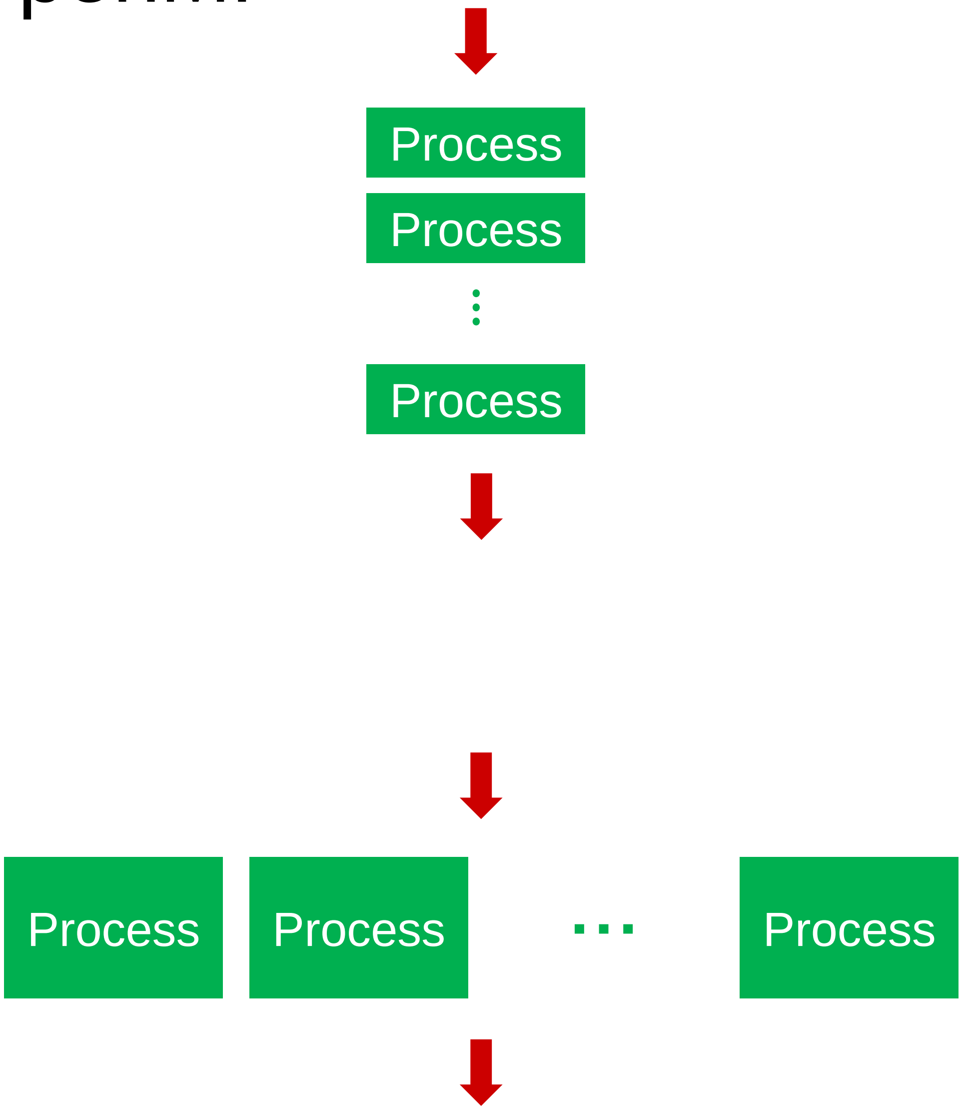
```
#ifdef _OPENMP
#include <omp.h>
#endif
```

- How to use it

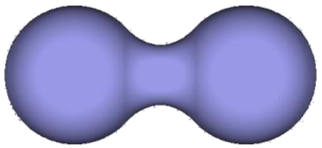
```
#ifdef _OPENMP
#pragma omp parallel private( i )
#pragma omp for
#endif
for( i = 0; i < NUM; i++ ){
        process to be parallelized
}
```

- Compile

```
gcc filename.c -fopenmp -o filename
```



Visualization using POV-Ray

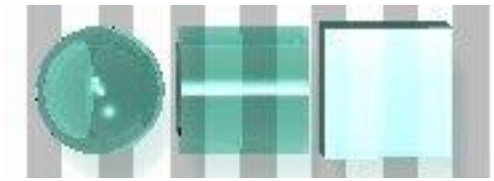


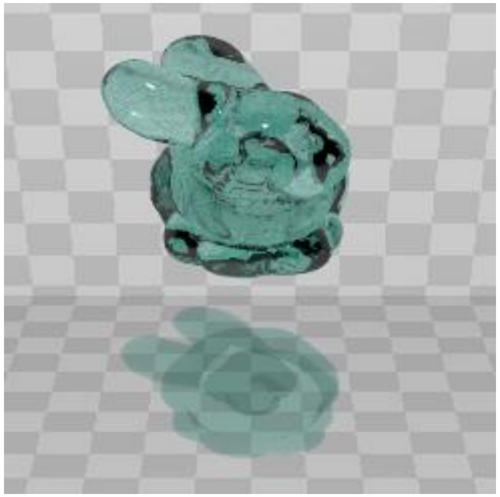
- Install
  - sudo apt-get install povray

Blobs are described as spheres and cylinders covered with "goo" which stretches to smoothly join them.
from POV-Ray 3.6 Documentation

- Scene file

```
#include "colors.inc"
. . .
camera {perspective location <0, 50, 100> look_at <0, -20, 0.0> up <0,1,0> right<1,0,0>}
light_source { <0, 100, 100> color White }
blob { threshold 5
    sphere { <3.850612,20.794310,-0.000577>, 2, 2 }
    . . .
    sphere { <-2.241859,6.902092,-0.417286>, 2, 2 }
    pigment {color Clear}
    finish { F_Glass1 }
    interior {  I_Glass1  fade_color Col_Emerald_03  ior 1.33 }
}
```





- Rendering command
  - povray filename.pov

Physics based simulation

Discussion

What you have to consider in simulation

- Explain your analysis design logically.
  - How to construct a model for the simulation.
  - Which is your analysis mode, static, dynamic, frequency, or bulking analysis?
  - What is boundary condition, input and output?
- It is possible to show the property but difficult to explain the essence.
- The investigation of validity is difficult.
- No appropriate imitation, no valid result.
- You cannot find your mistake if your model is too complicated.

Reference

- 石渡良三，流体力学入門，森北出版，東京，2004.
- 粒子法のプログラム, kamonama@blogger http://kamonama.blogspot.jp/2009/02/blog-post_23.html
- 粒子法 http://www.slis.tsukuba.ac.jp/~fujisawa.makoto.fu/cgi-bin/wiki/index.php?%CE%B3%BB%D2%CB%A1
- M. Muller, D. Charypar and M. Gross, Particle-based Fluid Simulation for Interactive Applications, In Proc. SCA2003, pp.154-159, 2003.
- POV-Ray, http://www.povray.org/