CSCI1950V Project 4 : Smoothed Particle Hydrodynamics

Due Date : Midnight, Friday March 23

1 Background

For this project you will implement a fluid simulation using Smoothed Particle Hydrodynamics (SPH). The main reference we will use is Lagrangian Fluid Dynamics Using Smoothed Particle Hydrodynamics by Kellager et al [3]. In particular, most of the equations below can be found in section four of the paper.

If you implemented the NBody project before this correctly, this project should not be too difficult to write. Essentially the gravitational forces you computed for the previous project will be replaced by the forces of fluid dynamics. The hardest part will probably adding in all the extra forces and tracking down any bugs.

2 Requirements

Your SPH simulation can either be 2D or 3D. As with all other projects, computation should be done in GLSL, however for the 3D visualization we only require point sprites (for 2D) or spheres (for 3D). You do not need to compute an explicit fluid surface for this project - a particle visualization is sufficient. However, we encourage you to color the particles or spheres according to some metric (ie. pressure).

While doing this in 3D means you have to change less code from your previous project (and things in 3D are of course cool), with a 2D simulation you will be able to add more particles per screen space, making the simulation look more interesting - it's up to you.

You are required to implement all the internal and external forces covered in the Kelager 2006 paper and presented below, for one type (either liquid or gas).

Force	Expression	Liquid / Gas
Mass Density 3.2.1	$\sum m_j W_{default}(\mathbf{r}-\mathbf{r}_j,h)$	L+G
Surface normal 3.2.2	$\sum \frac{m_j}{\rho_j}\nabla W_{default}(\mathbf{r}-\mathbf{r}_j,h)$	L
Pressure 3.2.1	$-\sum_{j\neq i}\frac{p_i+p_j}{2}\frac{m_j}{\rho_j}\nabla W_{pressure}(\mathbf{r}_i-\mathbf{r}_j,h)$	L+G
Viscosity 3.2.1	$\frac{\mu}{\rho_i}\sum_j(\mathbf{u}_j-\mathbf{u}_i)m_j\nabla^2 W_{viscosity}(\mathbf{r}-\mathbf{r}_j,h)$	L+G
Gravity 3.2.2	$\rho_i\mathbf{g}$	L
Buoyancy 3.2.2	$b(\rho_i-\rho_0)\mathbf{g}$	G
Surface tension 3.2.2	$-\sigma\frac{\mathbf{n}_i}{\|\mathbf{n}_i\|}\sum\frac{m_j}{\rho_j}\nabla^2 W_{default}(\mathbf{r}-\mathbf{r}_j,h)$	L

For example, if you choose to do liquid, you must implement mass, surface normal, pressure, viscosity, gravity, and surface tension.

Since each kernel W is zero outside of the core radius h, you should use the spatial hashing from the last project to quickly find all the nearest neighbors within about distance h so that you dont have to loop through every particle.

Your particles should be contained within a box and should properly interact with the walls and floor. A ceiling is optional. You must implement rudimentary collision handling against walls - this can be as simple as checking if the location of a particle is outside the bounds of a box and if so reflecting its velocity across the normal.

For the visualization you should use spheres or point sprites (or other similar particle-like objects). You should also draw the boundaries of your system (ie. the walls) - you probably want to make these a wireframe so that the particles are visible through the walls. We are not requiring any complex graphical shaders for this project.

You may keep the same integration scheme as the last project (Verlet), but feel free to experiment with other integration schemes as well.

Finally you should experiment with different fluid types (ie. change the mass and viscosity), as well as interactions between two or more different types of fluids.

Your writeup should be contained within a seperate html directory in your project.

3 Governing Equations

This section simply summarizes the derivation and governing of equations of your simulation. You can find all of these in .

3.1 Smoothed Particle Hydrodynamics (SPH)

SPH is essentially an interpolation method and is based off of integral interpolants which use kernels to approximate the response of a delta function. By definition, the integral interpolant, A_I of a function $A(\mathbf{r})$ over some space Ω is given by

$$A_I(\mathbf{r}) = \int_\Omega A(\mathbf{r}')W(\mathbf{r} - \mathbf{r}', h)d\mathbf{r}'$$

where $\mathbf{r} \in \Omega$, and W is a smoothing kernel with a width, or core radius of h. The choice of smoothing kernel W is important and careful selection can affect the accuracy of the interpolant. The kernel W satisfies the following four properties

$$\begin{aligned}
\int_\Omega W(\mathbf{r}, h)d\mathbf{r} &= 1 \\
\lim_{h \to 0} W(\mathbf{r}, h) &= \delta(\mathbf{r}) \\
W(\mathbf{r}, h) &\geq 0 \\
W(\mathbf{r}, h) &= W(-\mathbf{r}, h)
\end{aligned}$$

In practice, most kernels are Gaussian like, but the exact choice of kernel largely depends on what you are trying to interpolate.

The discretized version of the SPH interpolant can be expressed as a summation

$$A_S(\mathbf{r}) = \sum_j A_j V_j W(\mathbf{r} - \mathbf{r}_j, h)$$

where the summation is over all particles j, and V_j is the volume or area (in 2D) taken by particle j. Note that this holds only when A_j is close to constant on V_j (and is the integral as the volume of each particle goes to zero).

Now recall that

$$V = \frac{m}{\rho}$$

Substituting this into gives us

$$A_S(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h)$$

Now suppose we take the partial derivative between two particles, i and j in some direction, (in this case, the x direction),

$$\frac{\partial}{\partial x} A_S(\mathbf{r}_i) = \frac{\partial}{\partial x}\left(A_j \frac{m_j}{\rho_j} W(\mathbf{r}_i - \mathbf{r}_j, h)\right)$$

Note that the quantity

$$A_j \frac{m_j}{\rho_j}$$

is constant relative to the x direction (or any other direction for that matter), which means we can treat it as a constant under differentiation. Then

$$\frac{\partial}{\partial x} A_S(\mathbf{r}_i) = \left(A_j \frac{m_j}{\rho_j}\right) \frac{\partial}{\partial x} W(\mathbf{r}_i - \mathbf{r}_j, h)$$

It follows that the gradient of the integral interpolant for is simply

$$\nabla A_S(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} \nabla W(\mathbf{r} - \mathbf{r}_j, h)$$

Similarly,

$$\nabla^2 A_S(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} \nabla^2 W(\mathbf{r} - \mathbf{r}_j, h)$$

3.2 Navier Stokes Equations

The classical formulation of an incompressible Newtonian fluid over time is given by the Navier Stokes (NS) equations.

$$\overbrace{\underbrace{\rho}_{\text{Mass density}} \left(\underbrace{\frac{\partial \mathbf{u}}{\partial t}}_{\text{Unsteady acceleration}} + \underbrace{\mathbf{u} \cdot \nabla \mathbf{u}}_{\text{Convective acceleration}} \right)}^{\text{Inertia (per volume)}} = \overbrace{\underbrace{-\nabla p}_{\text{Pressure gradient}} + \underbrace{\mu \nabla^2 \cdot \mathbf{u}}_{\text{Viscosity}}}^{\text{Divergence of stress}} + \underbrace{\mathbf{f}}_{\text{Other body forces}}$$

Because we assume that density is constant (since the fluid is incompressible),

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

and the formula simplifies to

$$\rho \frac{\partial \mathbf{u}}{\partial t} = -\nabla p + \mu \nabla^2 \cdot \mathbf{u} + \mathbf{f}^{external}$$

We can rewrite this to become

$$\begin{aligned} \rho \frac{\partial \mathbf{u}}{\partial t} &= \mathbf{f}^{internal} + \mathbf{f}^{external} \\ \frac{\partial \mathbf{u}}{\partial t} &= \left(\mathbf{f}^{internal} + \mathbf{f}^{external} \right) / \rho \\ \mathbf{a}_i &= \left(\mathbf{f}_i^{internal} + \mathbf{f}_i^{external} \right) / \rho_i \qquad (1) \end{aligned}$$

Where the interal forces are the pressure and viscosity, while the external forces are things such as gravity. Thus, the acceleration of a particle i is given by the sum of its forces divided by ρ, the mass-density.

3.2.1 Internal Forces

Mass Density Using the SPH formulation to approximate mass density is fairly straightforward. The quantity we would like to interpolate, $A(\mathbf{r})$ is simply the mass density, ρ. Thus,

$$A_S(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h)$$

becomes

$$\begin{aligned}\rho_S(\mathbf{r}) &= \sum_j \rho_j \frac{m_j}{\rho_j} W(\mathbf{r}-\mathbf{r}_j, h) \\ &= \sum_j m_j W(\mathbf{r}-\mathbf{r}_j, h)\end{aligned}$$

Since we are evaluting $\rho_S(\mathbf{r})$ only at particle locations, for each particle i, this becomes

$$\rho_i = \sum_j m_j W_{default}(\mathbf{r}-\mathbf{r}_j, h) \tag{2}$$

For the kernel, W, [4] suggests

$$W_{default}(\mathbf{r}, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - ||\mathbf{r}||^2)^3 & 0 \leq ||\mathbf{r}|| \leq h \\ 0 & ||\mathbf{r}|| > h \end{cases} \tag{3}$$

Pressure Gradient Recall that the pressure gradient is given by $-\nabla p$. Thus the SPH formulation is simply

$$-\nabla p_i = -\sum_j p_j \frac{m_j}{\rho_j} \nabla W_{pressure}(\mathbf{r}-\mathbf{r}_j, h)$$

Unfortunately, this formulation is not symmetric (consider the case of two particles where particle 1 is dependent on particle 2 and vice versa - which is clearly not symmetrical), violating the action-reaction law. SPH has various symmetrical formulations, one of which is

$$-\nabla p_i = -\sum_{j \neq i} \frac{p_i + p_j}{2} \frac{m_j}{\rho_j} \nabla W_{pressure}(\mathbf{r}_i - \mathbf{r}_j, h) \tag{4}$$

Where p can be calculated using a modified ideal gas state equation (with an added rest pressure) such that

$$\begin{aligned}(p + p_0)V &= k \\ p + k\rho_0 &= k\rho \\ p &= k(\rho - \rho_0)\end{aligned} \tag{5}$$

Here, ρ_0 corresponds to the rest density. The kernel recommended by [1] and used in [4] is

$$W_{pressure}(\mathbf{r}, h) = \frac{15}{\pi h^6} \begin{cases} (h - ||\mathbf{r}||)^3 & 0 \leq ||\mathbf{r}|| \leq h \\ 0 & ||\mathbf{r}|| > h \end{cases}$$

Which has a gradient of

$$\nabla W_{pressure}(\mathbf{r}, h) = \frac{-45}{\pi h^6} \frac{\mathbf{r}}{||\mathbf{r}||} (h - ||\mathbf{r}||)^2 \tag{6}$$

Viscosity From the NS equations, viscosity is given by $\mu\nabla^2 \cdot \mathbf{u}$, where μ is the viscosity coefficient (and controls how viscous the fluid is). Therefore, the straightforward SPH formulation is

$$\mu\nabla^2 \cdot \mathbf{u}_i = \mu \sum_{j \neq i} \mathbf{u}_j \frac{m_j}{\rho_j} \nabla^2 W(\mathbf{r} - \mathbf{r}_j, h)$$

However, like the pressure gradient, we require that viscosity be symmetric, which the above formulation is not. In this case we can symmetrize the SPH approximation as

$$\mu\nabla^2 \cdot \mathbf{u}_i = \frac{\mu}{\rho_i} \sum_j (\mathbf{u}_j - \mathbf{u}_i) m_j \nabla^2 W_{viscosity}(\mathbf{r} - \mathbf{r}_j, h) \tag{7}$$

[23] suggests a kernel of

$$W_{viscosity}(\mathbf{r}, h) = \frac{15}{2\pi h^3} \begin{cases} -\frac{||\mathbf{r}||^3}{2h^3} + \frac{||\mathbf{r}||^2}{h^2} + \frac{h}{2||\mathbf{r}||} - 1 & 0 \leq ||\mathbf{r}|| \leq h \\ 0 & ||\mathbf{r}|| > h \end{cases}$$

which has a Laplacian of

$$\nabla^2 W_{viscosity}(\mathbf{r}, h) = \frac{45}{\pi h^6}(h - ||\mathbf{r}||) \tag{8}$$

3.2.2 External Forces

Gravity The force of gravity on particle i can be computed simply as

$$\mathbf{f}_i = \rho_i \mathbf{g} \tag{9}$$

Buoyancy The force of buoyancy on particle i is

$$\mathbf{f}_i = b(\rho_i - \rho_0)\mathbf{g} \tag{10}$$

where $b > 0$ is the buoyancy diffusion coefficient.

Surface Tension The surface tension of a liquid is given by [4]

$$\mathbf{f}_i = -\sigma \nabla^2 c_i \frac{\mathbf{n}_i}{||\mathbf{n}_i||} \tag{11}$$

Where σ is the surface tension coefficient and depends on the the fluids that form the surface, c_i is the colorfield of particle i, (which has a value of one at the particle's location and zero elsewhere) whose smoothed SPH formulation is

$$c_i = \sum_j \frac{m_j}{\rho_j} W_{default}(\mathbf{r}_i - \mathbf{r}_j, h) \tag{12}$$

The normal $\mathbf{n}$ can be computed as the gradient of the color field, c.

$$\begin{aligned} \mathbf{n}_i &= \nabla c(\mathbf{r}_i) \\ &= \sum_j \frac{m_j}{\rho_j} \nabla W_{default}(\mathbf{r}_i - \mathbf{r}_j, h) \end{aligned}$$

Keep in mind that $||\mathbf{n}_i||$ may go to zero, and division by zero is bad.

Note that

$$\begin{aligned} \nabla W_{default}(\mathbf{r}, h) &= -\frac{945}{32\pi h^9} \mathbf{r}(h^2 - ||\mathbf{r}||^2)^2 \\ \nabla^2 W_{default}(\mathbf{r}, h) &= -\frac{945}{32\pi h^9} (h^2 - ||\mathbf{r}||^2)(3h^2 - 7||\mathbf{r}||^2) \end{aligned}$$

4 Notes

Note that most of the force calculations depend on mass density (ρ). You probably need to have at least two stages, the first calculating mass density and storing in a texture. The second stage calculating the rest of the forces.

Good luck.

References

[1] M. Desbrun and M.-P. Cani. "Smoothed Particles: A new paradigm for animating highly deformable bodies". 1996.

[2] T. Harada, S. Koshizuka, and Y. Kawaguchi. "Smoothed Particle Hydrodynamics on GPUs". 2007.

[3] M. Kelager. "Lagrangian Fluid Dynamics Using Smoothed Particle Hydrodynamics". 2006.

[4] M. Muller, D. Charypar, and M.Gross. "Particle-Based Fluid Simulation for Interactive Applications". 2003.