

Particle Based Simulation

Ayden Cook

12/9/22

Table of contents

Introduction	2
Methods	2
Interactions	2
Particle Dynamics	2
Smoothed Particle Hydrodynamics (SPH)	2
Constraints	2
Extended Position Based Dynamics (XPBD)	2
Boundary Conditions	2
Results	2
Discussion	2
Conclusion	2
Update Loop	2
Relavent Links	3

Introduction

Methods

To remain very general purpose, modular, and extensible, the framework has been created in such a way that allows for any type of particle behavior to easily tap into the system's main update loop. Two different APIs have been created to do so.

The first API is called an **Interaction**, and the entrypoint for an Interaction occurs at the beginning of the main loop, while the second is called a **Constraint**, and occurs after the particle integration step.

Interactions

Particle Dynamics

Smoothed Particle Hydrodynamics (SPH)

The foundational starting point of the SPH method is the following identity:

$$A(\vec{r}) = \iiint_{\mathbb{R}^3} A(\vec{r}') \delta(\vec{r} - \vec{r}') dV' \quad (1)$$

By recognizing that $dm = \rho dV$, and by substituting a kernel approximation to the Dirac-Delta distribution, we arrive at the following continuous formula:

$$A(\vec{r}) = \iiint_{\mathbb{R}^3} \frac{A(\vec{r}')}{\rho(\vec{r}')} W(\vec{r} - \vec{r}', h) dm'$$

The final step is the discretization. Our spatial domain is sampled by a collection of particles, and so the integral over that domain will become the following discrete sum over the particles:

$$A(\vec{r}) = \sum_i^N m_i \frac{A_i}{\rho_i} W(\vec{r} - \vec{r}_i, h) \quad (2)$$

Equation 2 is the central equation to the SPH method, as it allows any set of continuum equations

to be discretized and for the interpolation of any continuous field using a set of discrete sampling particles.

In a continuum, the particles experience an acceleration due to the local pressure field¹.

$$\frac{d\vec{r}_i}{dt} = -\frac{1}{\rho_i} \nabla_{\vec{r}_i} p$$

Discretizing this force alone will allow us to capture a lot of the behavior of a fluid, and so it will serve as the primary example shown here.

Constraints

Extended Position Based Dynamics (XPBD)

Boundary Conditions

Results

Discussion

Conclusion

Update Loop

```
pub fn step_forward(&mut self, dt: f64) {
    if !self.running || dt == 0_f64 {
        return;
    }

    let sub_dt = dt / (self.substeps as f64);
    for _ in 0..self.substeps {
        for interaction in &mut self.interactions {
            interaction.handle(&mut self.particles, sub_dt);
        }

        for particle in &mut self.particles {
            particle.integrate(sub_dt);
        }
    }
}
```

¹They also experience other forces such as viscosity, body forces, elasticity, etc, but we can treat all of them as independent.

```
        particle.forces.clear();
    }

    for constraint in &mut self.constraints {
        constraint.project(&mut self.particles, sub_dt, false);
    }

    for particle in &mut self.particles {
        particle.update_vel(sub_dt);
    }
    self.time += dt;
}
```

Relavent Links

[Project GitHub Repository](#)