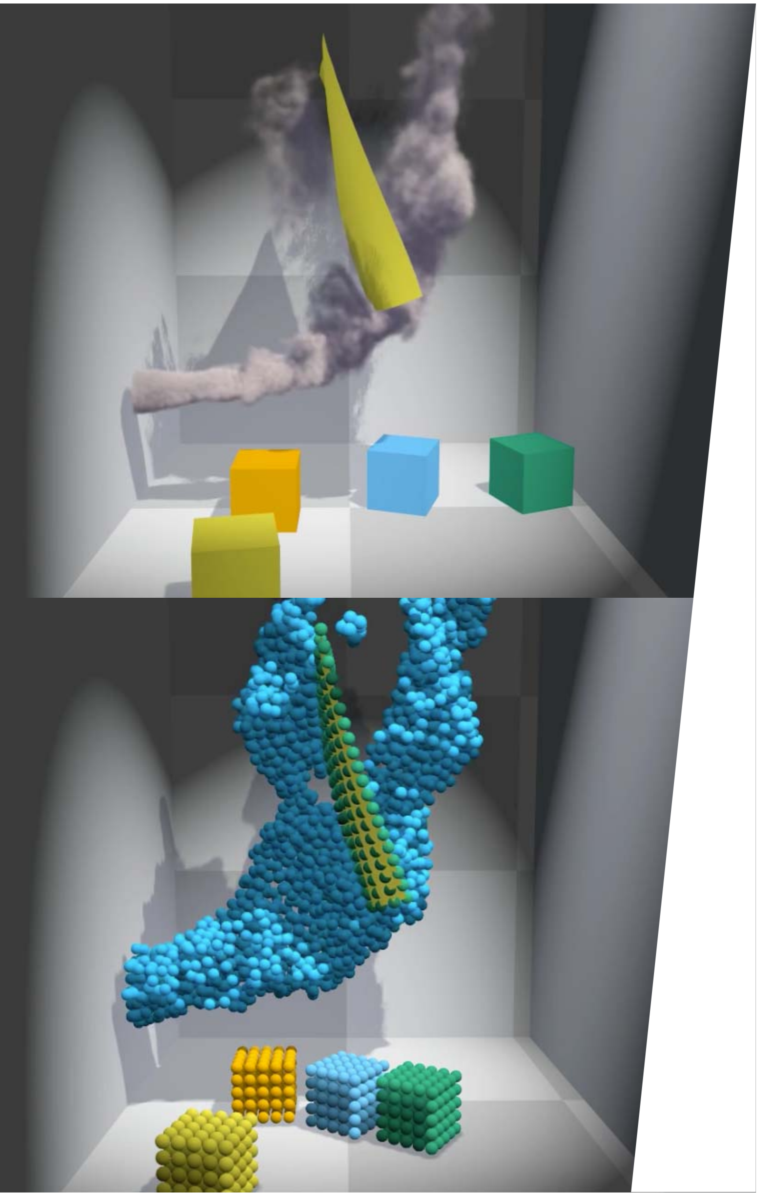
Unified Particle Physics for Real-Time Applications

By Bailey Allen

Paper by: Miles Macklin, Matthias Müller, Nuttapong Chentanez, Tae-Yong Kim



What is Unified Particle Physics?

Unified Particle Physics is a visual effects solver based on the concept that everything is represented by particles and constraints [1].

A solver is a piece of mathematical software that 'solves' a mathematical problem. In the context of this paper, a solver is an algorithm for simulating objects and Visual Effects.

Where does it fit in?

Solvers are used extensively for offline visual effects processing in video games and other interactive software[1].

The Unified Particle Physics system is currently being used in a technology developed by NVIDIA called FleX, which is a particle-based version of their popular PhysX engine [2], [3]

NVIDIA®

Why do we need a Unified Solver?

Usually, an application would have unique solvers for each type of object and interaction between the objects that they wish to simulate [1].

The goal of a Unified solver is to create one system that can simulate any object or interaction, which significantly improves maintainability as there is less code to write, as well as improves the reliability of object interactions [1].

What distinguishes the topic from associated topics?

Unified Particle physics is based on the paper Position-Based Dynamics [5].

It aimed to extend the functionality of the position based dynamics solver to a particle-based system [1]. Position Based Dynamics also did not have any method for handling Rigid bodies (Solid objects that cannot be misshapen)[5].

Other related Unified Solvers include Maya’s Nucleus solver[6] and Softimage’s Lagoa (Discontinued).

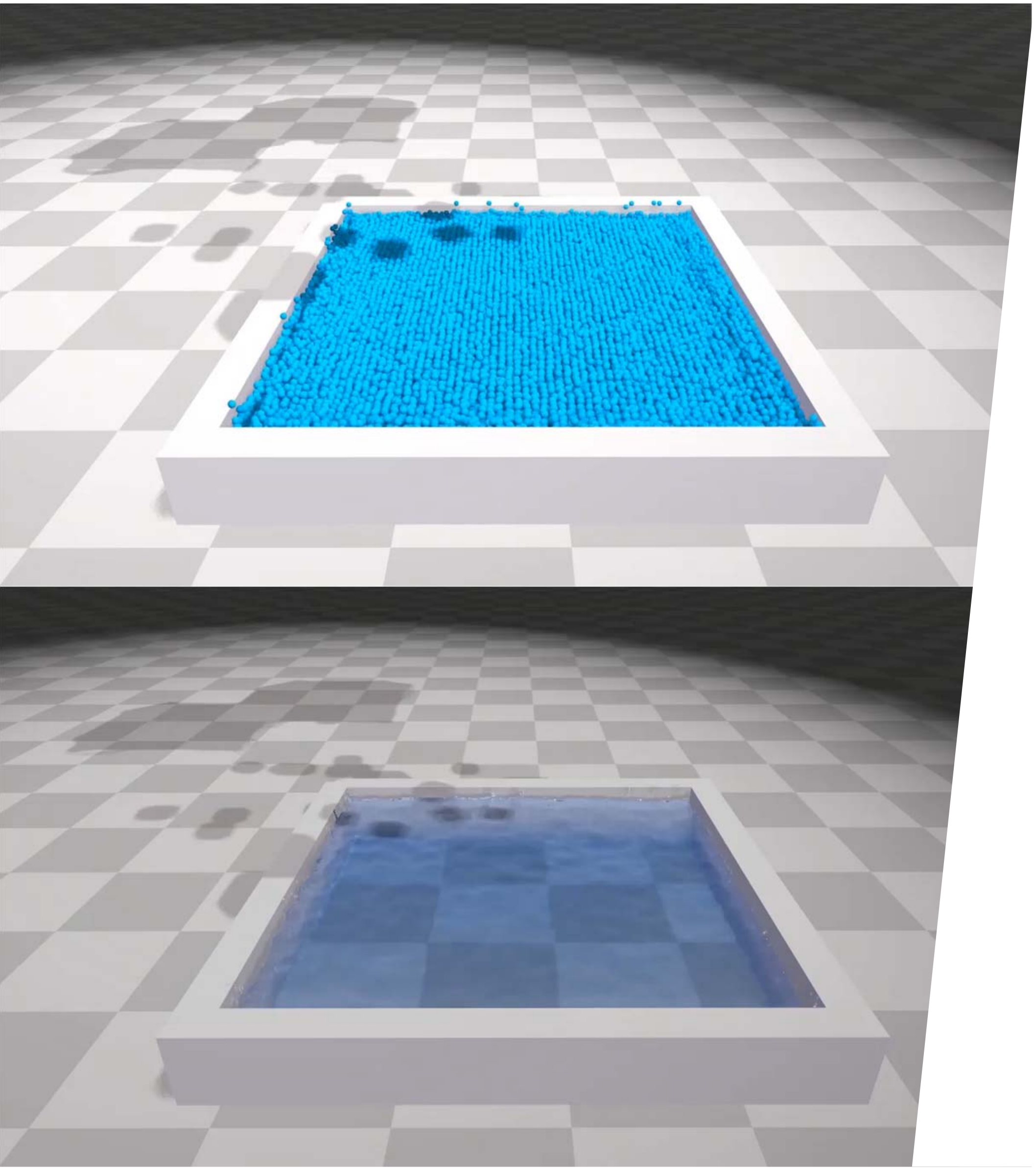
The Unified Particle Physics solver is able to perform similar simulations as these solvers, but its main advantage is that it can be rendered in real-time [7].

Objective?

“Everything is a set of particles connected by constraints” [7]

The goal of the paper was to reduce the number of solvers and redundant processor work in current graphics processing options and allow two-way interaction between all object types [1], [7].

This was to be achieved by simplifying all objects down to pure particles.



Unified Particle Physics in Action [8]

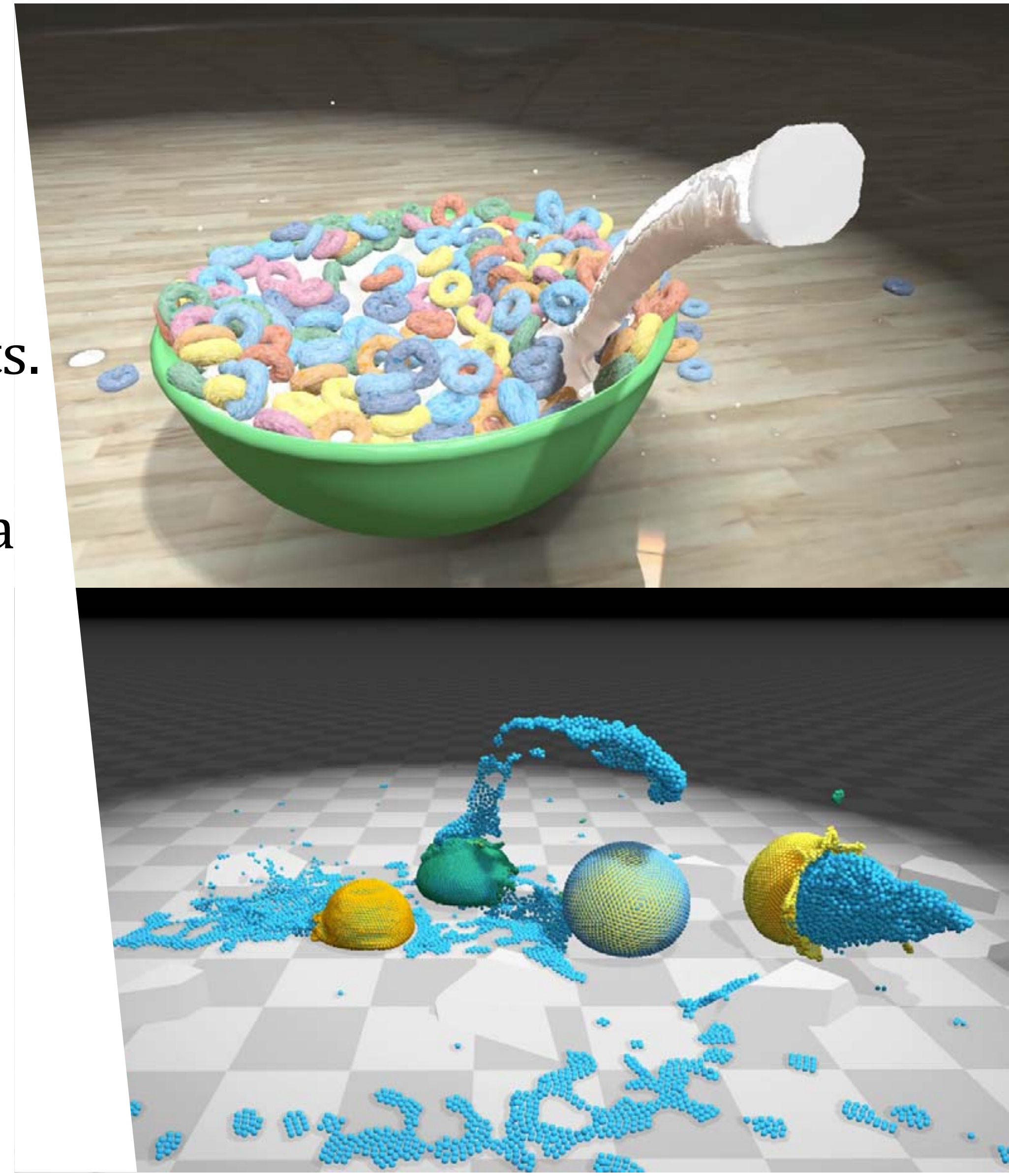
Advantages of this approach

Two-way interaction with all objects.

Simpler collision detection.

Improved workload scaling across a GPU [7].

A particle-based algorithm is well suited for parallel execution [1].



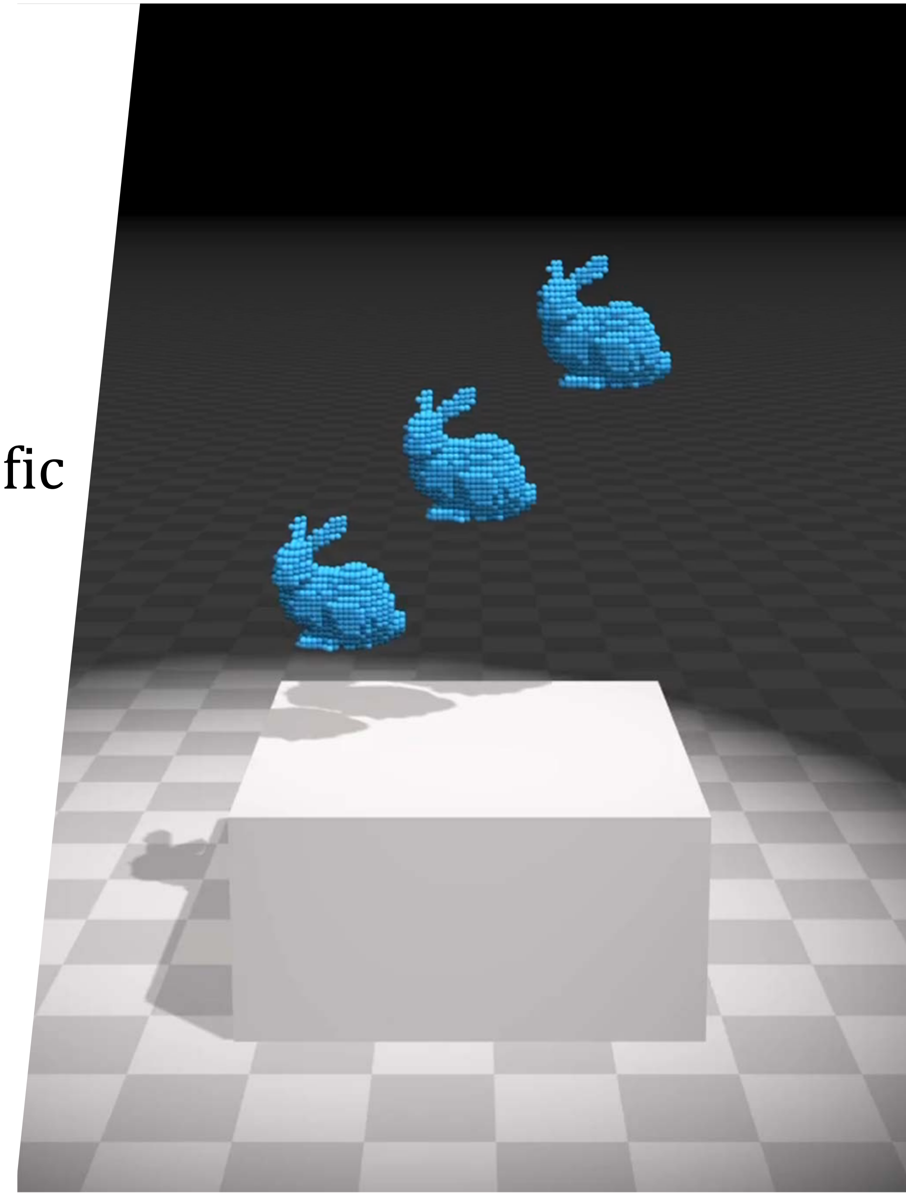
How is this achieved?

Everything is made of particles

No particle belongs to a specific object

Particle movement is restricted by specific constraints [7]:

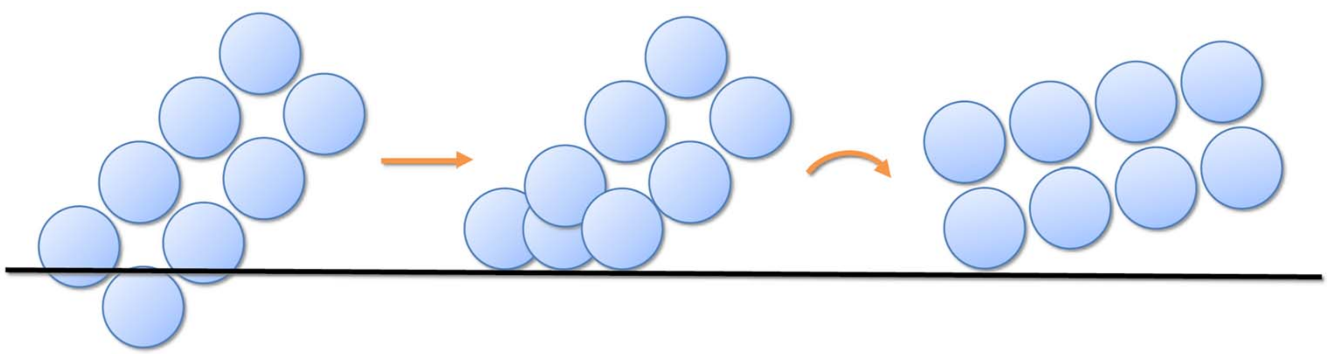
- Distance (Cloth and Clothing)
- Shape (Rigid and plastic objects)
- Density (Fluids)
- Volume (Inflatable objects)
- Contact (Friction, non-penetration)



Rigid Body Simulation

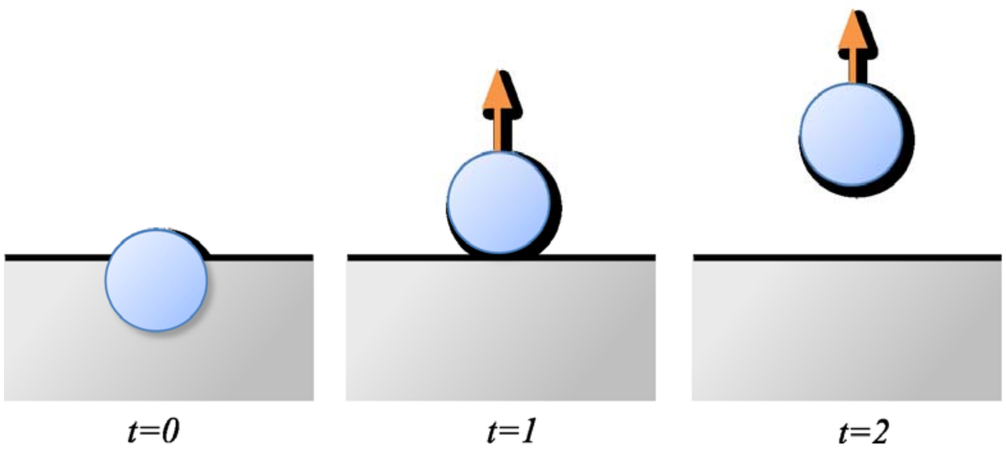
Rigid bodies are constructed of particles placed in a cell-like structure within a rigid mesh.

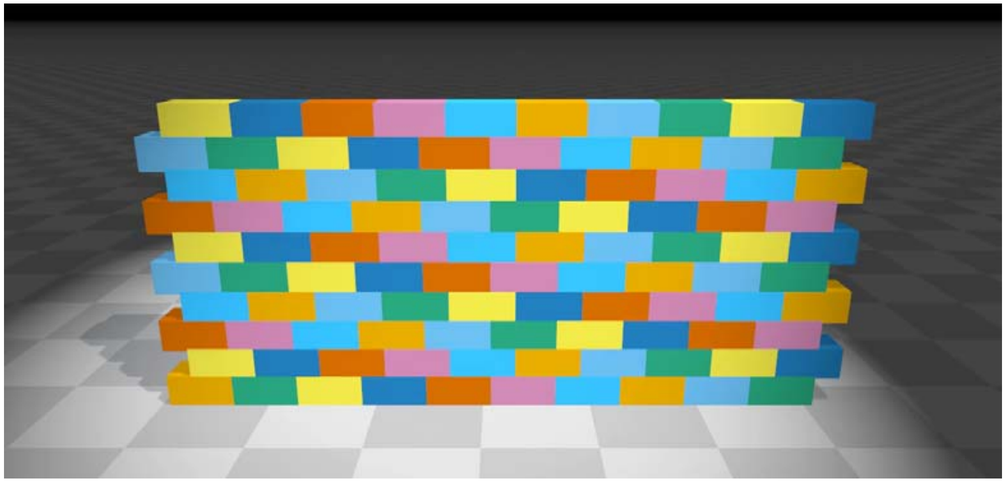
Collisions are handled, as usual, however, they always map back to their original state afterwards.

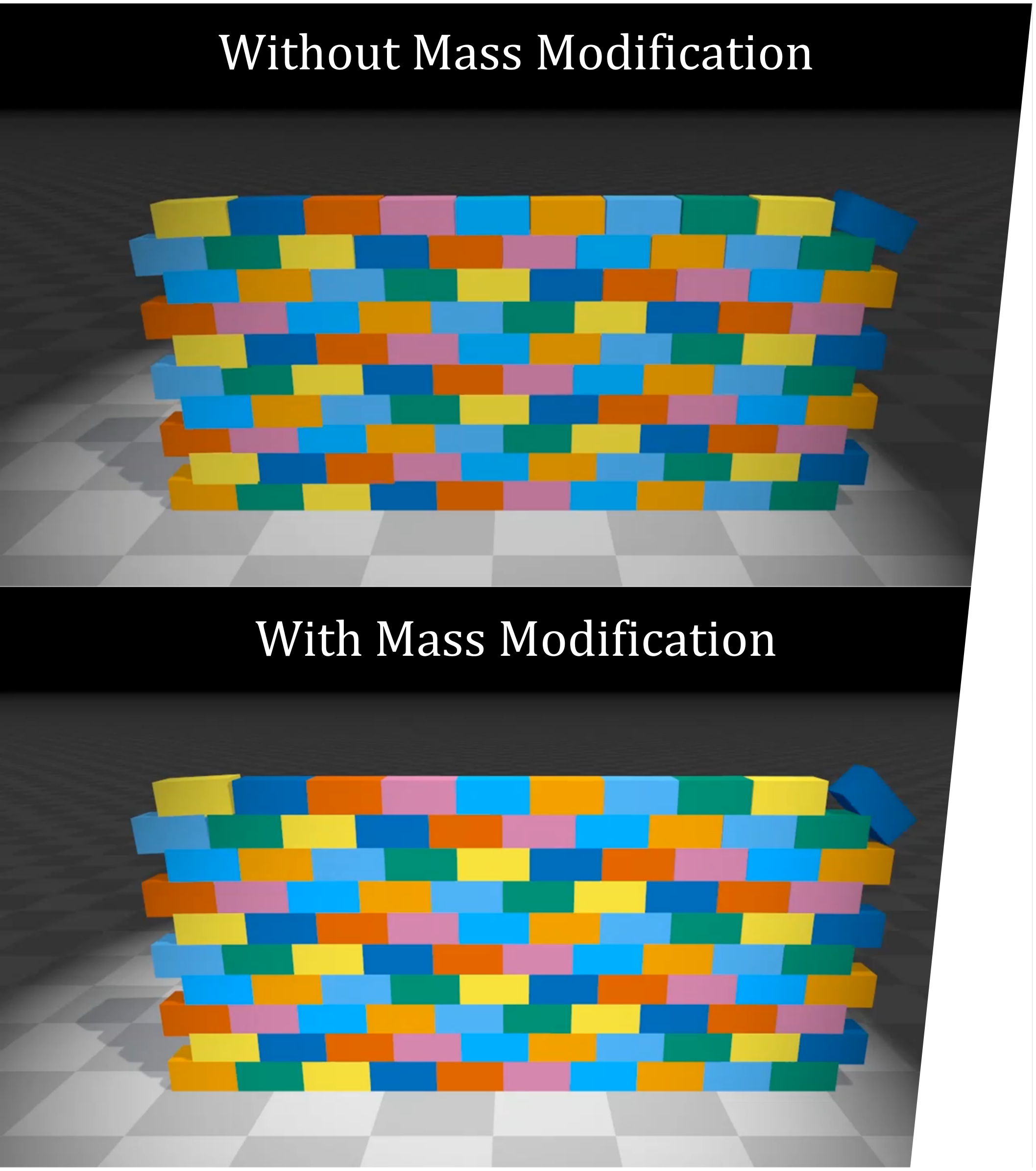


Problems with The Rigid Body model

There are problems with this method for modelling Rigid Bodies
It is not easy to stack objects without them deforming
Commonly shock propagation is used to fix each layer iteratively
This is not ideal when working with a parallelised system.





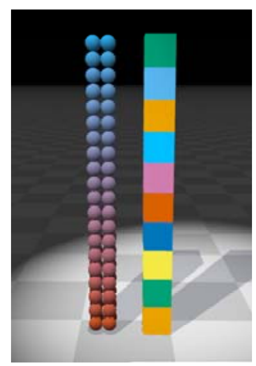


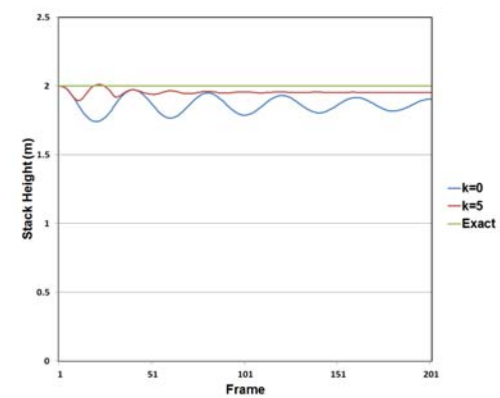
The Solution

This issue is solved instead by altering the mass of each particle.

This puts less pressure on lower objects, causing them to deform less.

Particles can still be processed in parallel, so performance is not affected.



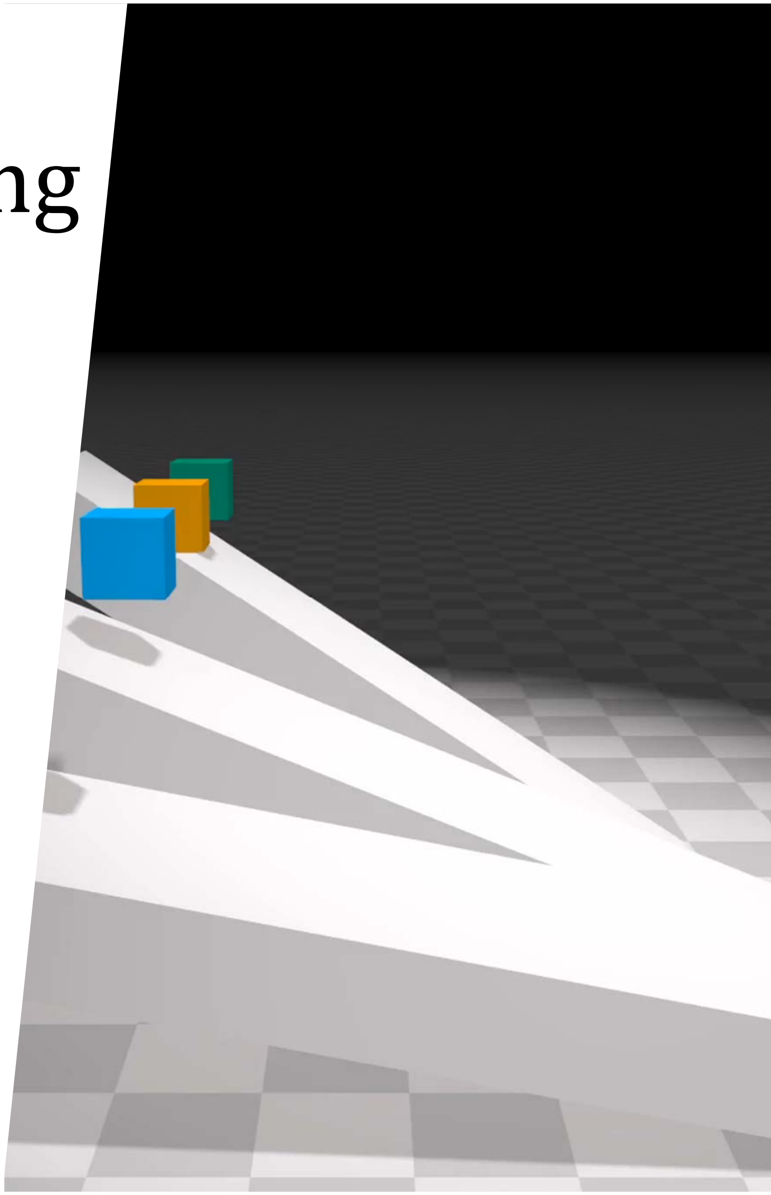


Friction and Collision Handling

Two types of collision:

- Particle to Particle
- Particle to Mesh.

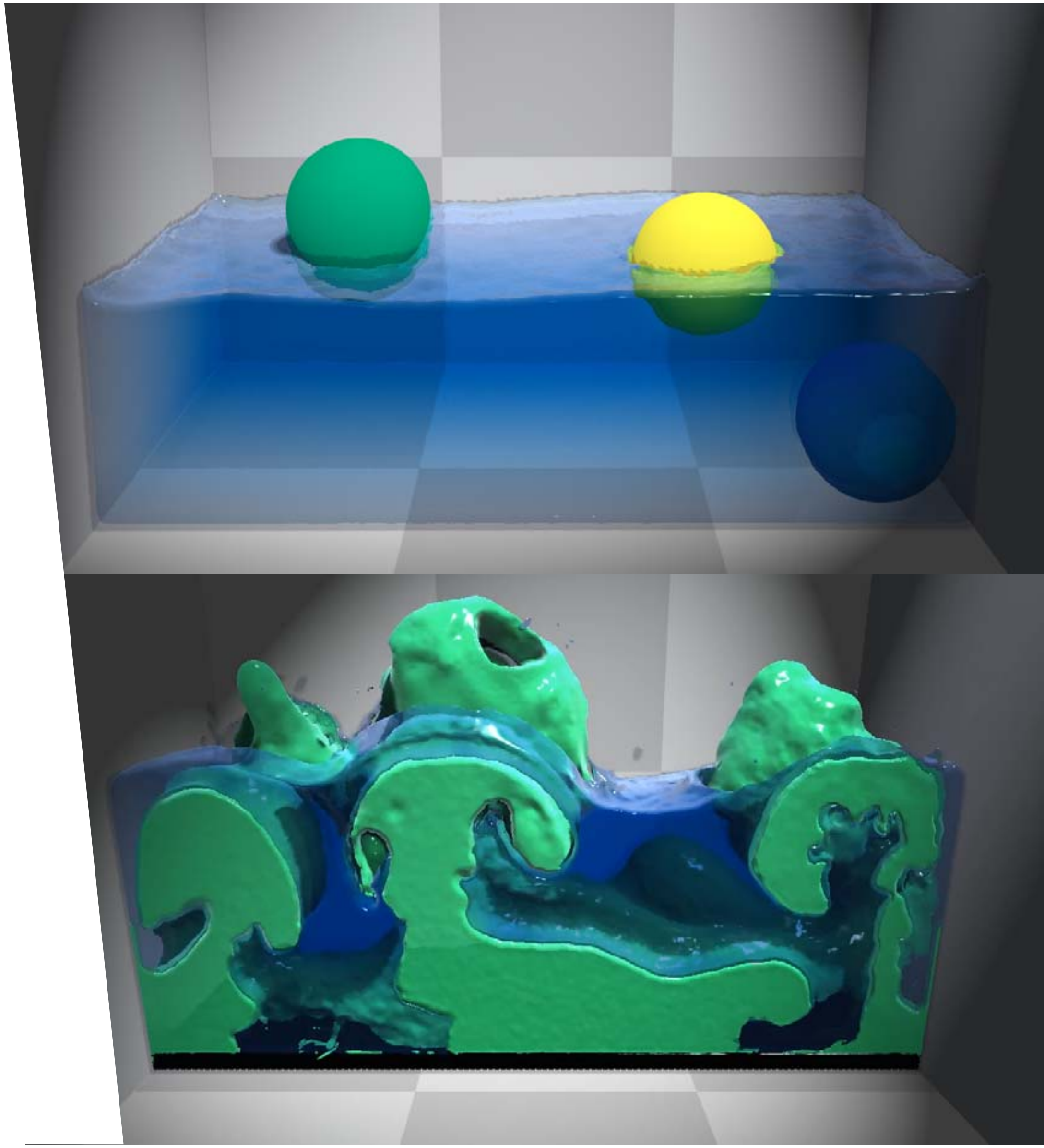
Friction is handled by adding a constraint to each particle limiting its motion.



Fluids and Gases

Liquids and Gases are simulated using a density constraint.

This method can easily simulate effects like fluid separation and floating.



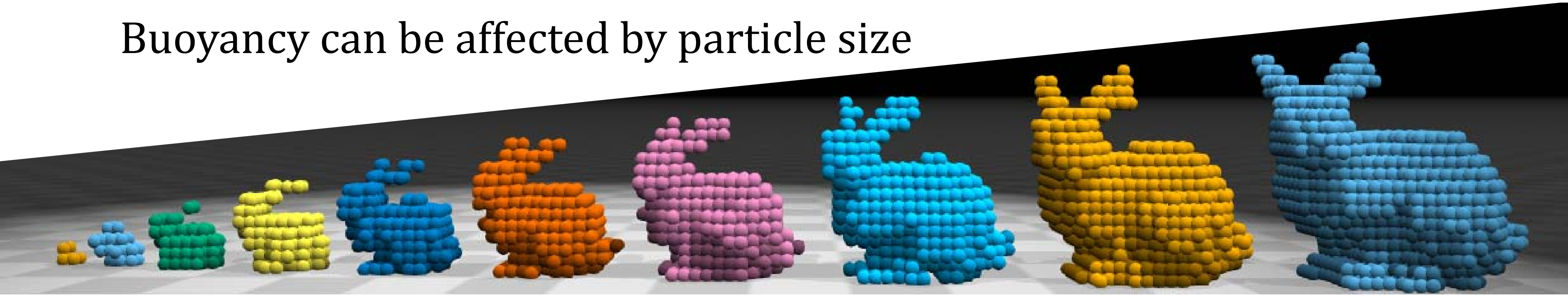
Disadvantages and Limitations

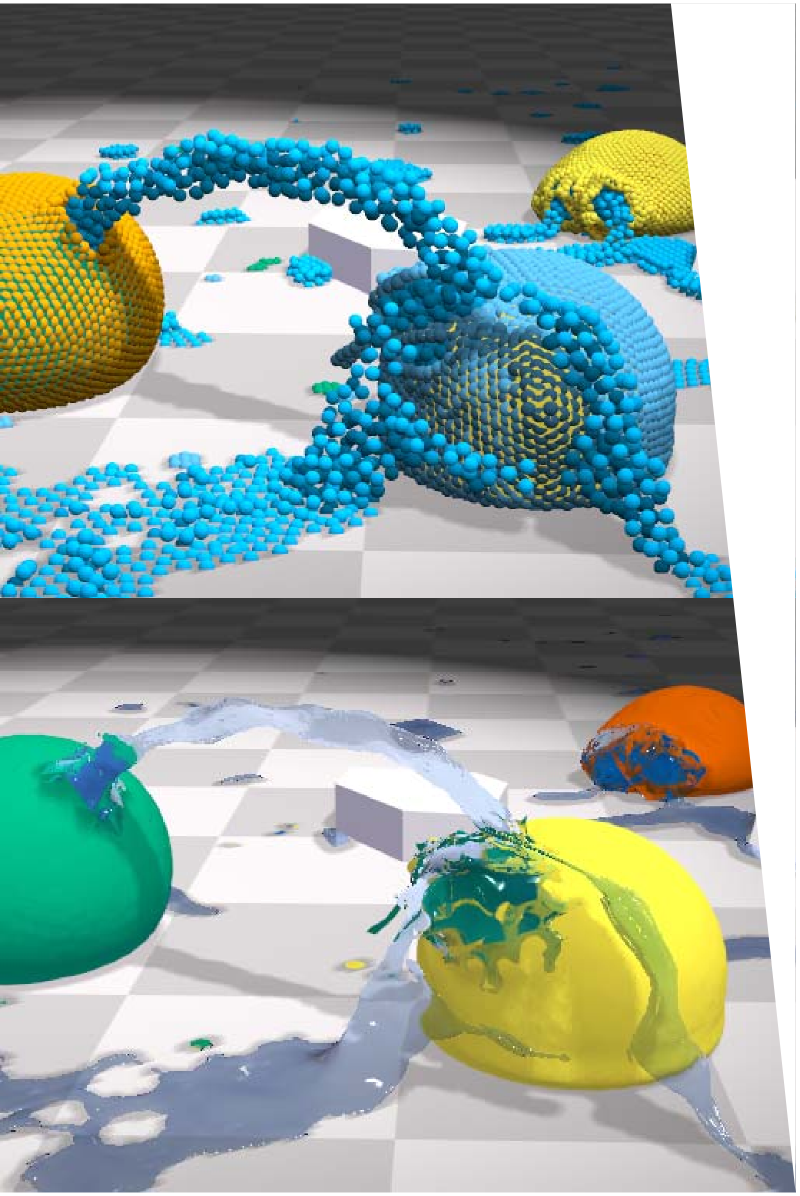
Particles are all the same size. Without this, it can affect collision detection

There is a limit between the smallest and largest objects before they either have too many particles, or there are not enough particles to represent the object correctly.

Simulation is suited for smaller objects, so larger objects like walls and floors must be rendered using alternative methods.

Buoyancy can be affected by particle size



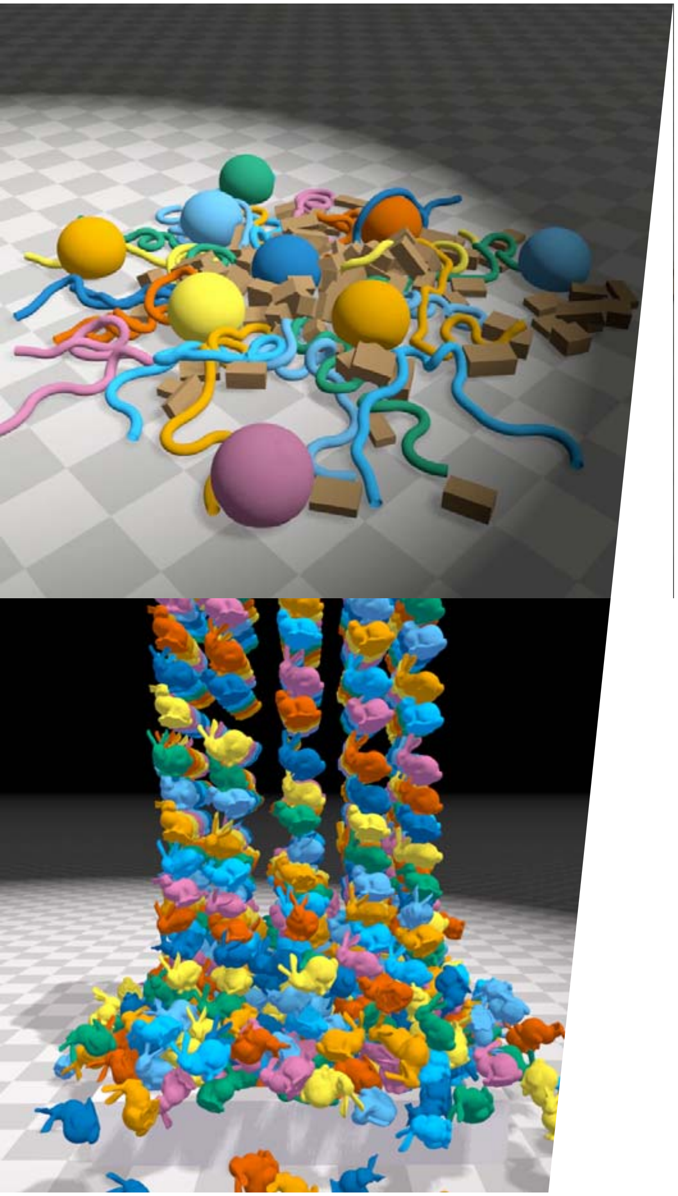


Future Work

First step is to allow particles to be different sizes. This makes simulation more efficient.

Improve collision detection for large flat surfaces.

Allow variable frequencies for different object types to also improve efficiency.



Conclusion

Powerful system for simulating small objects that performs well in real-time.

It can simulate many different object types

References

[1] M. Macklin, M. Müller, N. Chentanez, and T.-Y. Kim, ‘Unified particle physics for real-time applications’, ACM Trans. Graph., vol. 33, no. 4, pp. 1–12, Jul. 2014, doi: 10.1145/2601097.2601152.

[2] NVIDIA, ‘NVIDIA FleX’, NVIDIA Developer, Aug. 13, 2015. https://developer.nvidia.com/flex (accessed Aug. 19, 2020).

[3] JEGX, ‘NVIDIA FLEX (Unified GPU PhysX) and Flame Works: New Tools for Game Developers’, Geeks 3D, Oct. 17, 2013. https://www.geeks3d.com/20131017/nvidia-flex-unified-gpu-physx-flame-works-gi-works-new-tools-for-game-developers/ (accessed Aug. 19, 2020).

[4] M. Macklin and M. Müller, ‘Position based fluids’, ACM Trans. Graph., vol. 32, no. 4, p. 1, Jul. 2013, doi: 10.1145/2461912.2461984.

[5] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff, ‘Position based dynamics’, J. Vis. Commun. Image Represent., vol. 18, no. 2, pp. 109–118, Apr. 2007, doi: 10.1016/j.jvcir.2007.01.005.

[6] Autodesk, ‘Nucleus in Autodesk Maya White Paper’, 2009. [Online]. Available: https://autodeskresearch.com/sites/default/files/autodeskmaya_nucleus_whitepaper.pdf.

[7] M. Macklin, M. Müller, N. Chentanez, and T.-Y. Kim, ‘Unified particle physics for real-time applications (SIGGRAPH 2014 Presentation)’, Jul. 27, 2014, Accessed: Aug. 02, 2020. [Online]. Available: http://blog.mmacklin.com/project/flex/.

[8] M. Macklin, ‘Unified Particle Physics for Real-Time Applications SIGGRAPH submission video.’, 2014, Accessed: Aug. 19, 2020. [Online]. Available: http://mmacklin.com/uppfrta.mp4.

Videos sourced from [8]. Images sourced from [1],[7]. All videos were used with permission from M. Macklin.