Material Point Methods in the Browser

Jacky Lu, Henry Yang, Chetan Parthiban

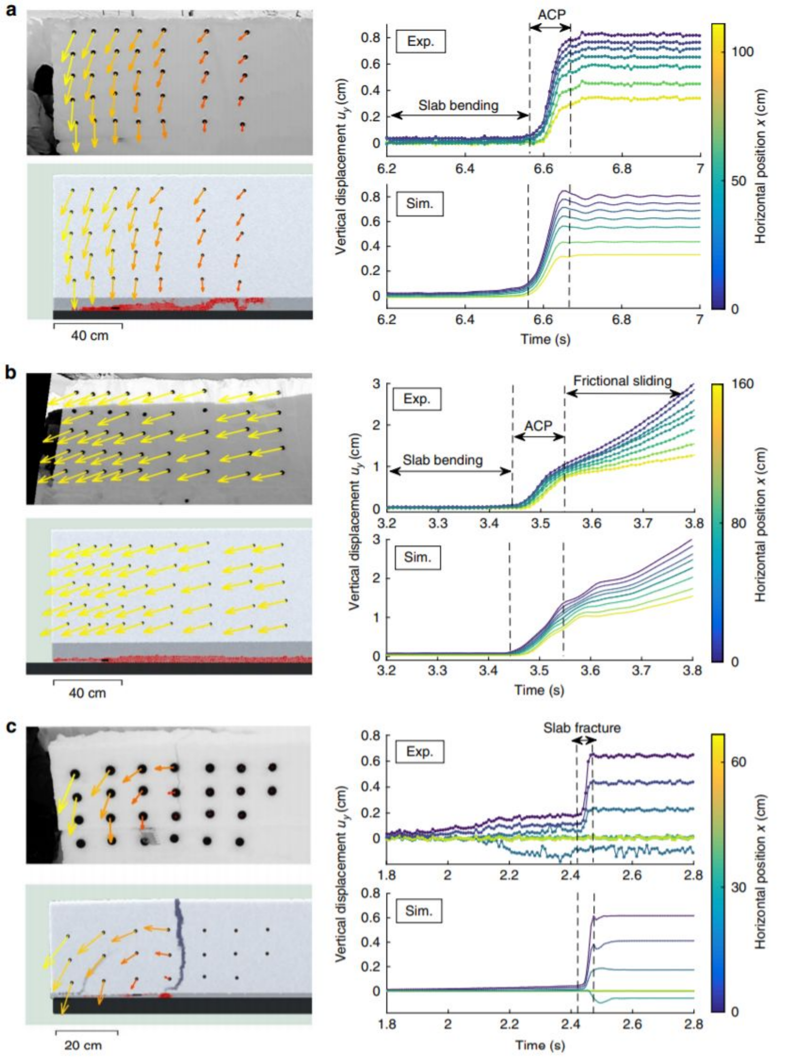
see this presentation with gifs at: https://docs.google.com/presentation/d/1AmkR8h1SU_69glhijSGEYvMLTdvV20eTBSlzSBuYG60/edit?usp=sharing

Material Point Method

A Hybrid Lagrangian-Eulerian Method

Advantages:

- *Physically Accurate*
- *Multi-Phase Interactions*
- *Large Deformation Handling*
- *Automatic Collision Handling*



Gaume, J., Gast, T., Teran, J. *et al.* Dynamic anticrack propagation in snow. *Nat Commun* 9, 3047 (2018). https://doi.org/10.1038/s41467-018-05181-w

Computational Tasks

~200,000 *Particles*

~130,000 *Grid Nodes*

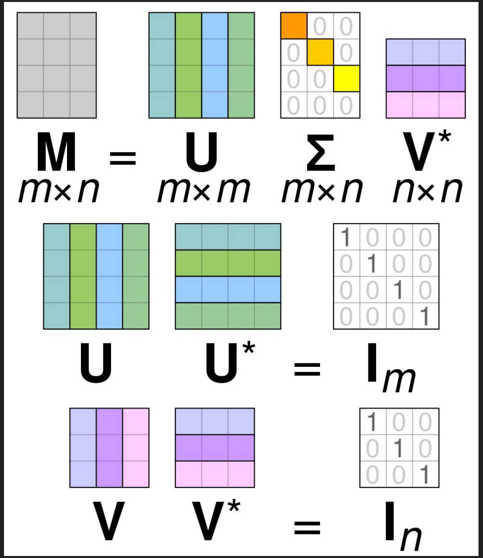
2X Particle & Grid Information Transfers

Physically Accurate Force Computation

Physically Accurate Material Attribute Update

40X Iterations Per 24Hz Frame For EVERYTHING!

4 Million Triangles!

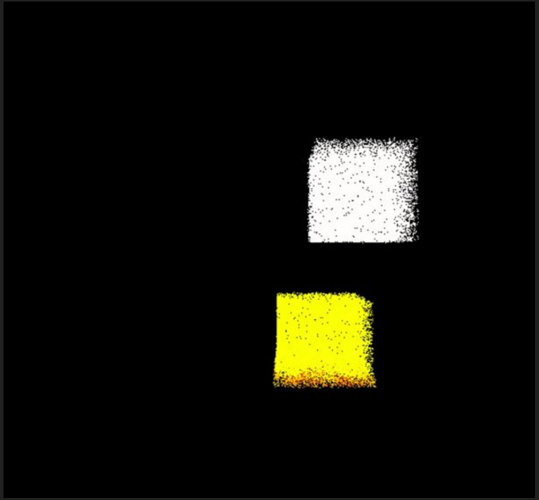


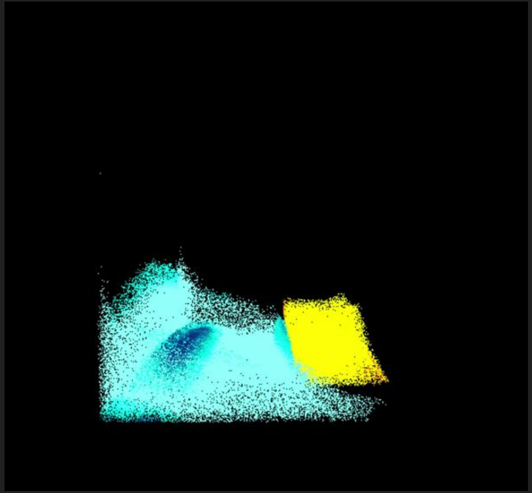
$\underset{m\times n}{\mathbf{M}} = \underset{m\times m}{\mathbf{U}} \ \underset{m\times n}{\mathbf{\Sigma}} \ \underset{n\times n}{\mathbf{V}^*}$
$\mathbf{U} \ \mathbf{U}^* = \mathbf{I}_m$
$\mathbf{V} \ \mathbf{V}^* = \mathbf{I}_n$

WebGL™ → WebGPU

Simulator Features

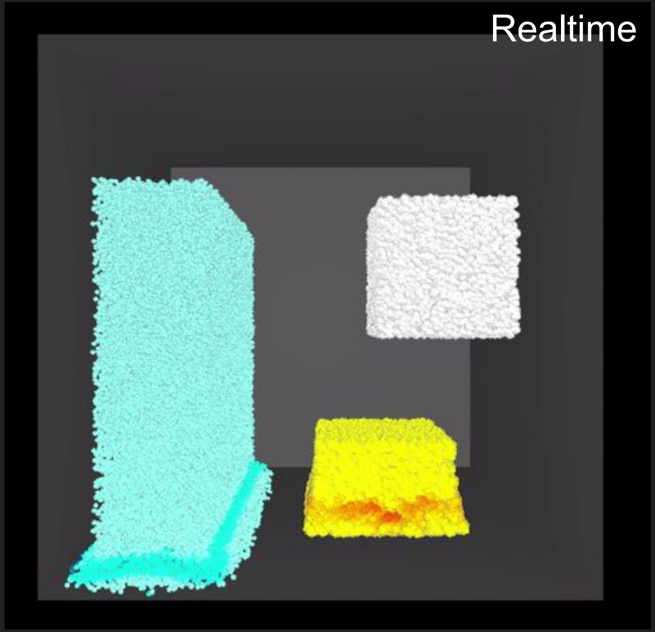
- Runs in the browser in near real-time w/ 200k particles
  - Of course this requires a suitable powerful GPU on the device
- Handles 3 different materials (Snow, Jello, Fluid)





Simulator Features

- Utilizes instancing to allow for lambertian shading of materials
- Easy to create and customize scenes

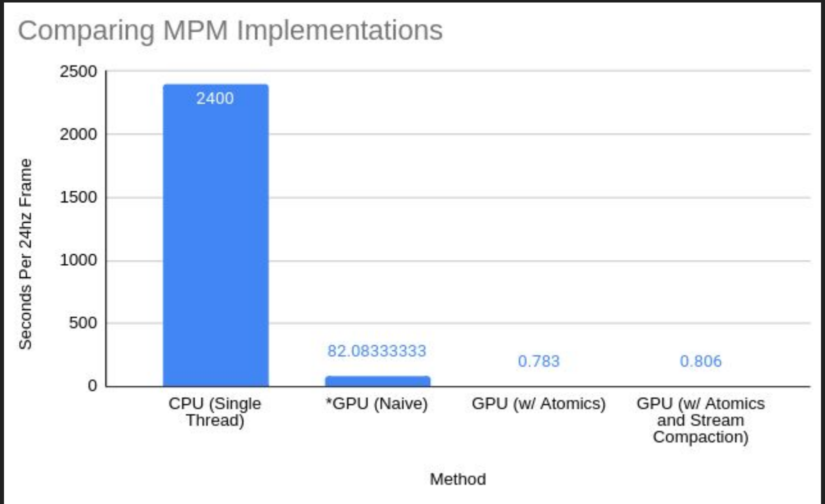


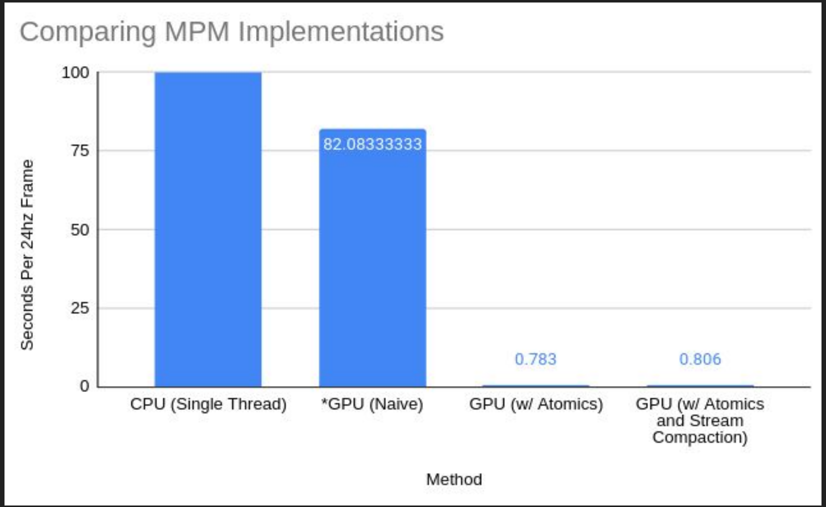
GIFs

GIFs

LIVE DEMO!!!!!!

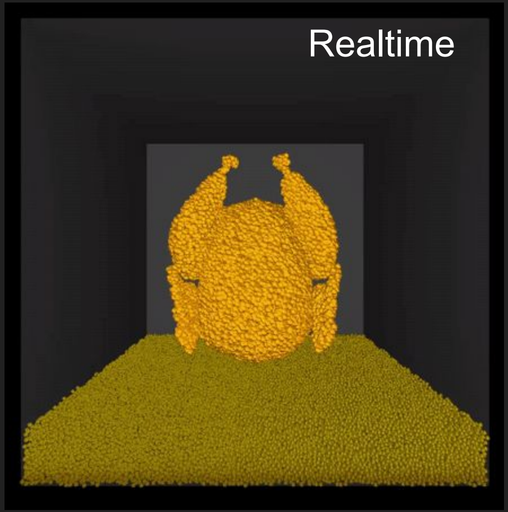
Comparing Implementation Performance





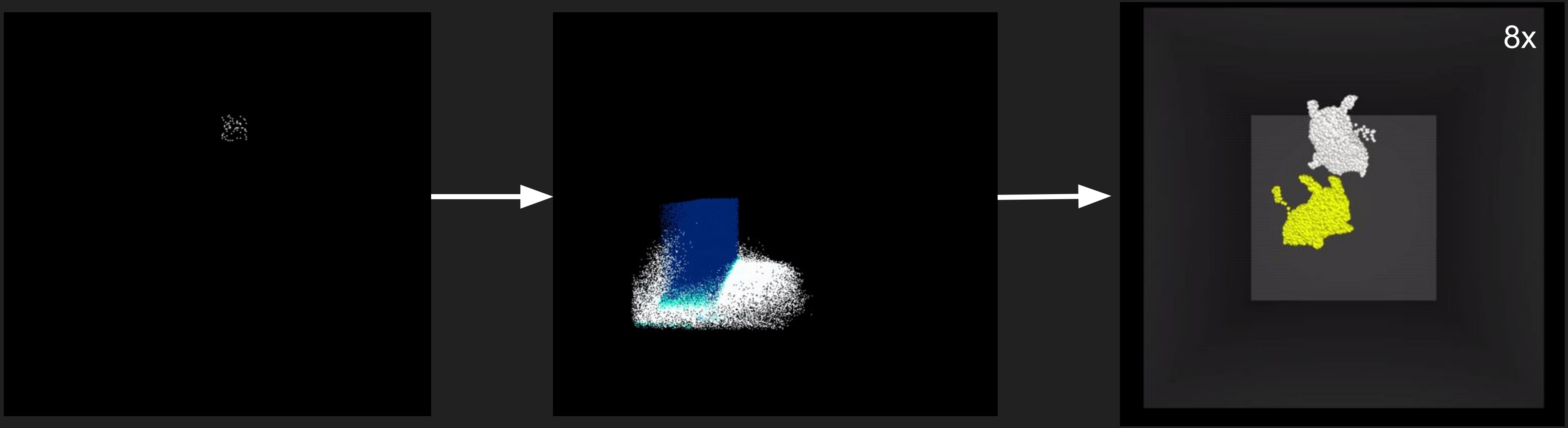
Shortcomings

- Because we utilize atomics, our program is slowed down significantly due to write conflicts
- Creating models for objects with “thin” geometry does not work well
- Limited to lower number of particles due to timeout safety features



Special Thanks To Kai, Austin, Dr. Jiang, and the WebGPU Team!!!

Thanks for watching… Questions?



You can try our demo out right now using Chrome Canary at these webpages:

chetanp.io/WebGPUMPM
jackylu0124.github.io/WebGPUMPM

Do you have more questions?

Here's how you can contact us!

- Jacky Lu
  - jackylu0124@gmail.com
  - https://www.linkedin.com/in/jacky-lu-506968129/
- Chetan Parthiban
  - cparthiban08@gmail.com
  - https://www.linkedin.com/in/chetan-parthiban/
- Haorong (Henry) Yang
  - haorongy@seas.upenn.edu
  - https://www.linkedin.com/haorong-henry-yang

Credits

1. https://gpuweb.github.io/gpuweb/
2. Ming Gao, Xinlei Wang, Kui Wu, Andre Pradhana, Eftychios Sifakis, Cem Yuksel, and Chenfanfu Jiang. 2018. GPU optimization of material point methods. ACM Trans. Graph. 37, 6, Article 254 (November 2018), 12 pages.
3. GLSL SVD Implementation From @alexsr
4. Joshuah Wolper, Yunuo Chen, Minchen Li, Yu Fang, Ziyin Qu, Jiecong Lu, Meggie Cheng, and Chenfanfu Jiang. 2020. AnisoMPM: Animating Anisotropic Damage Mechanics. ACM Trans. Graph. 39, 4, Article 37 (July 2020), 16 pages.
5. Gaume, J., Gast, T., Teran, J. et al. Dynamic anticrack propagation in snow. Nat Commun 9, 3047 (2018).
6. Stomakhin, A., Schroeder, C., Chai, L., Teran, J., Selle, A. 2013. A Material Point Method for Snow Simulation. ACM Trans. Graph. 32, 4, Article 102 (July 2013), 12 pages.
7. https://en.wikipedia.org/wiki/Material_point_method
8. Chenfanfu Jiang, Craig Schroeder, Joseph Teran, Alexey Stomakhin, and Andrew Selle. 2016. The material point method for simulating continuum materials. In ACM SIGGRAPH 2016 Courses (SIGGRAPH '16). Association for Computing Machinery, New York, NY, USA, Article 24, 1–52.
9. https://developers.google.com/web/updates/2019/08/get-started-with-gpu-compute-on-the-web