Interactive Visual Exploration of Peridynamic-Based Fracture Simulation

Chakrit Watcharopas
joint work with
Joshua A. Levine and Robert M. Geist

18 Mar 2015 @ NVIDIA GTC

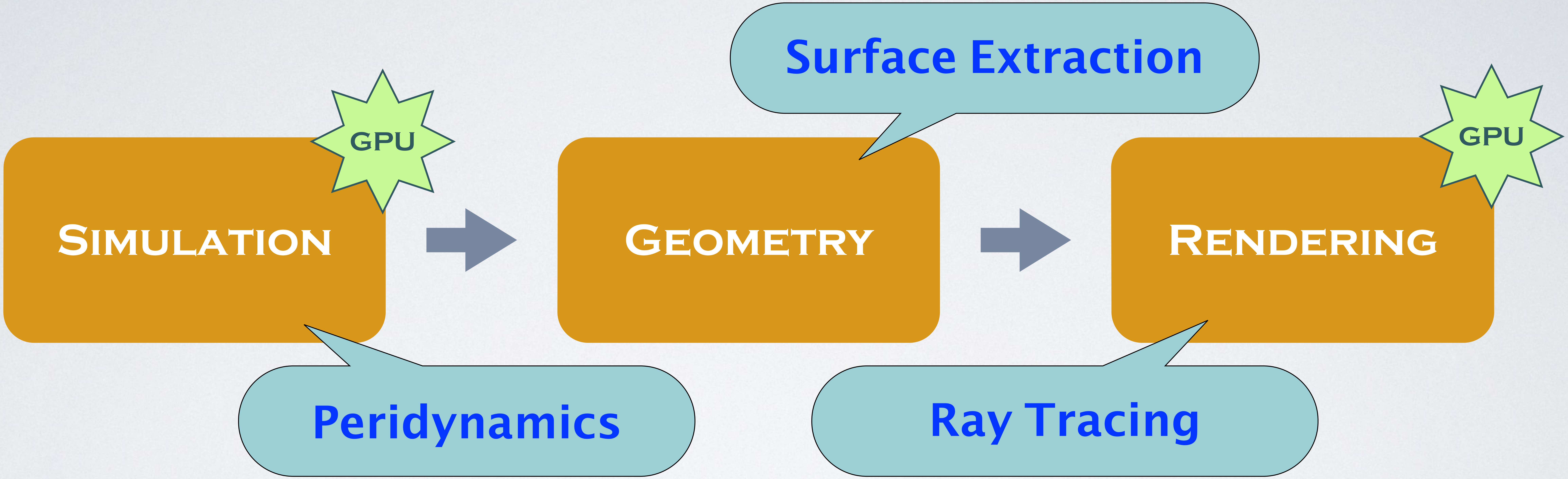
CLEMSON School of COMPUTING
Kasetsart University
NVIDIA CUDA TEACHING CENTER
NVIDIA CUDA RESEARCH CENTER

Our Previous Works

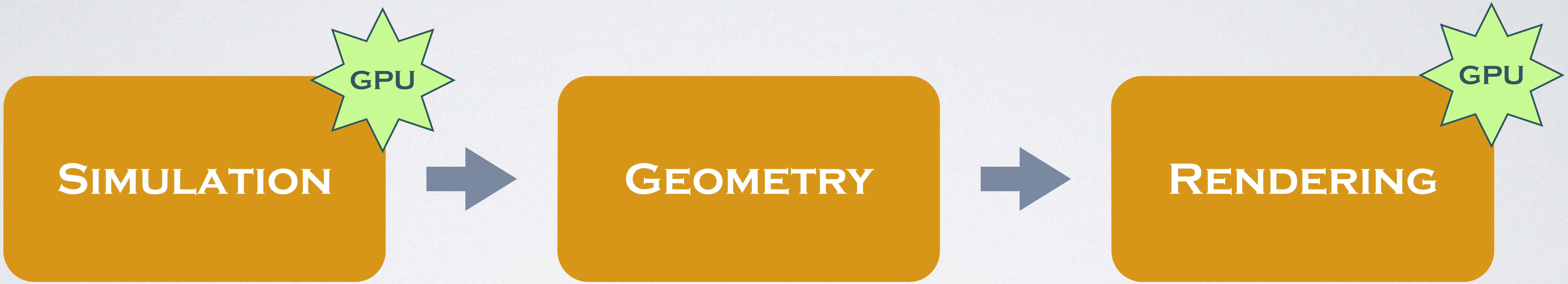
- "*Modeling Fracture on the GPU with Peridynamics*" @ GTC 2014
  - Presented by Joshua Levine
- "*A Peridynamic Perspective on Spring-Mass Fracture*" @ SCA 2014
  - Authored by Joshua Levine, Adam Bargteil, Christopher Corsi, Jerry Tessendorf, and Robert Geist
- Motivation from:
  - Silling S. Journal of Mechanics and Physics of Solids. 48(1): 175-209, 2000.
  - many others…

Vase fractured on the table in slow motion

Our Workflow



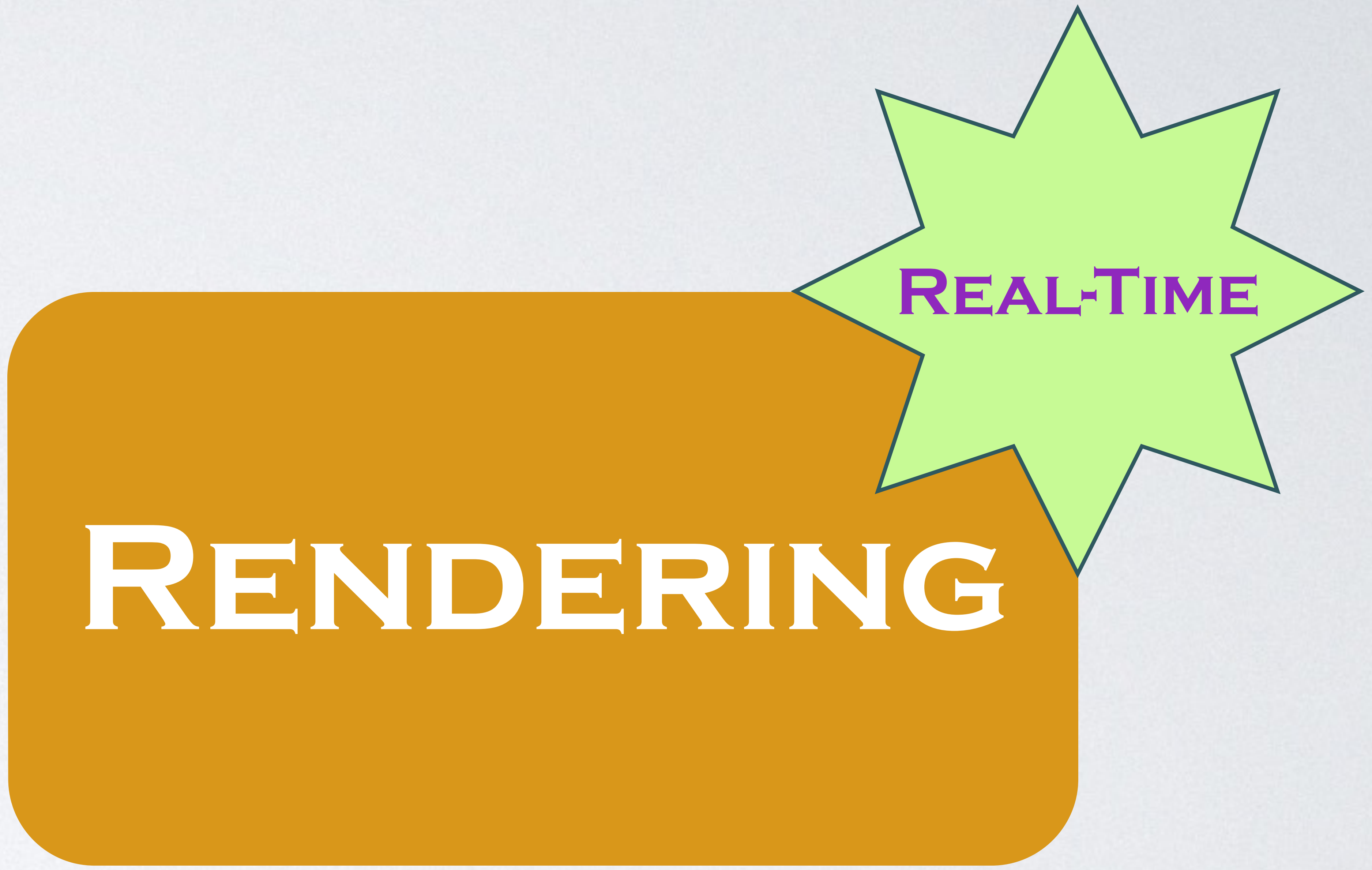
Our Workflow



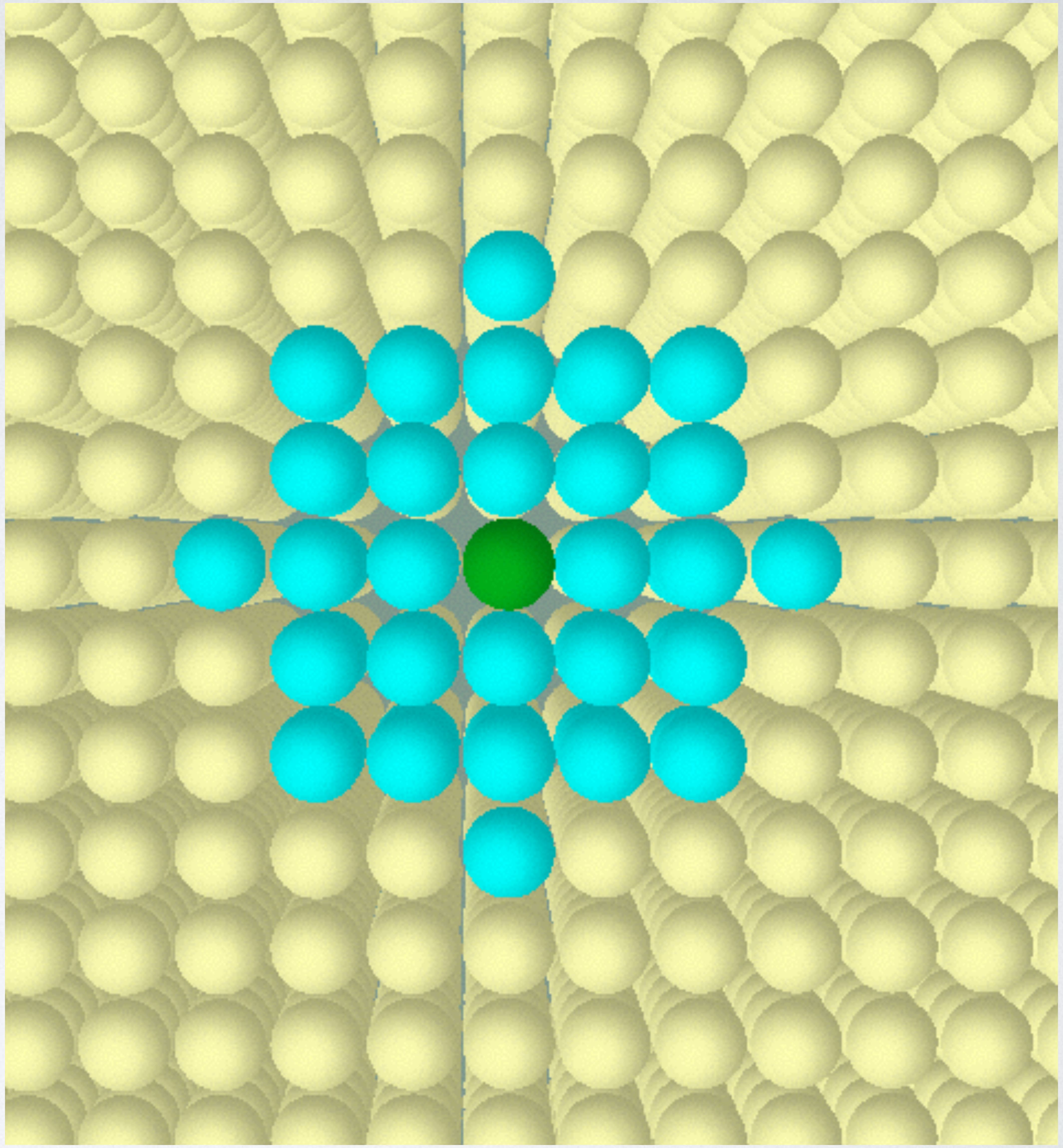
time(simulation) ~ time(rendering) >>>>> time(geometry)

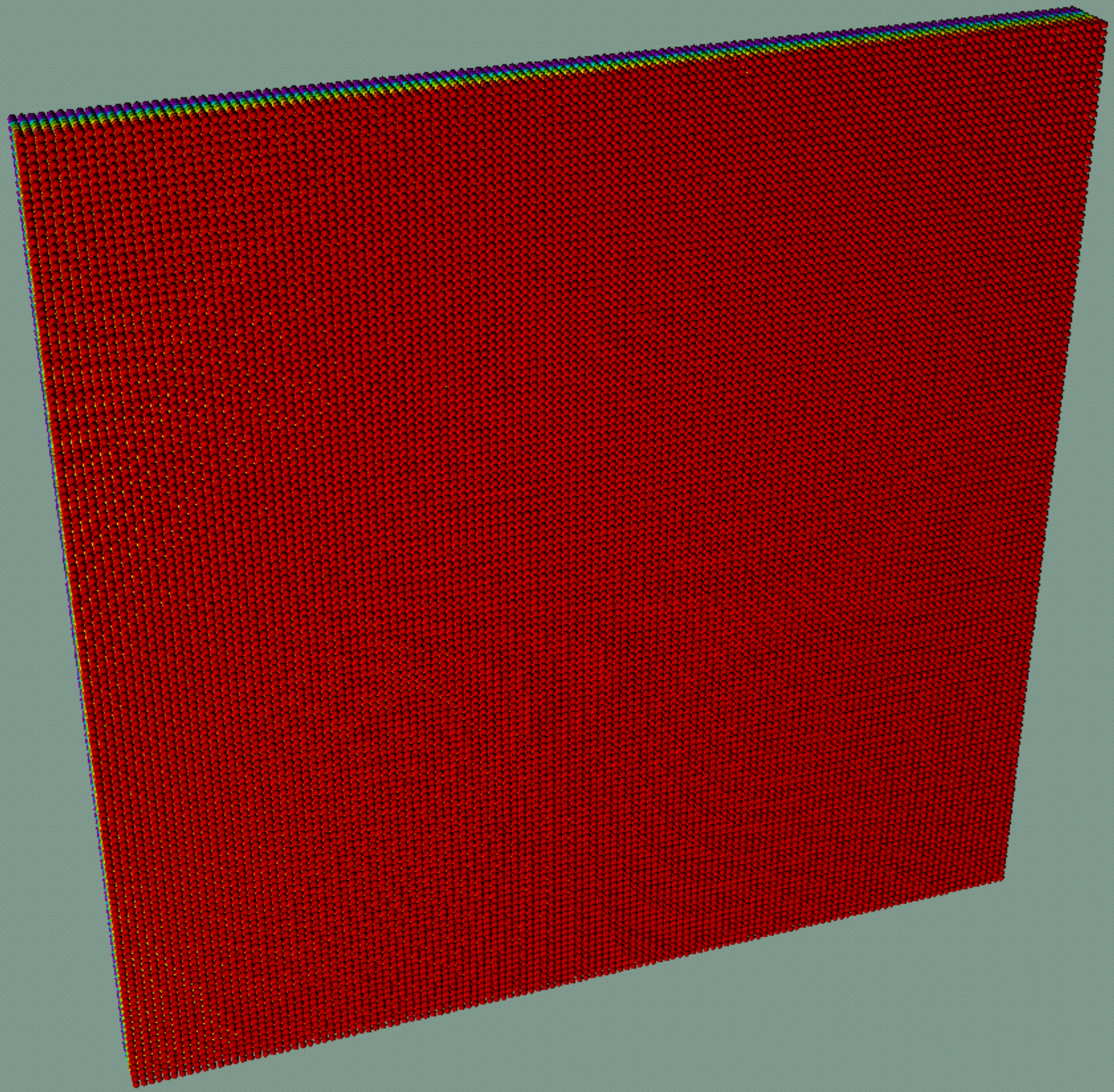
Our Modified Workflow

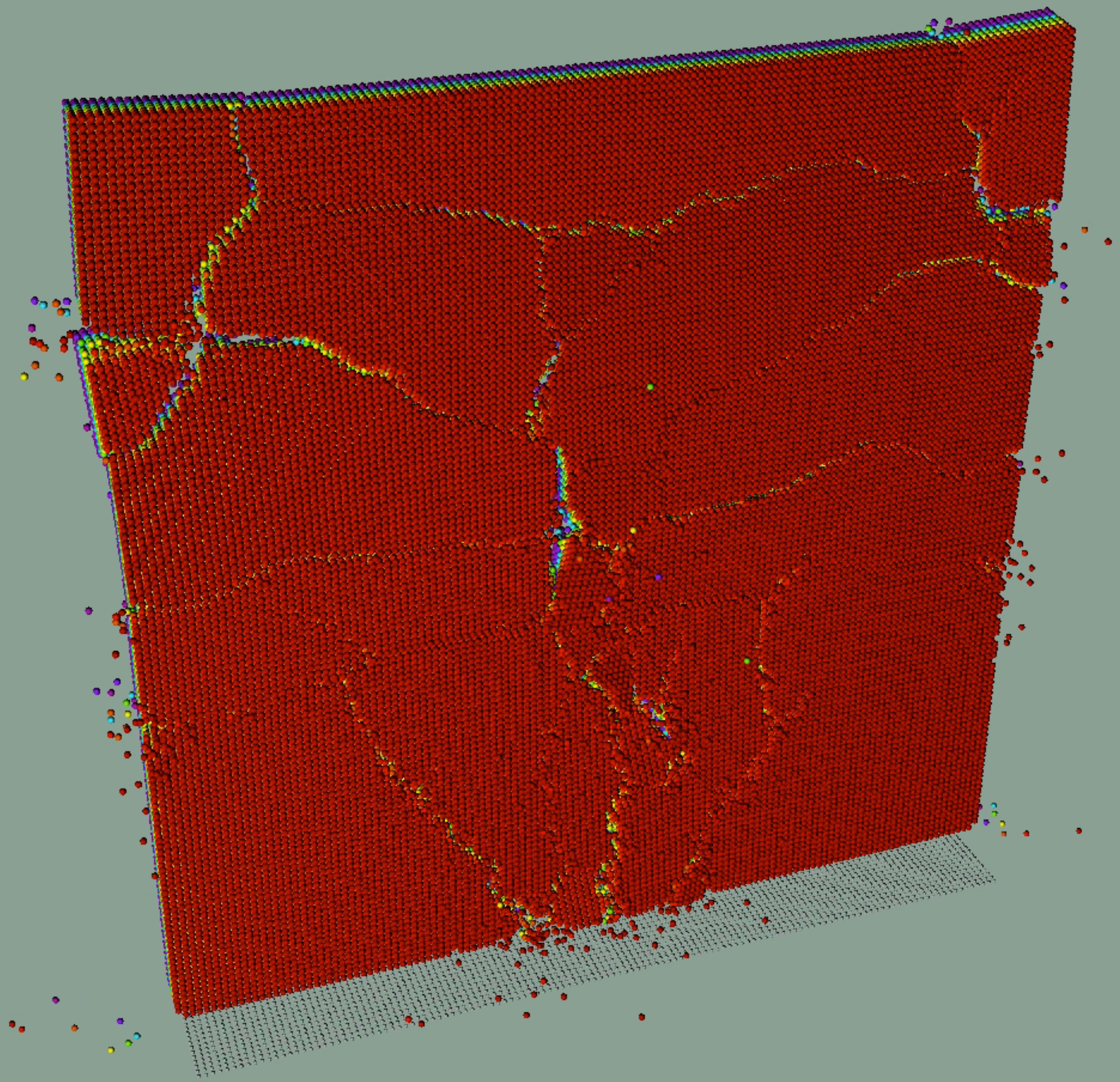
- We make the rendering faster; make it real-time
- We build an Interactive Visual Exploration Tool

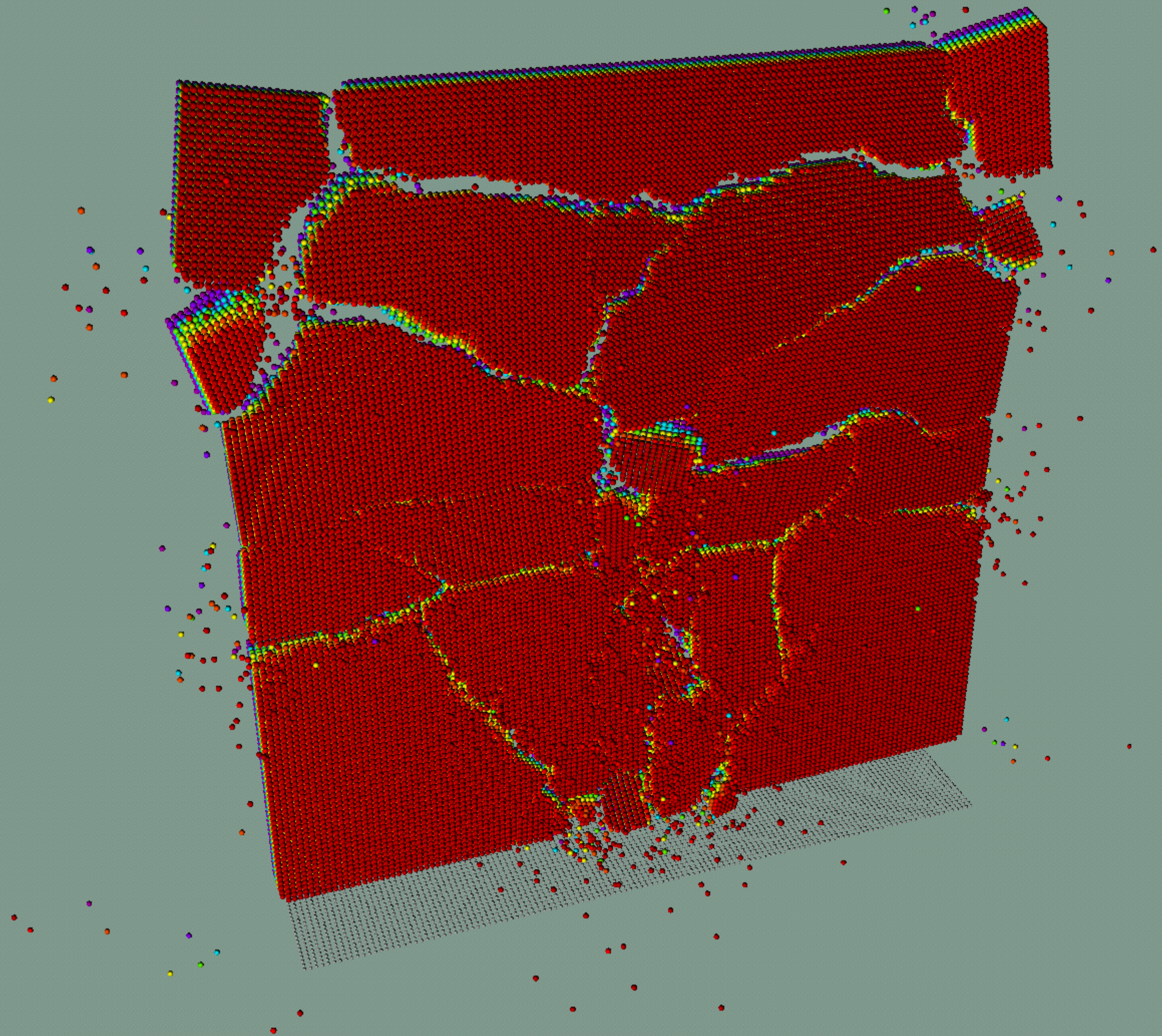


What is Peridynamics?

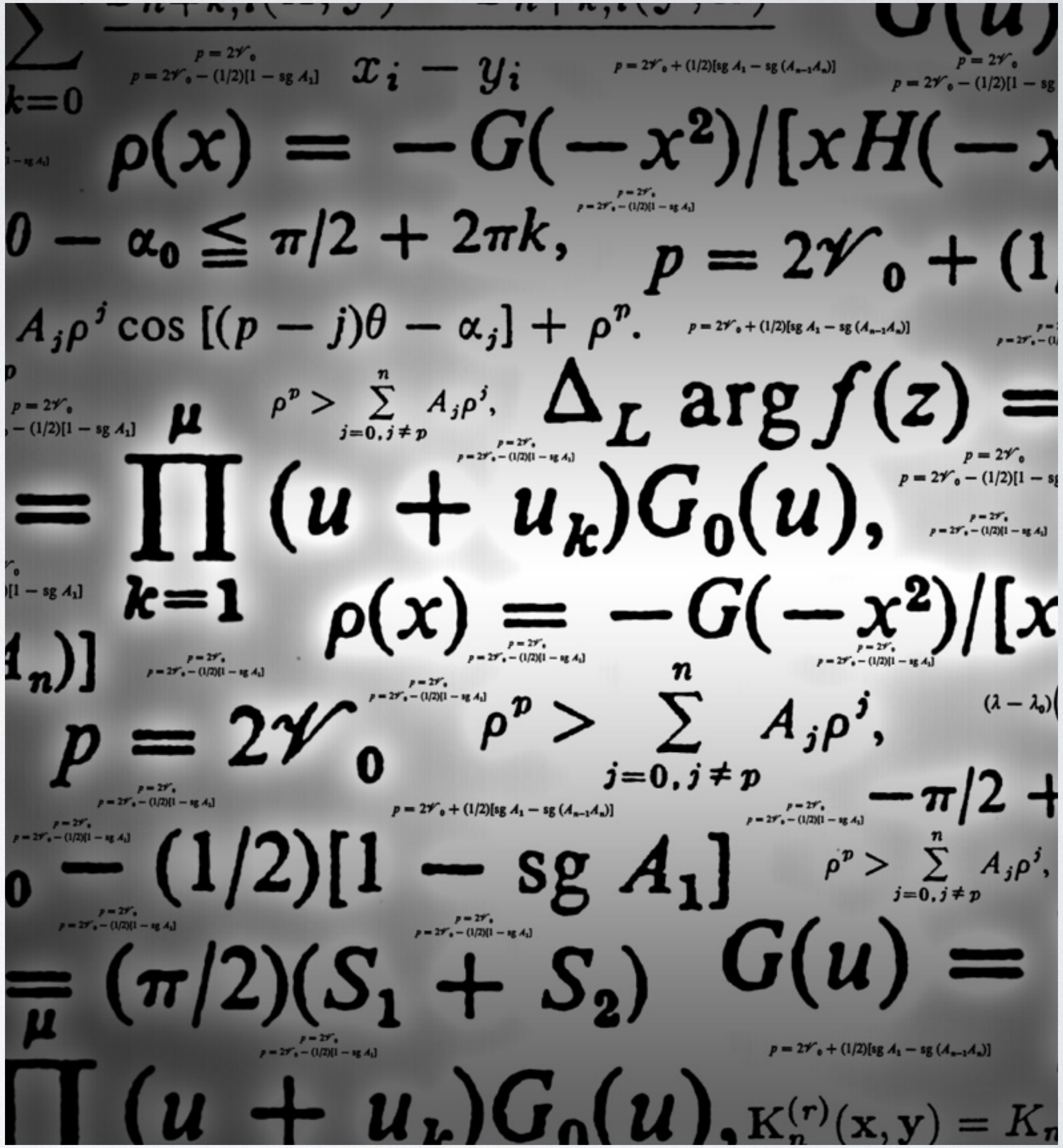








Peridynamic Formula



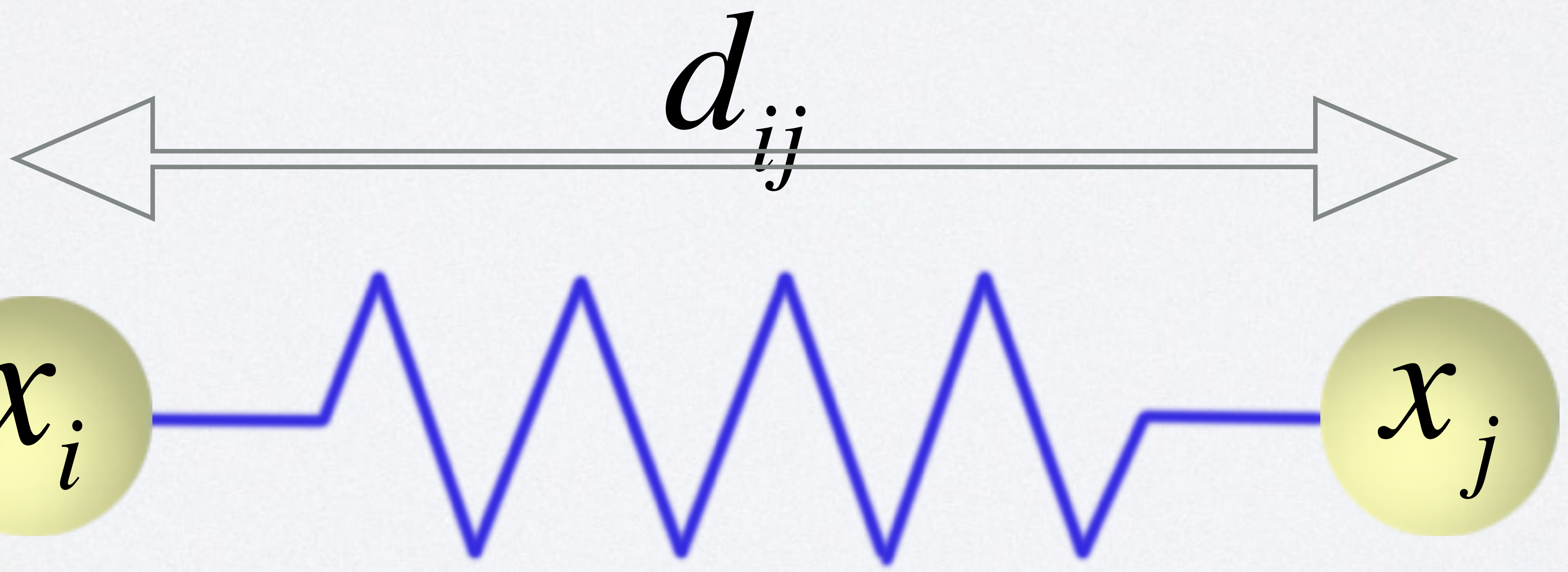
Spring Force

- Traditional spring force exerted on particles i by j

$$f_s = -k_s(\|x_i - x_j\| - d_{ij})\frac{x_i - x_j}{\|x_i - x_j\|}$$

scalar term

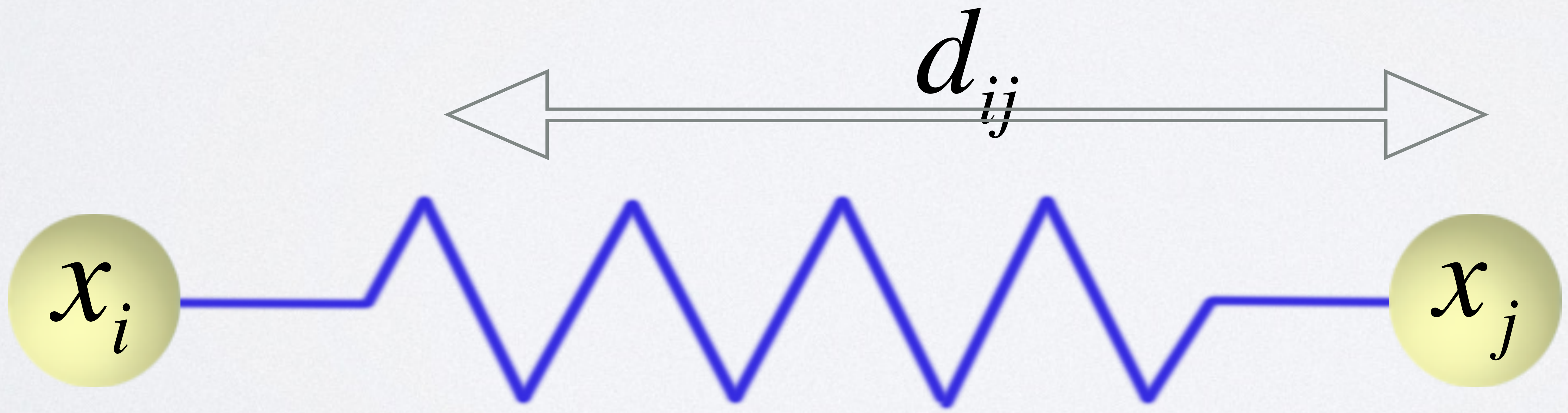
vector



Spring Force

- Traditional spring force exerted on particles i by j

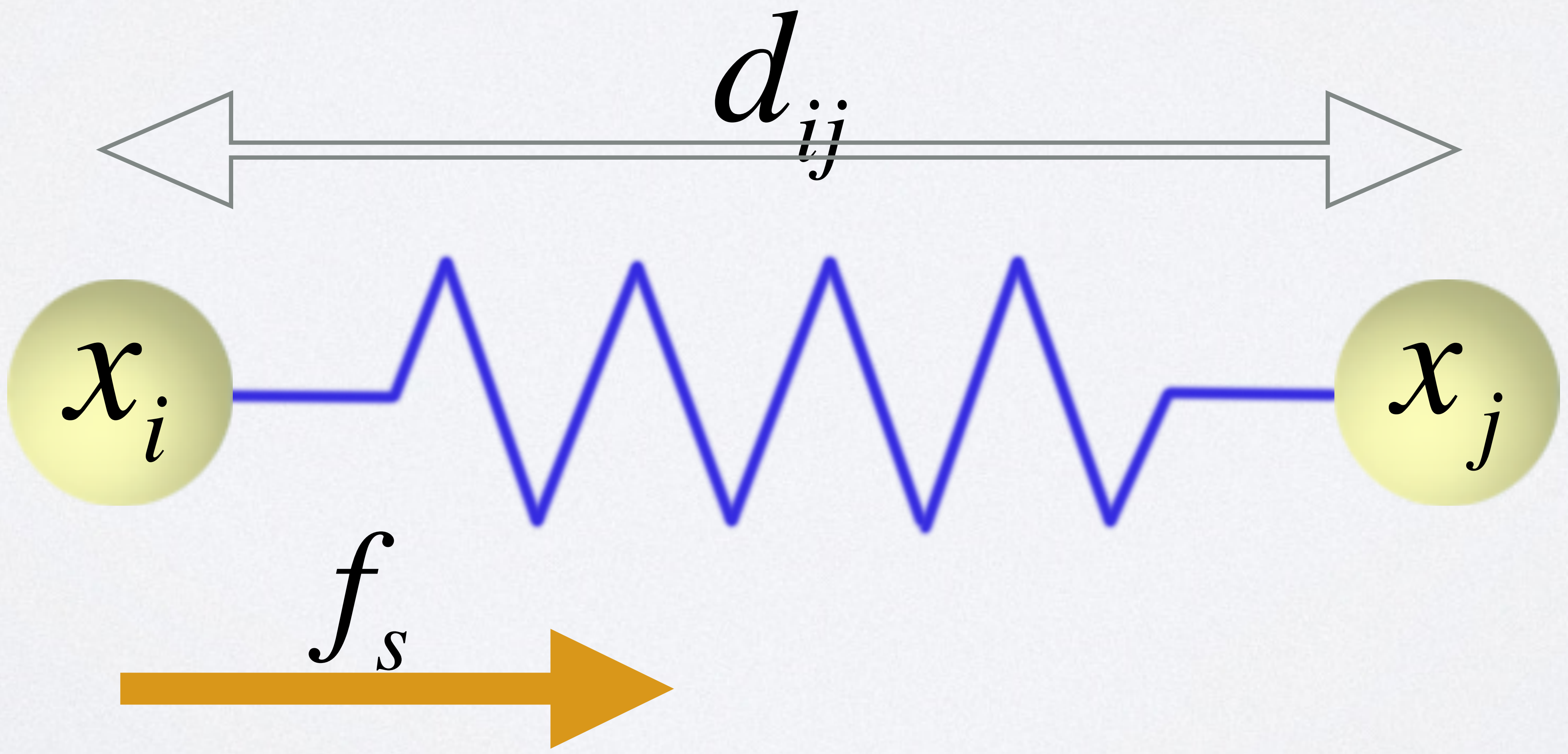
$$f_s = -k_s(\|x_i - x_j\| - d_{ij})\frac{x_i - x_j}{\|x_i - x_j\|}$$



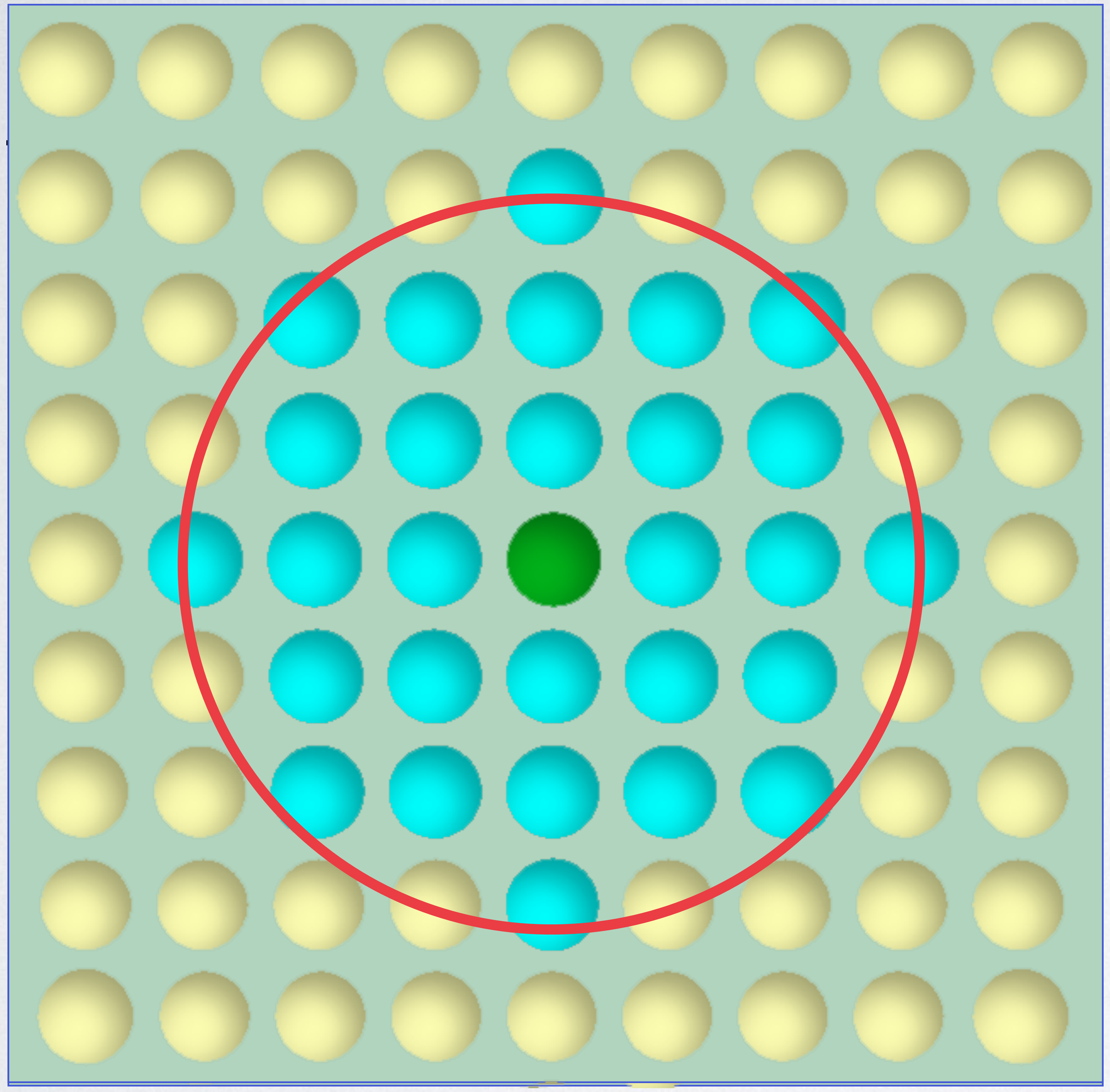
Spring Force

- Traditional spring force exerted on particles i by j

$$f_s = -k_s(\|x_i - x_j\| - d_{ij})\frac{x_i - x_j}{\|x_i - x_j\|}$$



Traditional Spring Force & Peridynamic Force



- Peridynamic force[1,2]

$$f_s = -k_s\,\varepsilon \frac{x_i - x_j}{\left\| x_i - x_j \right\|}$$

$$\varepsilon = \frac{\left\| x_i - x_j \right\| - d_{ij}}{d_{ij}}$$

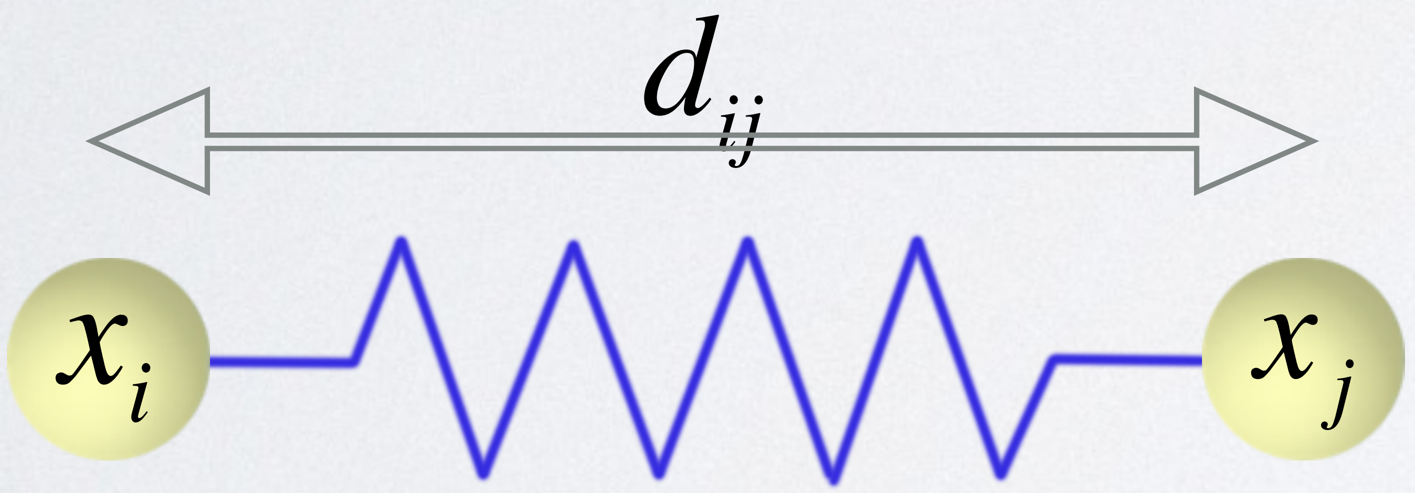
[1] Silling. J. of Mechanics and Physics of Solids. 48(1): 175-209, 2000.

[2] Emmrich, Lehoucq, and Puhst. In Meshfree Methods for Partial Differential Equations VI, vol. 89 of LNCSE, 45–65, 2013.

Traditional Spring Force & Peridynamic Force

- Traditional spring force

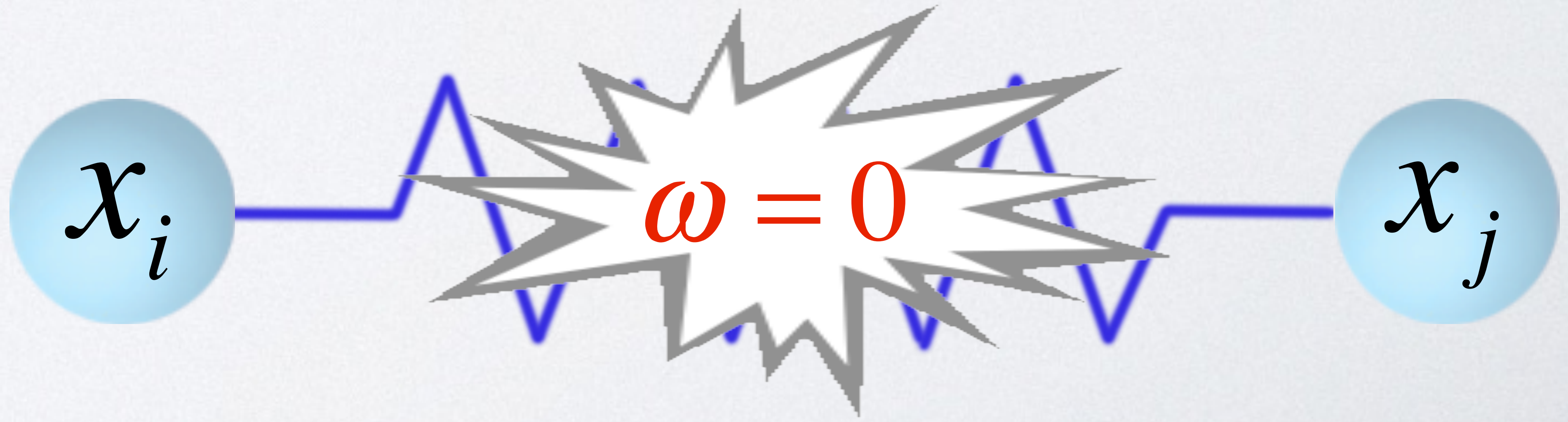
$$f_s = -k_s(\|x_i - x_j\| - d_{ij})\frac{x_i - x_j}{\|x_i - x_j\|}$$



- Peridynamic force[1,2]

$$f_s = -k_s\omega\,\varepsilon\frac{x_i - x_j}{\|x_i - x_j\|}$$

$$\varepsilon = \frac{\|x_i - x_j\| - d_{ij}}{d_{ij}}$$

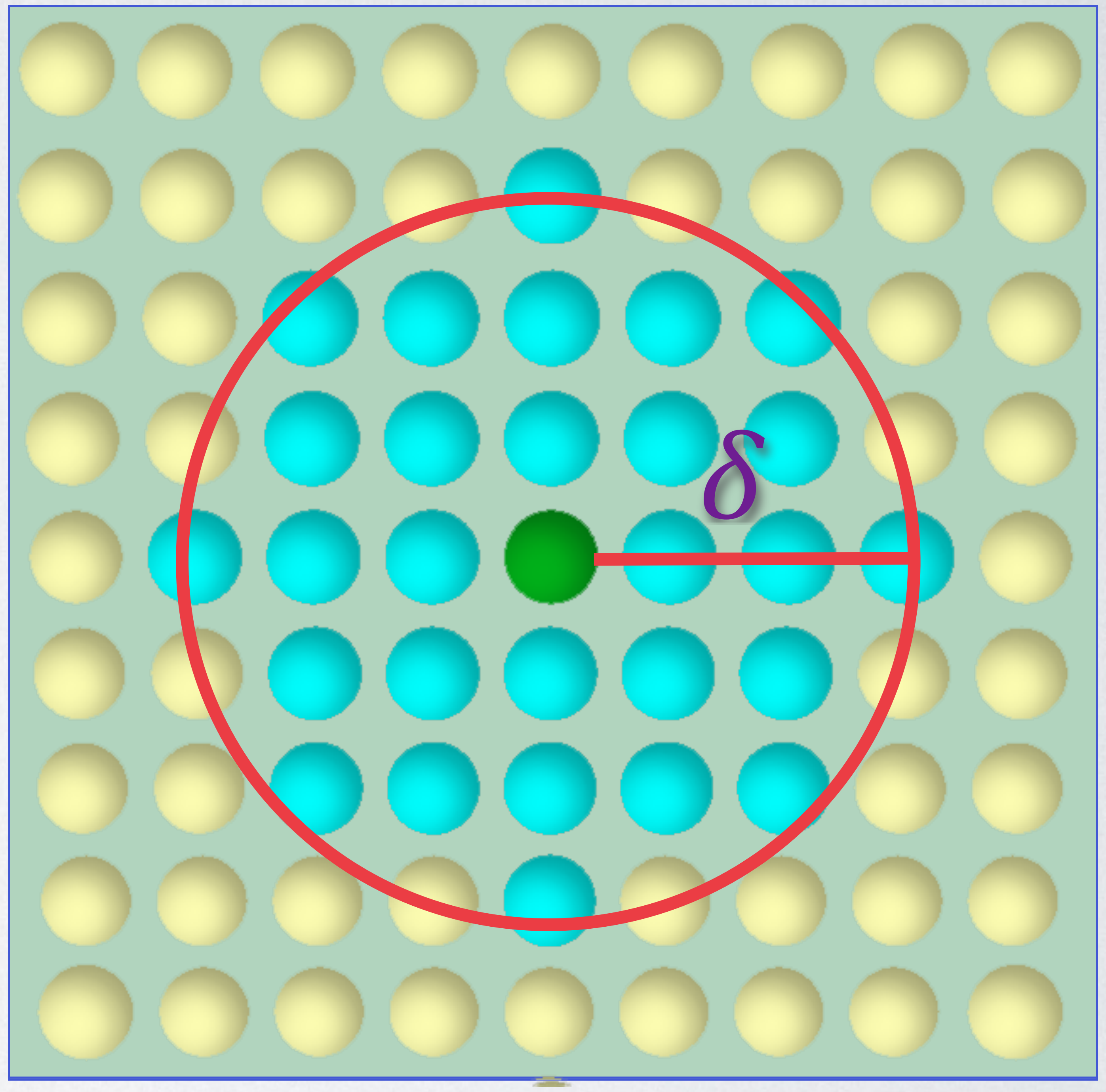


[1] Silling. J. of Mechanics and Physics of Solids. 48(1): 175-209, 2000.

[2] Emmrich, Lehoucq, and Puhst. In Meshfree Methods for Partial Differential Equations VI, vol. 89 of LNCSE, 45–65, 2013.

Peridynamic Particles

- Particles within radius δ are initially bonded
- When a bond stretches too far, the bond breaks
  - its strain ε exceeds the strain limit τ
  - $\omega \to 0$



Particle-Particle Collision

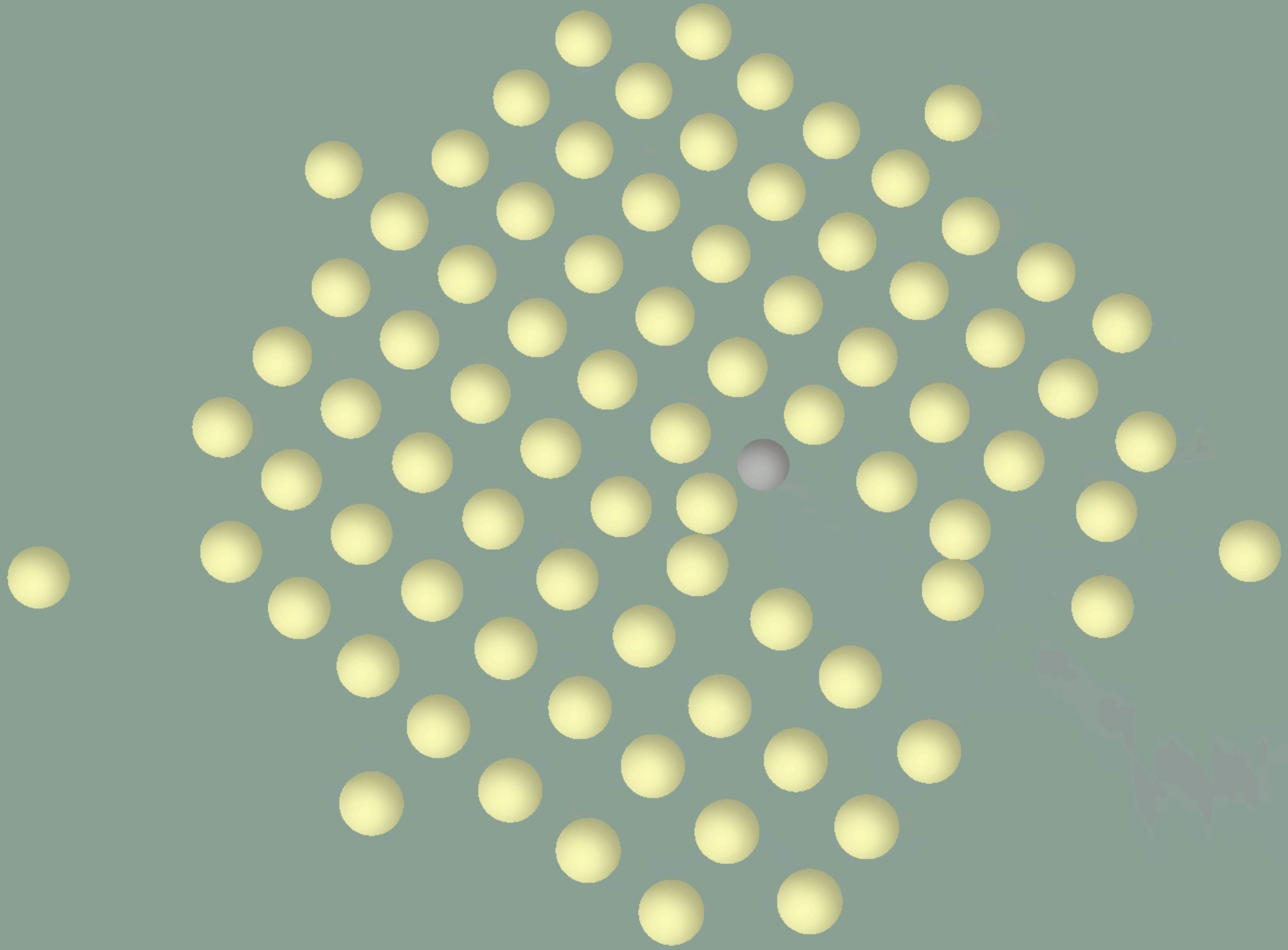
- Force model for collision is quadratic repulsion
- The force exerted on particle i by j:

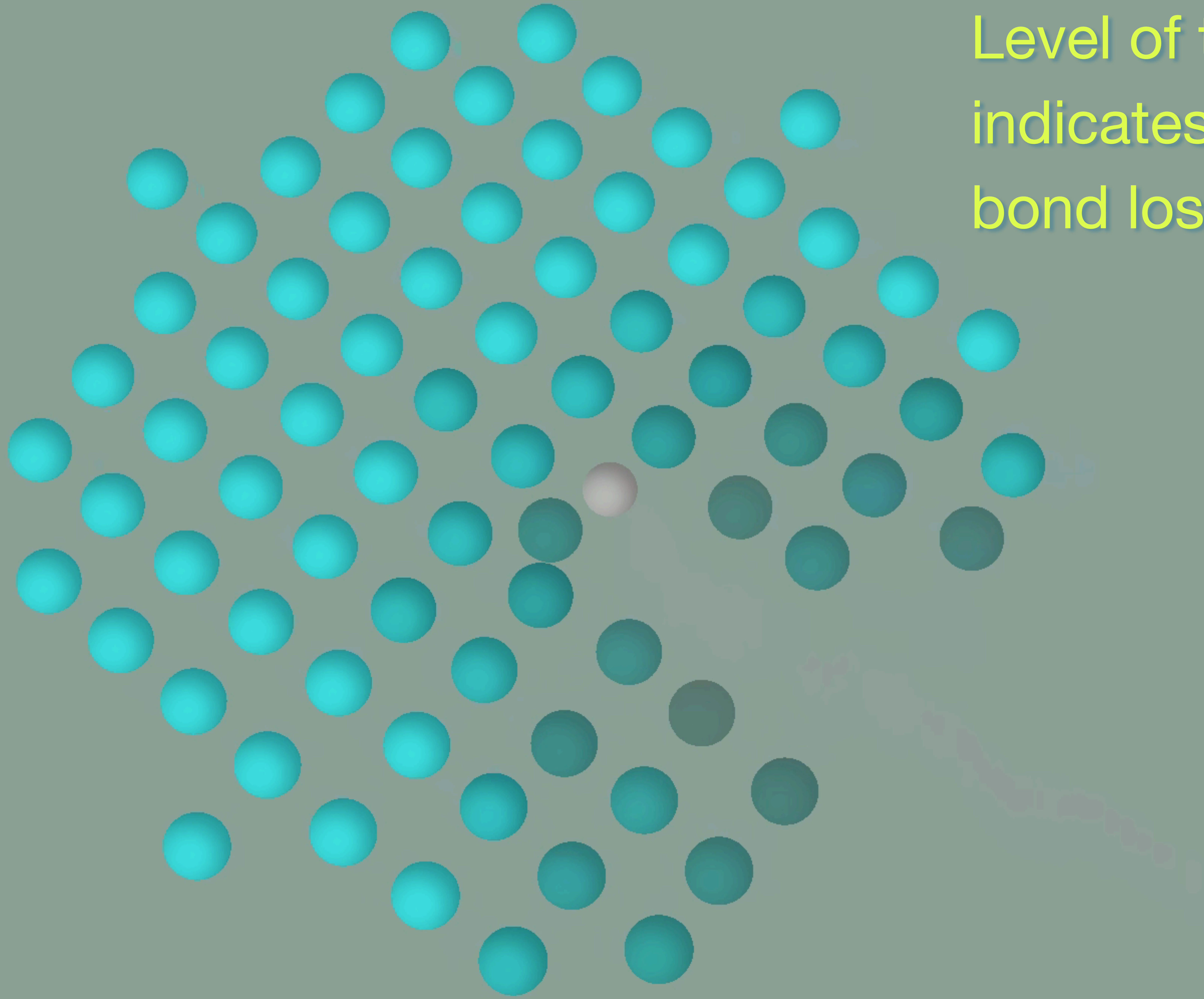
$$f_c = -k_c(\|x_i - x_j\| - \frac{\lambda}{2})^2 \frac{x_i - x_j}{\|x_i - x_j\|}$$

where k_c is a collision constant, e.g. 10^8 Pa

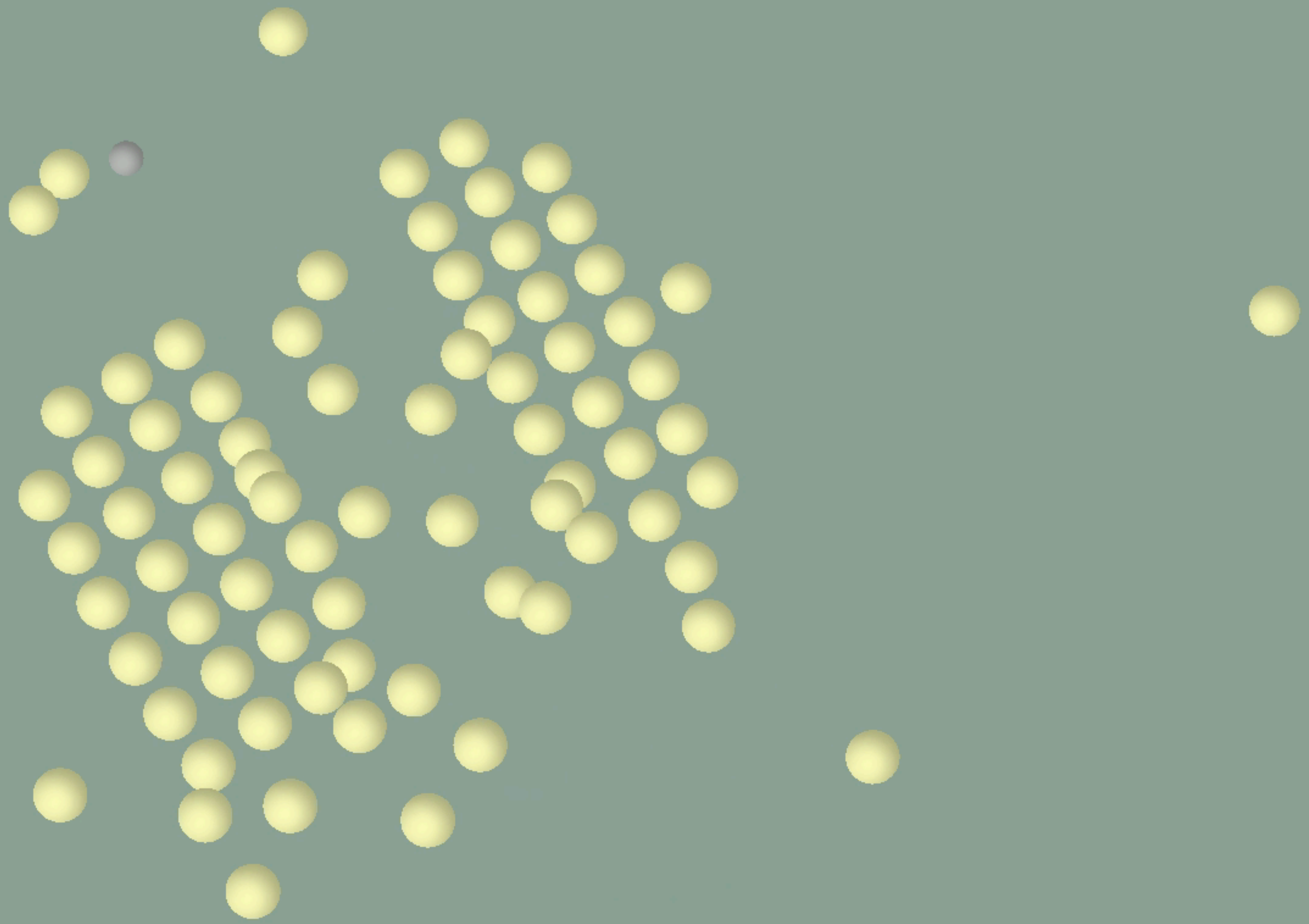
λ is a minimum bond distance

- Particles in same connected component don't collide
- Collision applied to particles at distance less than $\frac{\lambda}{2}$

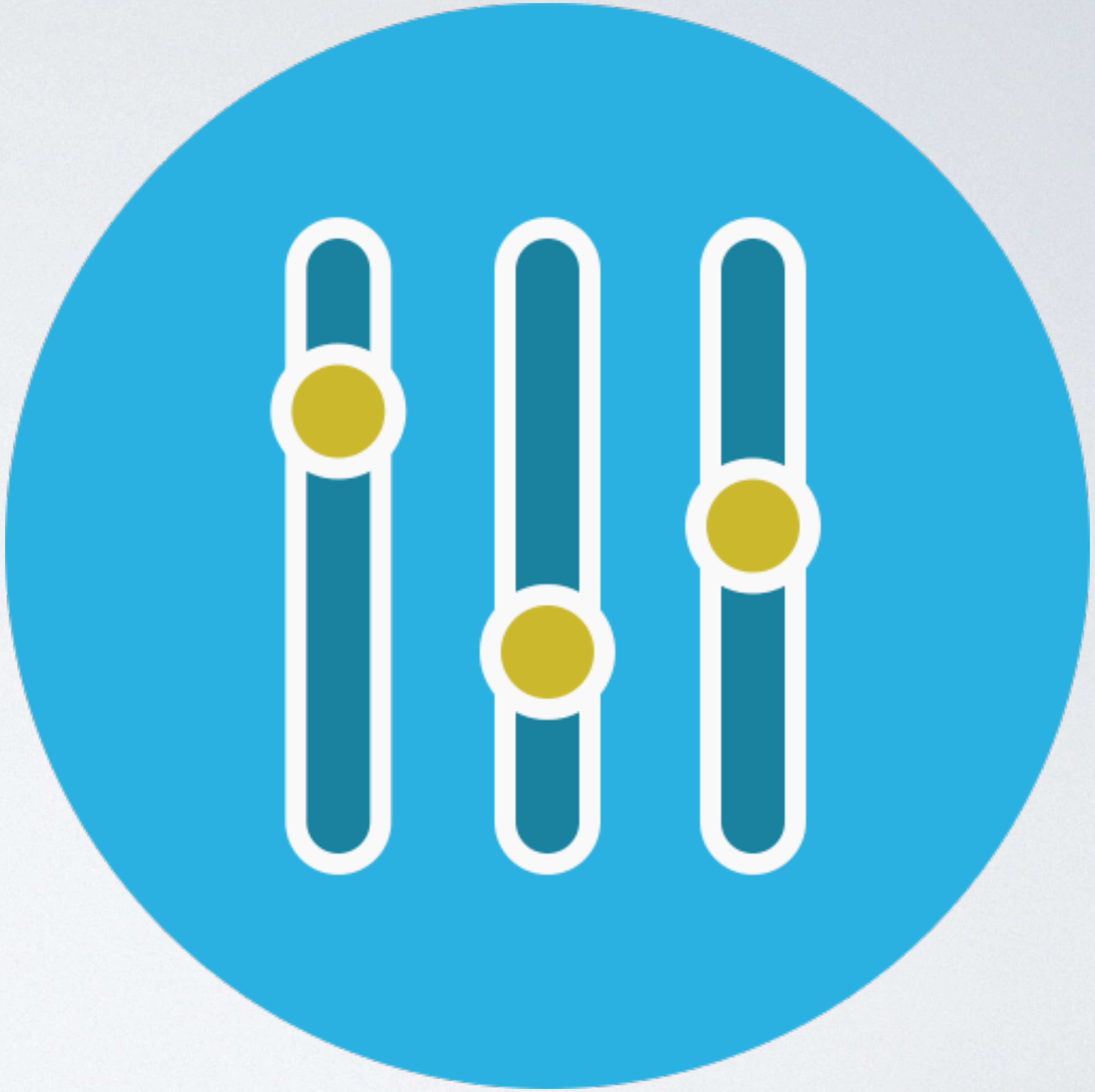




Level of transparency indicates percentage of bond loss



Peridynamic Parameters



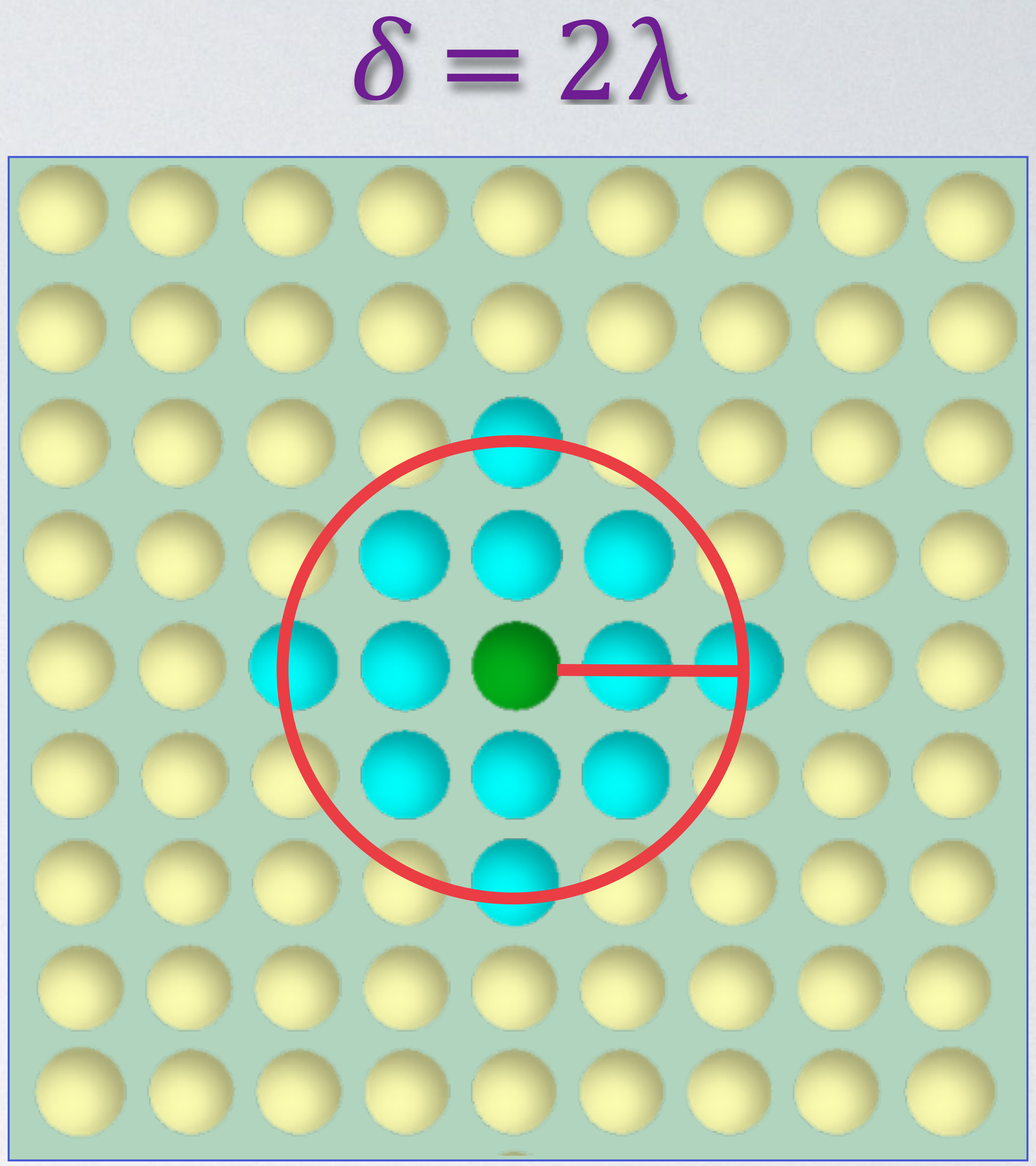
Peridynamic Parameters

- A peridynamic spring constant k_s is given by

$$k_s = \frac{18\mathrm{K}}{\pi\delta^4}$$

  - $\mathbf{K}$ is a bulk modulus of the material
    - e.g. 35 GPa for glass, 160 GPa for steel

Peridynamic Parameters



- A peridynamic spring constant k_s is given by

$$k_s = \frac{18\mathrm{K}}{\pi\delta^4} = \frac{18\mathrm{K}}{\pi(N\lambda)^4}$$

  - K is a bulk modulus of the material
    - e.g. 35 GPa for glass, 160 GPa for steel
  - N is a scaling integer, typically a value between 2 and 6
  - λ is a minimum bond distance

Peridynamic Parameters

- A strain limit τ is given by[1]

$$\tau = \sqrt{\frac{5\mathrm{G}}{\mathrm{K}\delta}} = \sqrt{\frac{5\mathrm{G}}{\mathrm{K}(N\lambda)}}$$

  - $\mathbf{G}$ is the fracture energy of the material

- We also allow a user to give a specific value to it.

[1] Silling S., Askari E.: Peridynamic modeling of impact damage. In *ASME Conference Proceedings* (2004), pp. 197– 205.

Tool Design

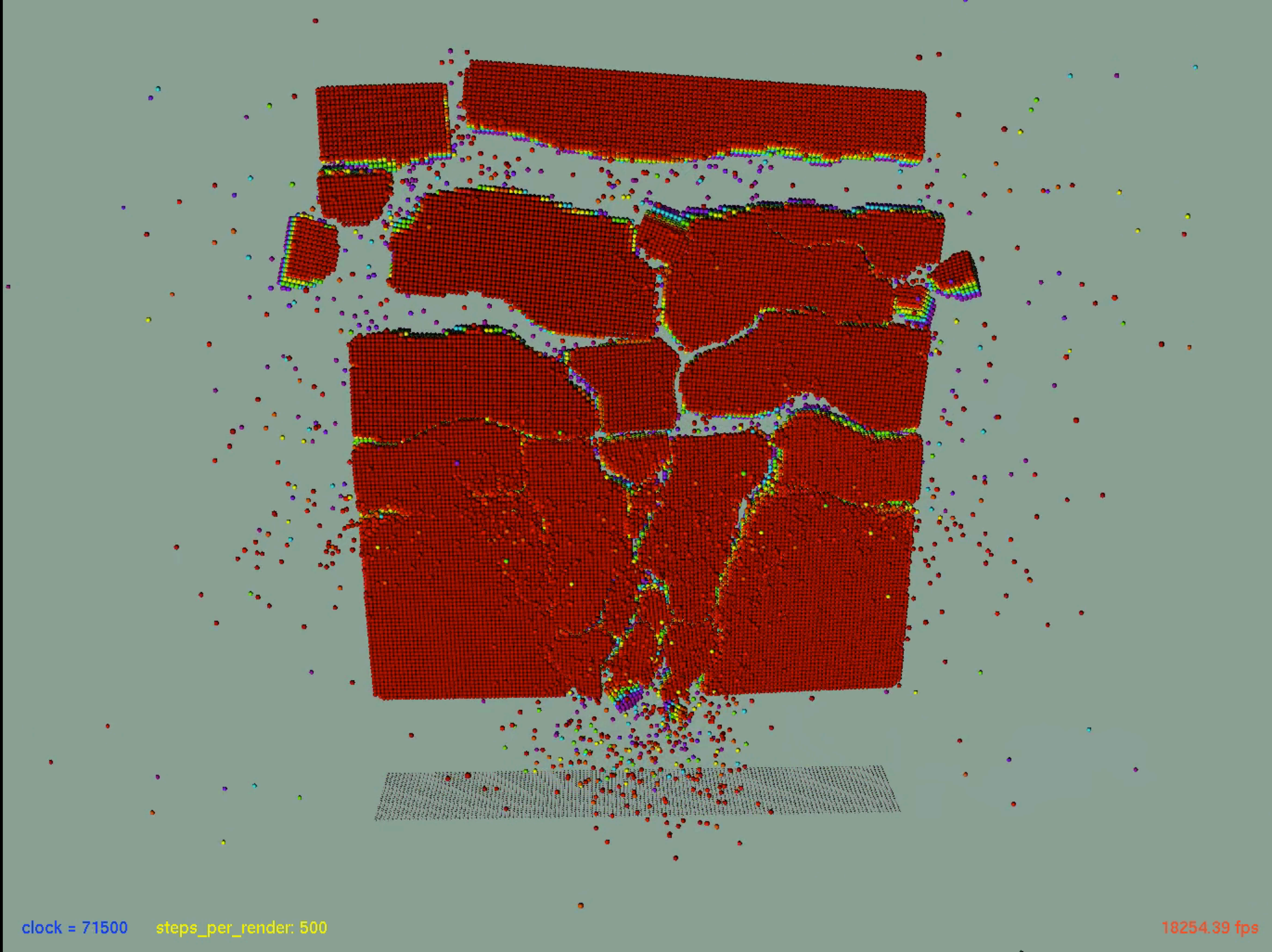
Visual Exploration Tool

Visual Exploration Tool

- Simulation takes many small time steps to compute
  - considerably small time step, e.g. 120 ns
- Record simulation data on the disk for later manipulation
- Rapid development with Python
  - using PyOpenGL with GLSL for display
  - using PyCUDA for intensive calculation

Overview:

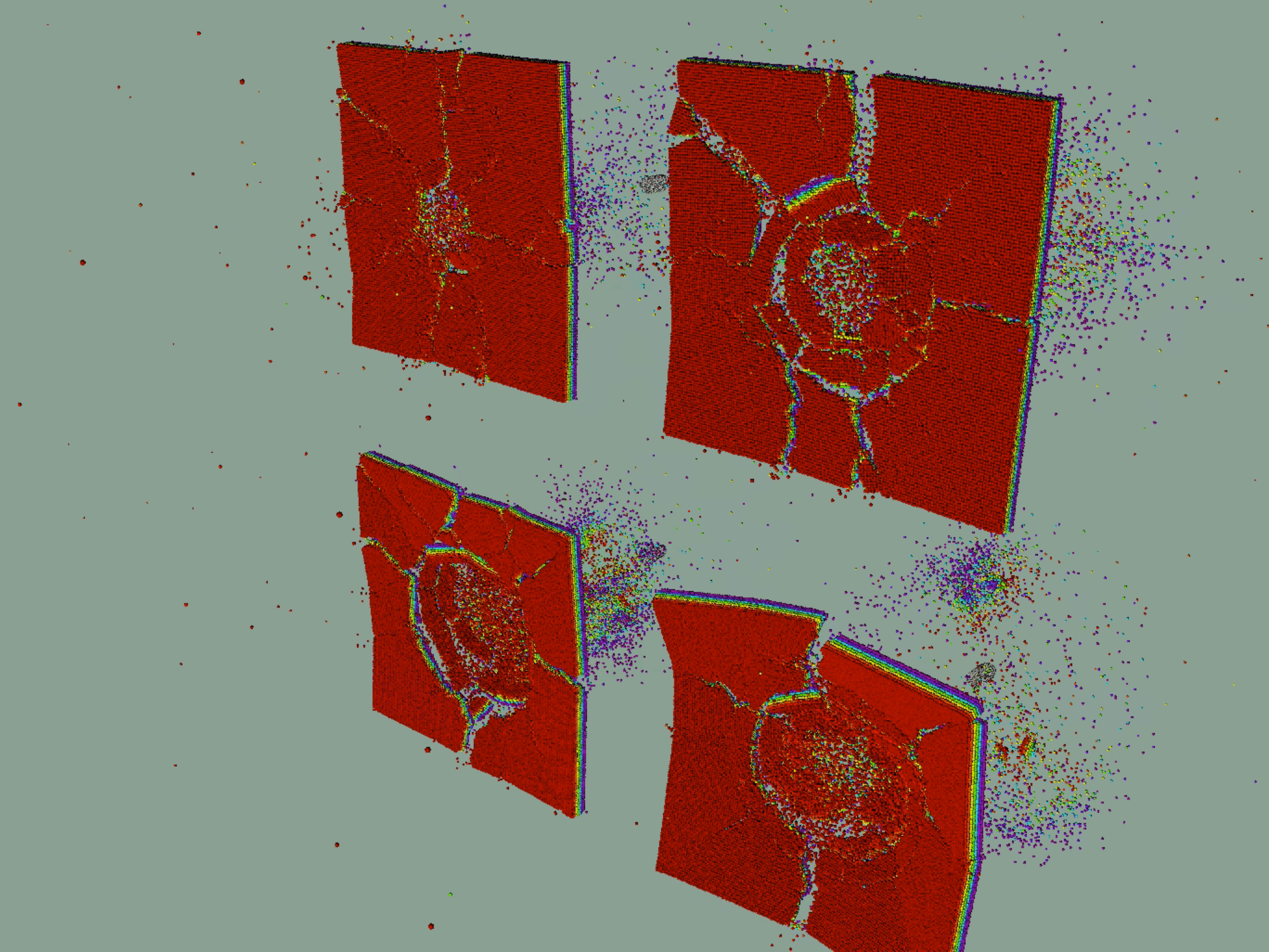
spectral colors
forward
backward
pause
step through
resume



Single World:

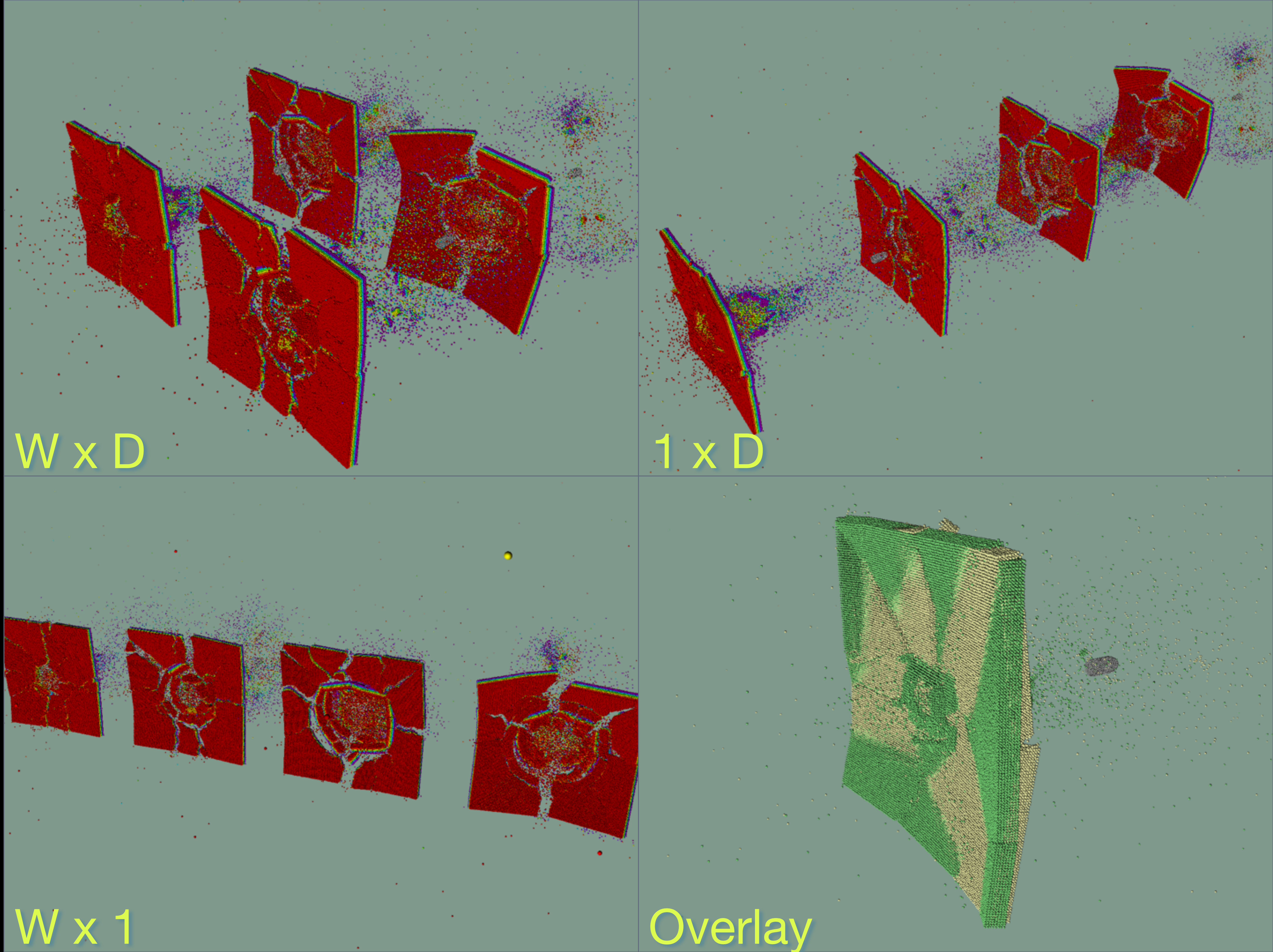
Show multiple simulation runs simultaneously in same space

Front Layout



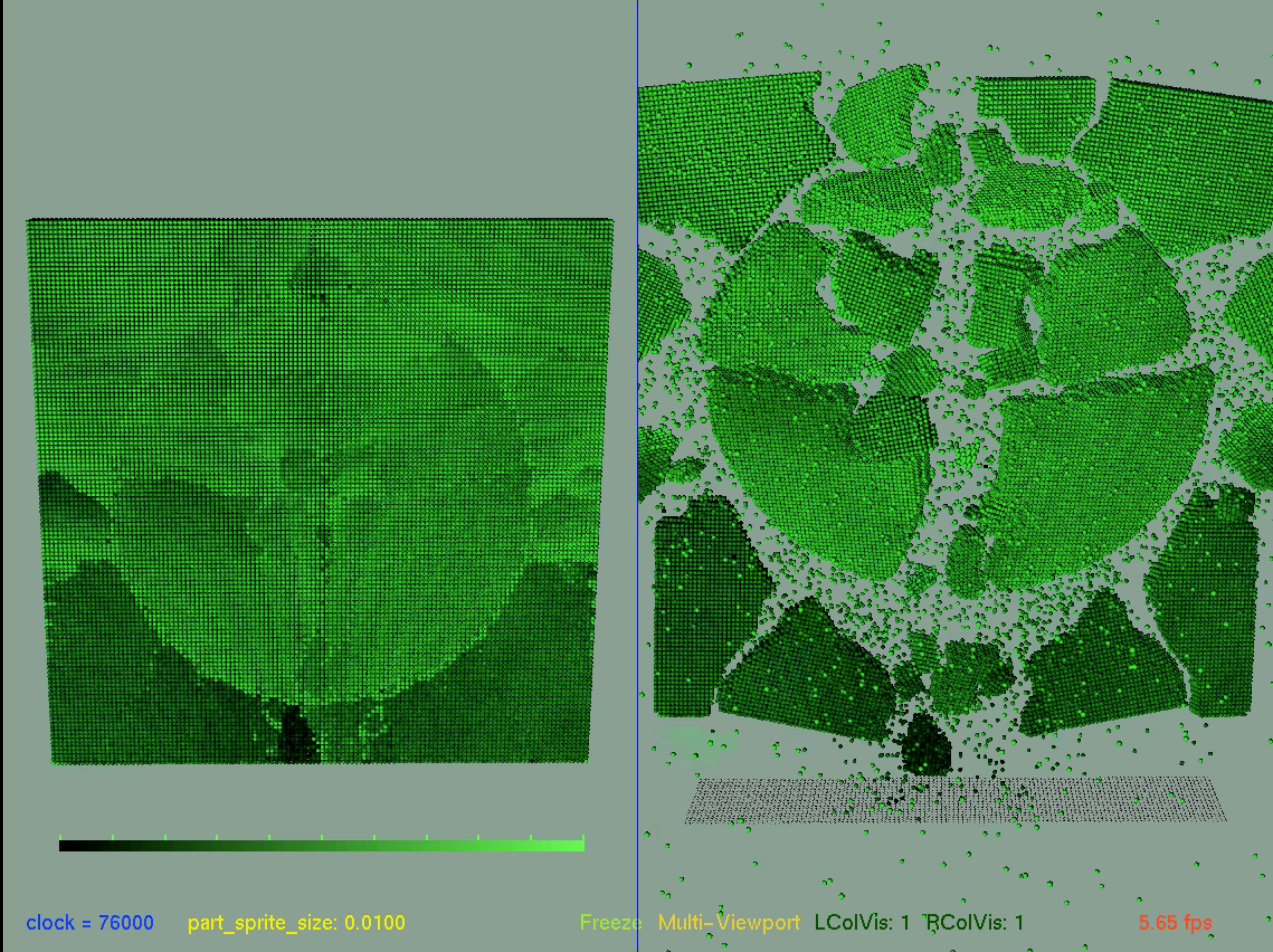
Single World:

Other Layouts



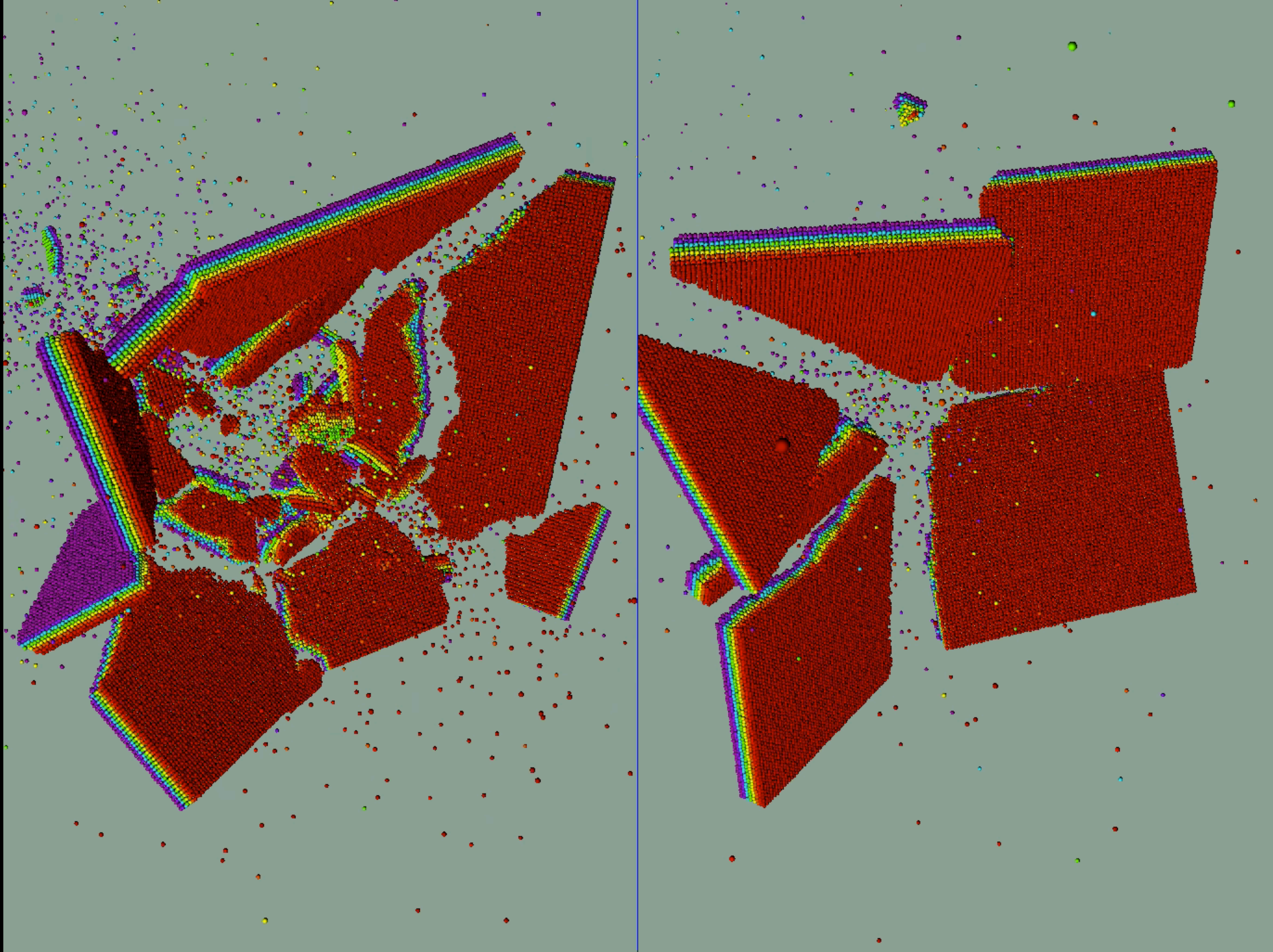
Particle Colors

Visualizing magnitude of particle velocity



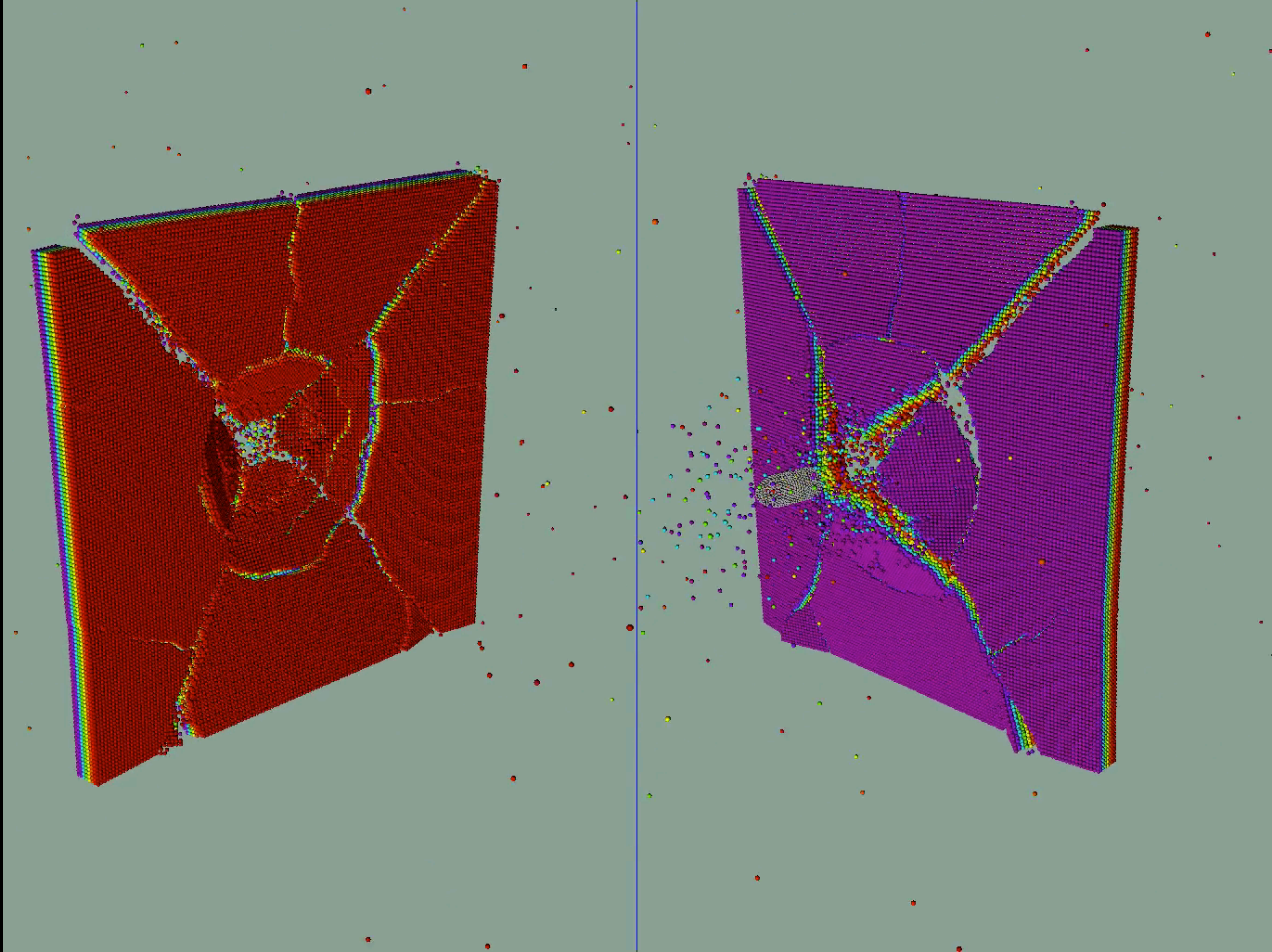
Multi-Viewport:

Side by side comparison



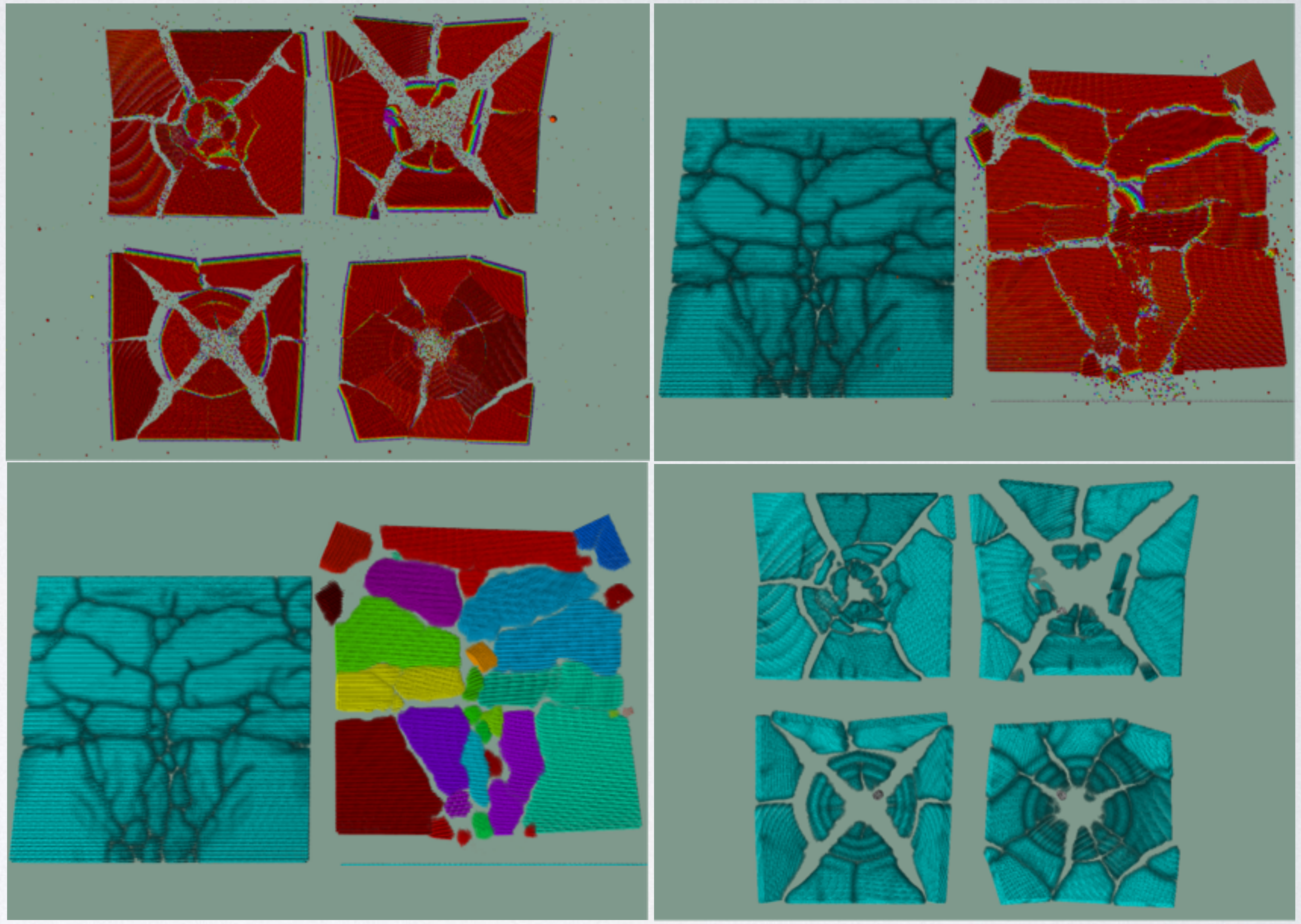
Multi-View:

Show impacts at front and back simultaneously



Visualizing The Simulation

The "Results" of The Tool



Parameter N: varying from 4 to 6

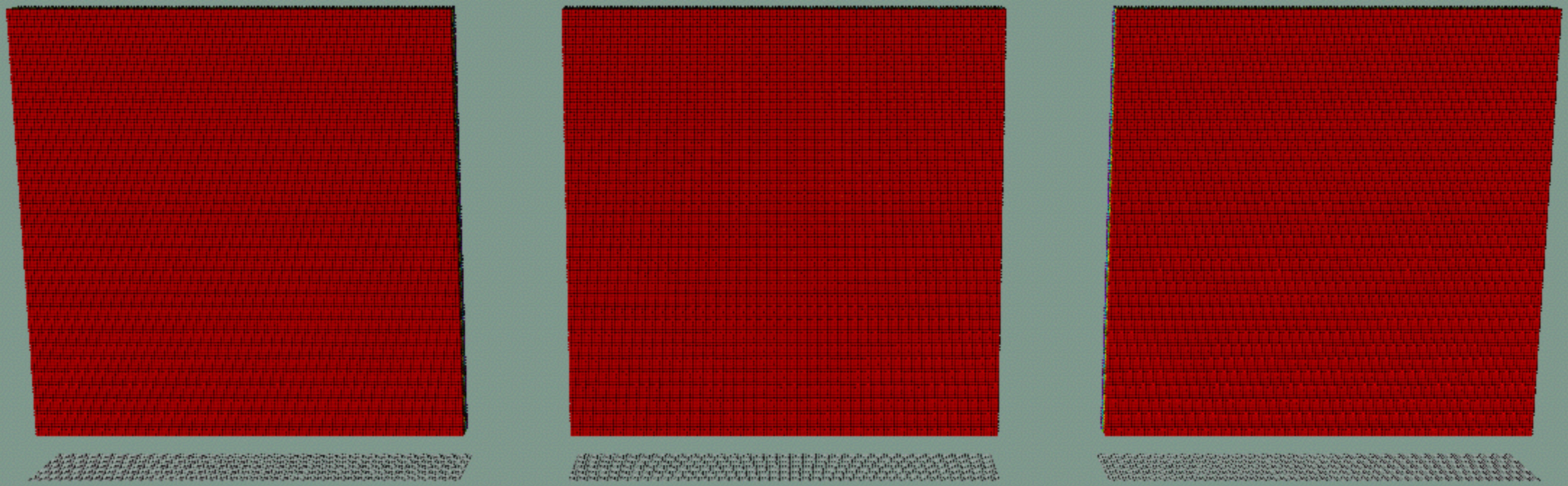
No. of particles 131,072 + 3,072

$K = 35$ GPa

$k_{c:g\text{-}g} = 0.1$ GPa

$k_{c:f\text{-}g} = 10$ GPa

$\tau = 0.005$



N = 4
maxbonds = 255

N = 5
maxbonds = 483

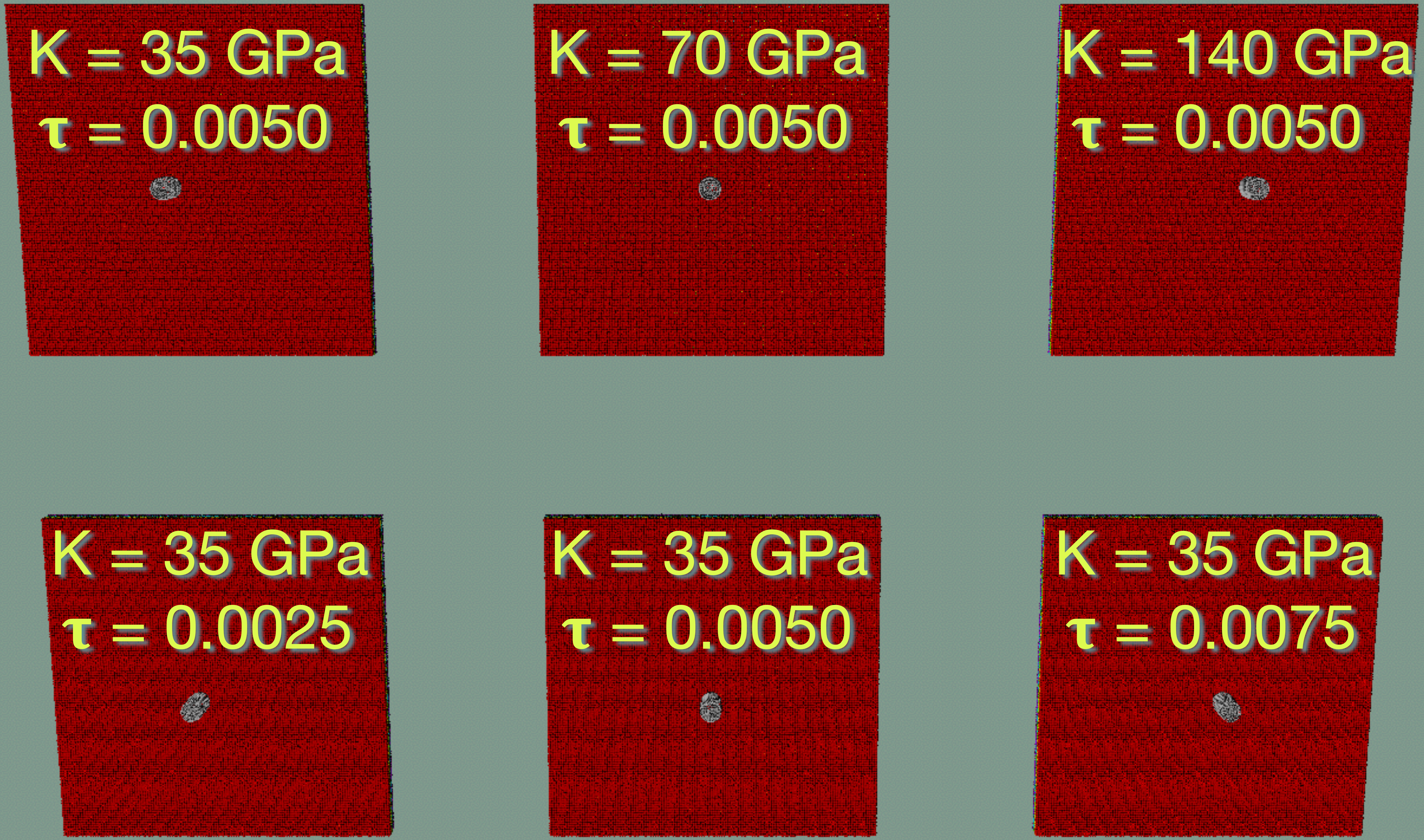
N = 6
maxbonds = 779


N = 4
N = 5
N = 6

Parameter K, τ:

No. of particles
131,072 + 1,093

N = 3
maxbonds = 117
$k_{c:g\text{-}g}$ = 0.1 GPa
$k_{c:b\text{-}g}$ = 10 GPa

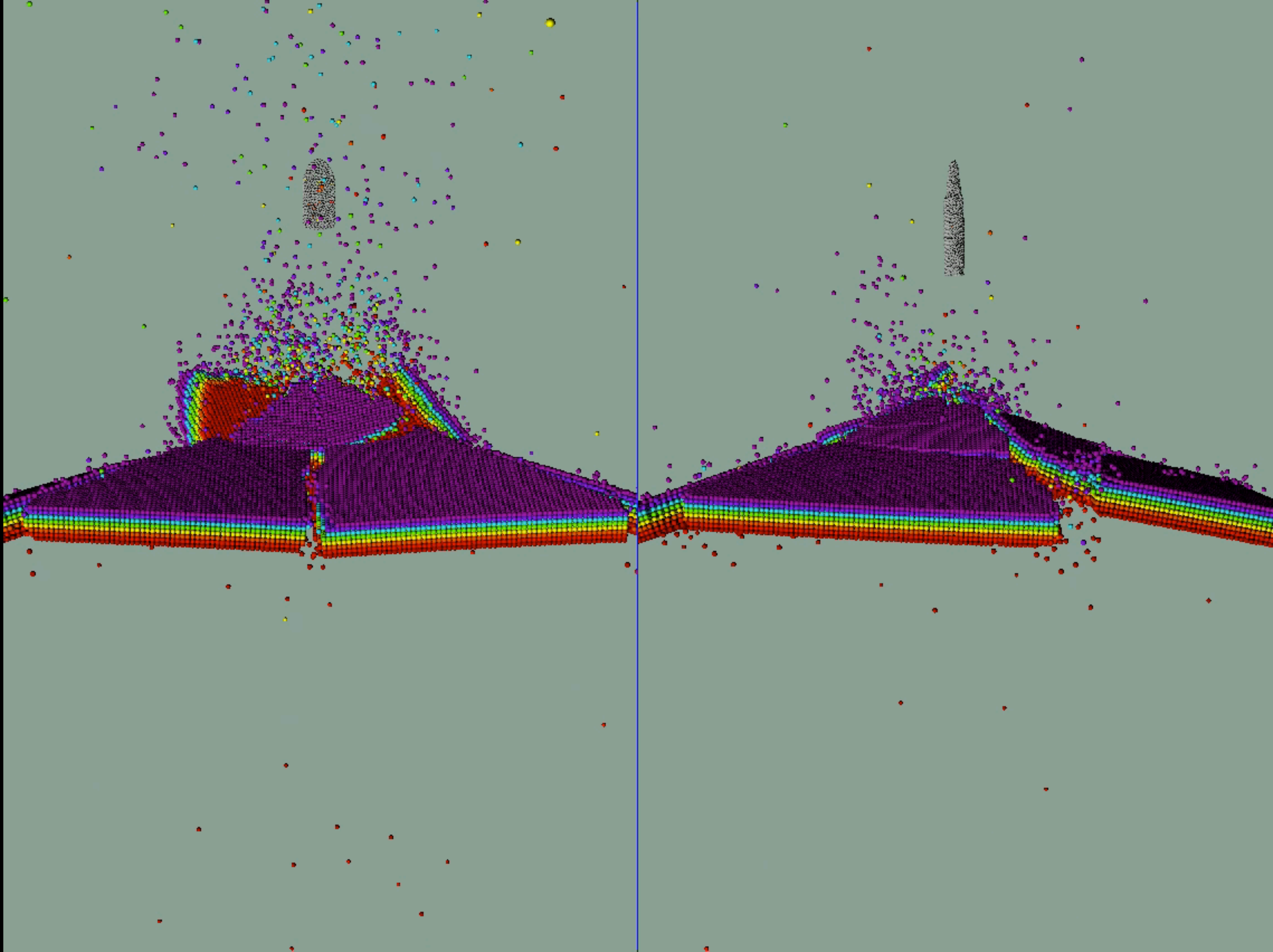



K = 35 GPa
τ = 0.0050
K = 70 GPa
τ = 0.0050
K = 140 GPa
τ = 0.0050
K = 35 GPa
τ = 0.0025
K = 35 GPa
τ = 0.0050
K = 35 GPa
τ = 0.0075

Bullet Shapes

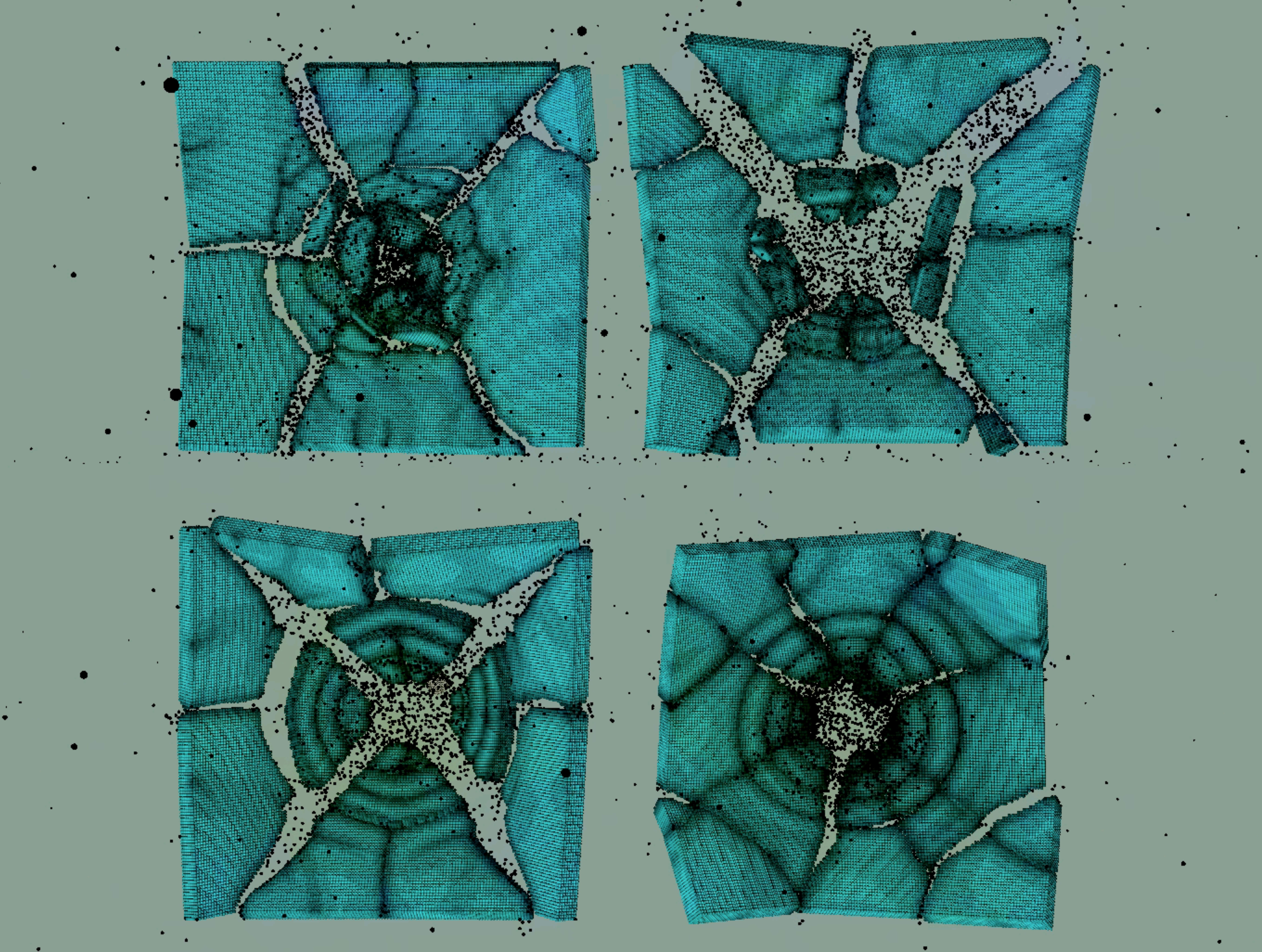
No. of particles
131,072
(L) Round 1,093
(R) Sniper 2,158

N = 4
maxbonds = 255
K = 35 GPa
$k_{C:\text{g-g}}$ = 0.1 GPa
$k_{C:\text{b-g}}$ = 10 GPa
τ = 0.005



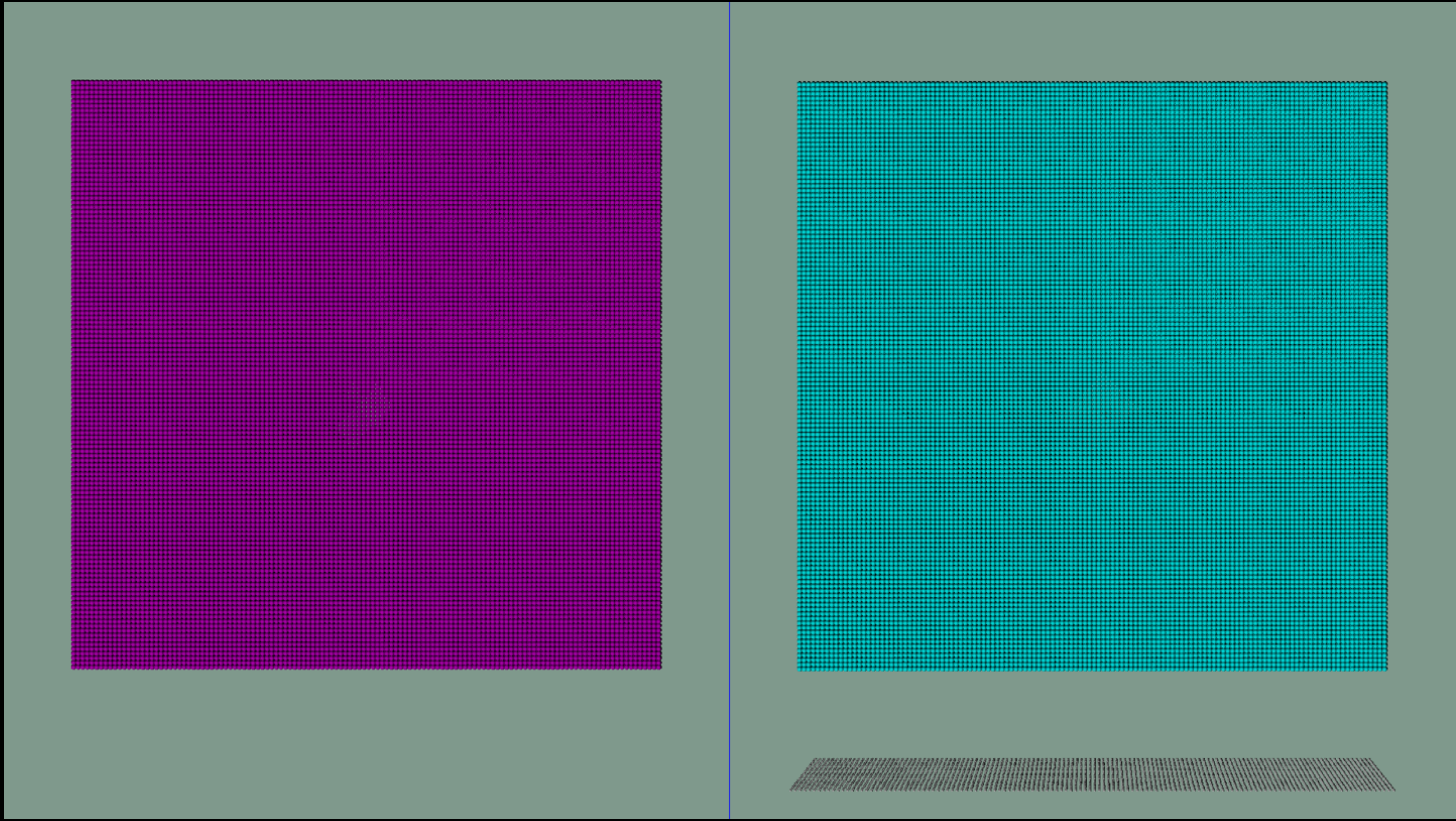
Visualizing damages

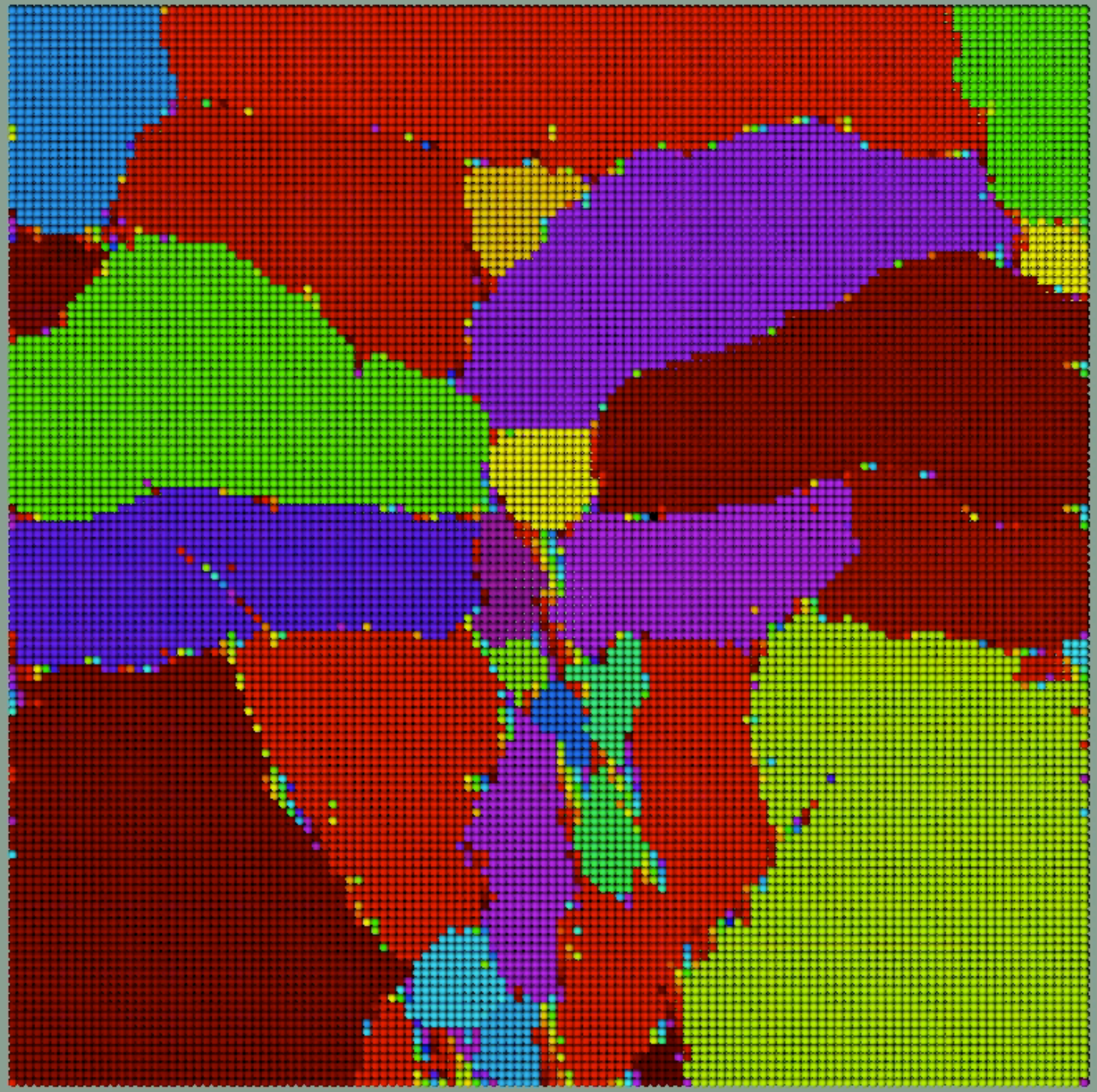
Color intensity
Transparency

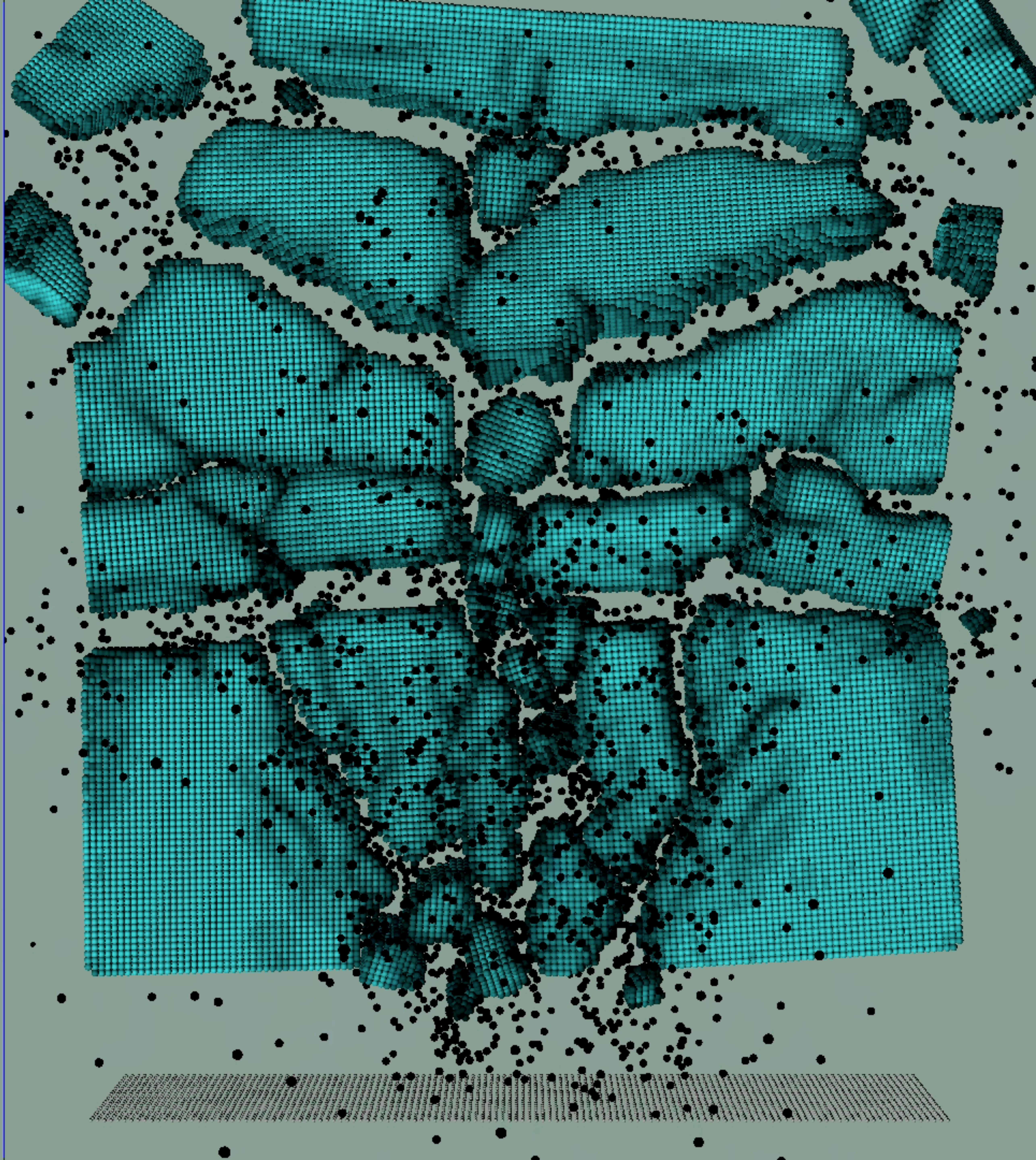


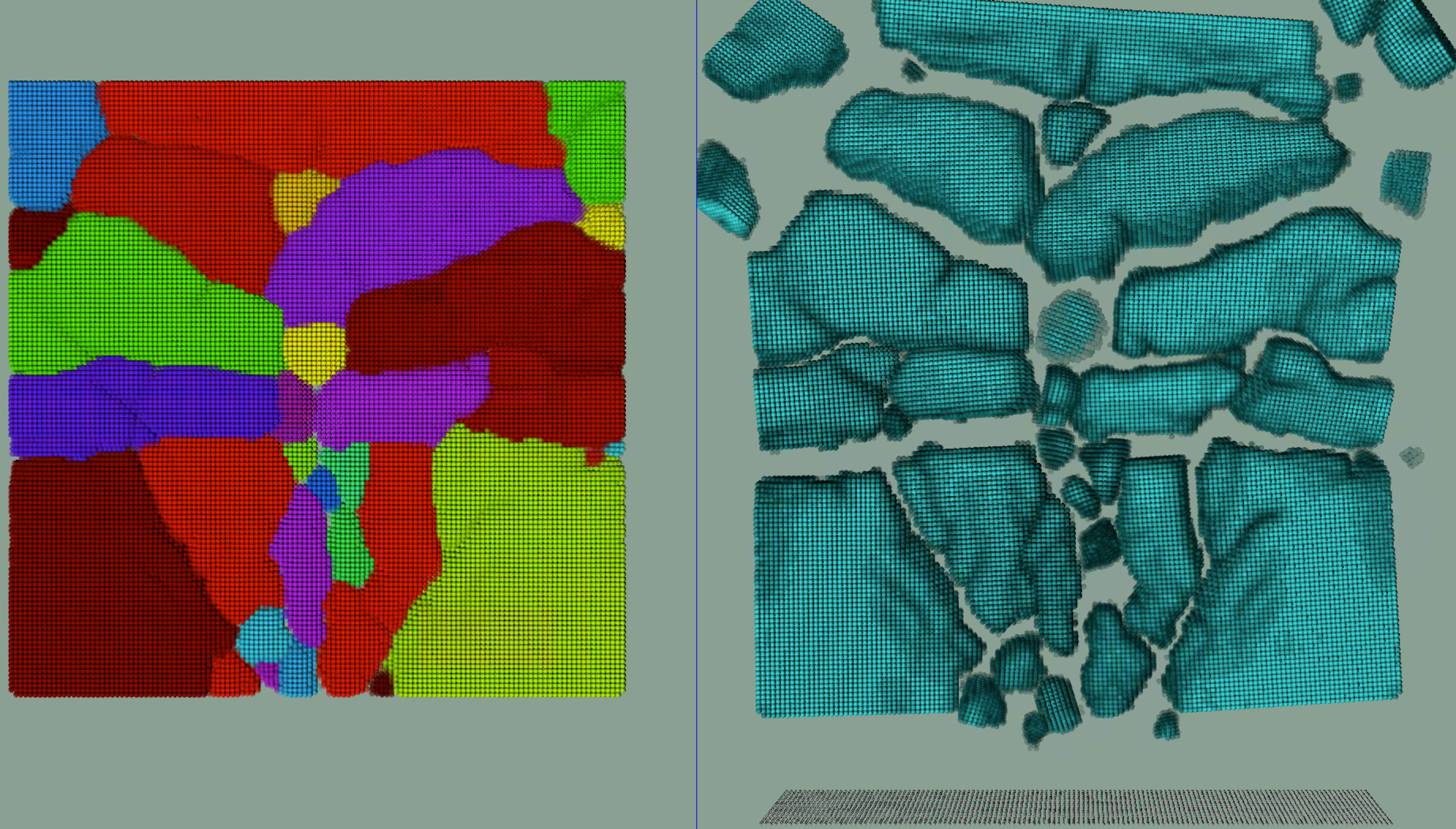
Visualizing connected components

Particles: 131,072
N = 3
max bonds = 122



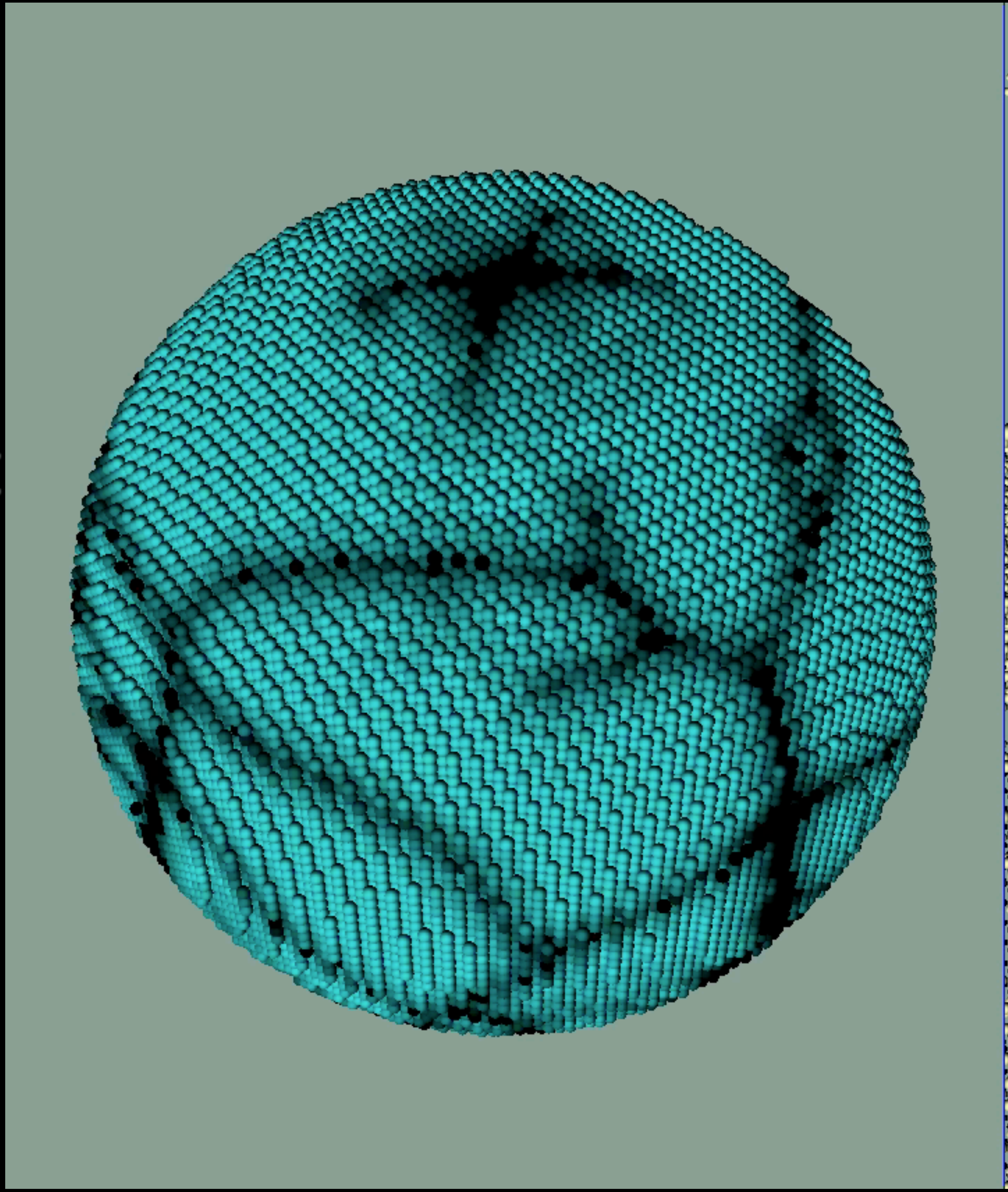


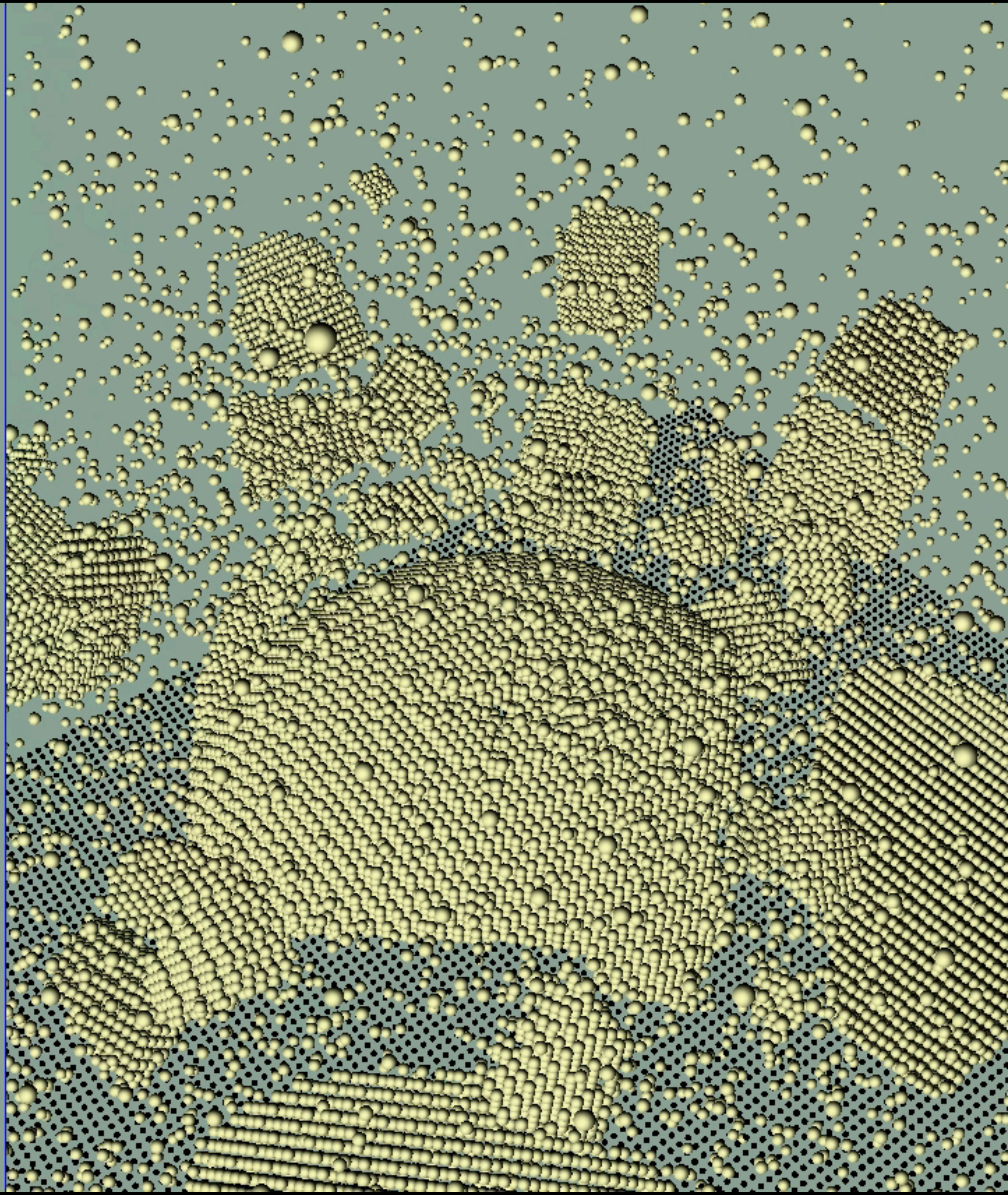




Data Visualization

View fractured line development





Conclusion and Future Directions

- Simulations are not in real time, and parameter setting can be nontrivial
- Parallelism across multiple GPUs
- Real-time surface extraction
- Anti-aliasing on point cloud rendering
    - Screen space ambient occlusion

Acknowledgments

- Chakrit Watcharopas (cwatcha@clemson.edu)
- Special thanks to Cliff Woolley, Dave Luebke, and Chandra Cheij (NVIDIA)
- Research generously supported by:

Please complete the Presenter Evaluation sent to you by email or through the GTC Mobile App. Your feedback is important!