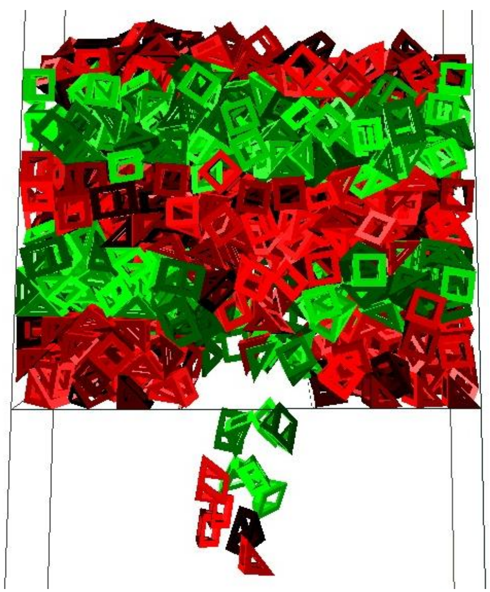
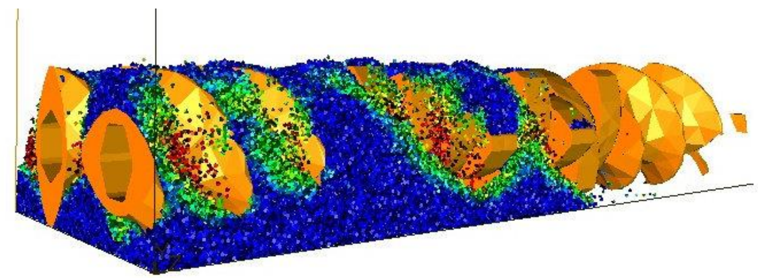
ADVANCES IN COMPUTATIONAL MECHANICS USING GPUS

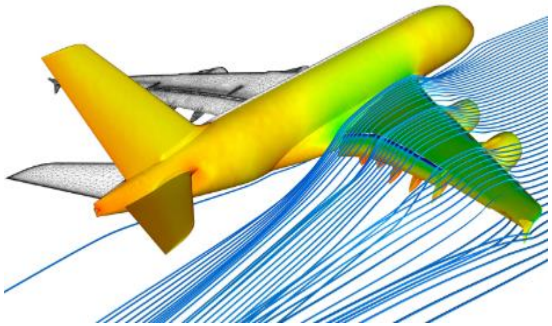
Nicolin Govender (Surrey,UJ), Charley Wu (Surrey),Daniel Wilke (UP)





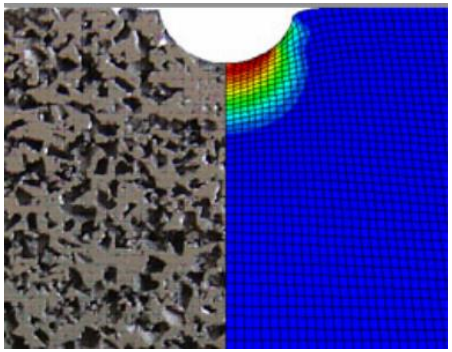
COMPUTATIONAL METHODS

CFD
(Volume of Fluid ,Finite Difference)



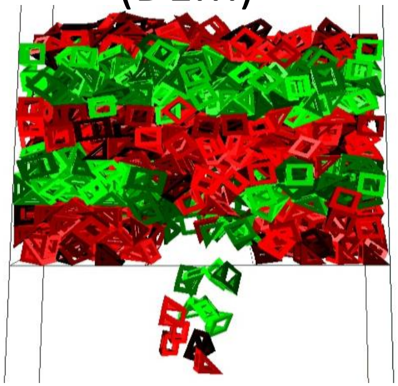
(1951)

Finite Element
(FEM)



(1956)

Discrete Element
(DEM)



Treats material as a continuum, computationally cheap.

Discrete nature cannot be ignored

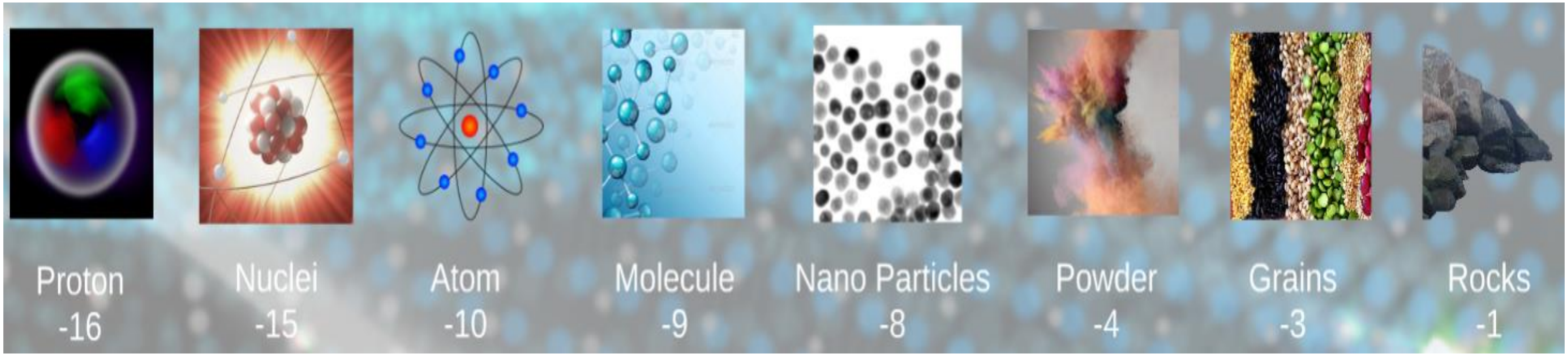
Even at home..

Focus of this talk: Particulate Material

Second most manipulated substance on the planet after water.

Granular material is out of this world!

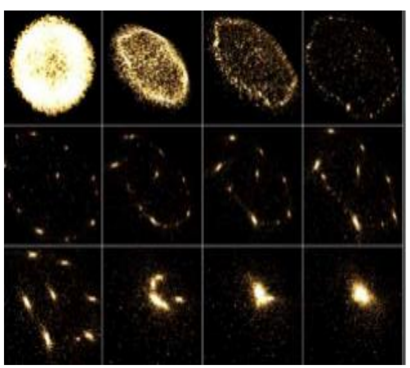
PARTICULATE SIZES AND INTERACTION

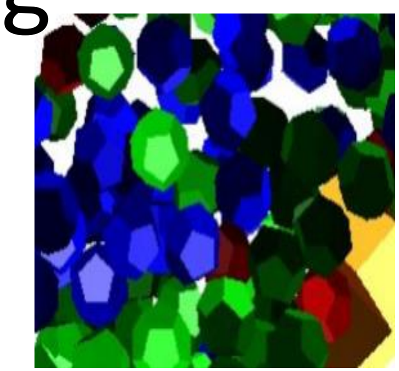


Log_{10} (m)

Particle Size

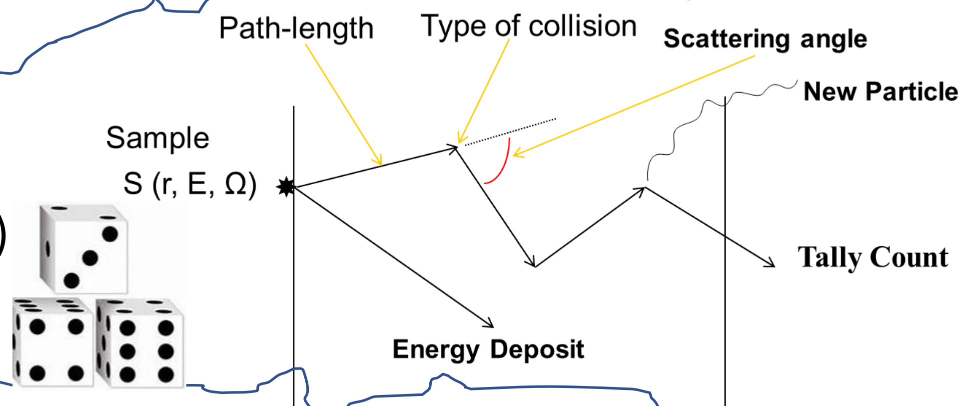
Importance of considering physical interaction





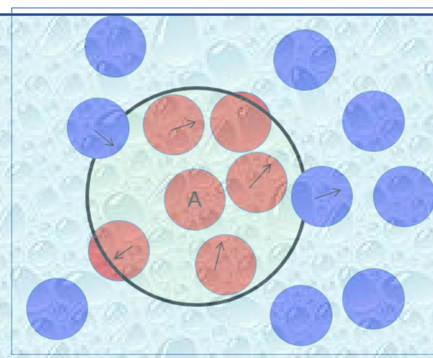
SOLUTION CLASSES

Event Based
(Monte Carlo)



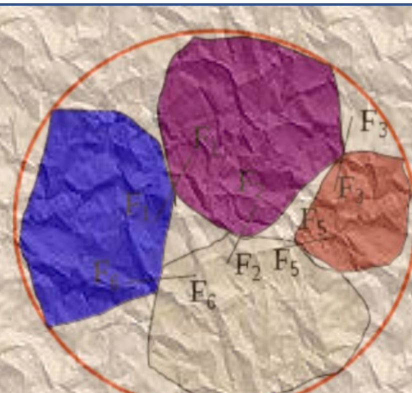
At particle level embarrassingly parallel. Instruction complexity some what divergent.

Proximity Based
(Molecular Dynamics)



At particle level embarrassingly parallel. Instruction complexity fairly similar.

Contact Based
(DEM, Impulse)

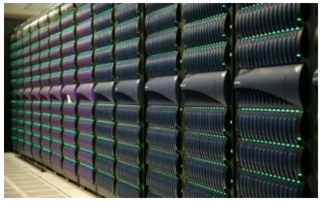


At particle level embarrassingly parallel. Instruction complexity is divergent for complex shape.

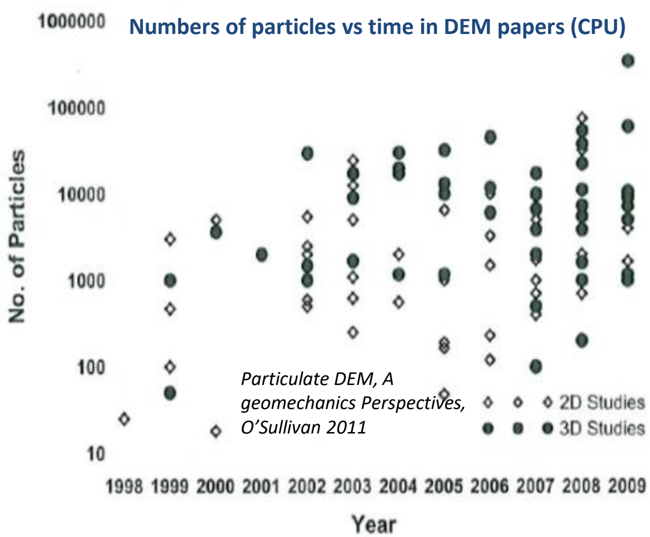
CHALLENGES IN DEM

On typical computers!

Not clusters !



Particle Number: *Numerous papers keyword: “large scale”, showing hundreds of thousands to a few millions of particles taking months to run.*

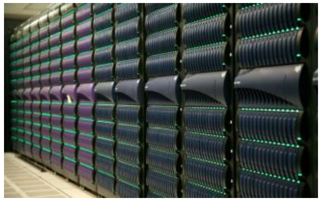


UNIVERSITY OF SURREY

Challenges in DEM

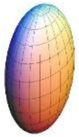
On typical computers!

Not clusters !



Particle Shape: *Spheres are the simplest of shapes and when "large scale" is for spheres.*

Ellipsoids:
Better estimation of shape, contact detection more expensive spheres.

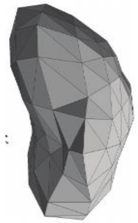


Clumped spheres:
Requires many spheres to create a given shape.
Surface has artificial roughness (raspberry effect).
Computationally very expensive for complex shapes.

Actual Shape

Super quadratics:
More accurate than clumped spheres for many shapes.
Can become expensive to solve.
Difficulties encountered for concave exponents.

Polyhedra:
Most general of all shapes, physically most accurate.
Computationally very expensive.

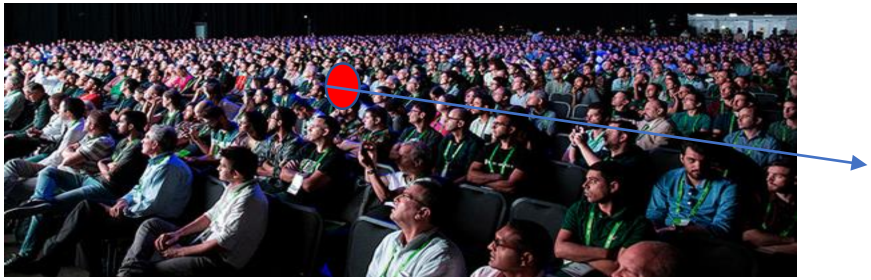


John Lane, A Review of Discrete Element Method (DEM) Particle Shapes and Size Distributions for Lunar Soil , NASA, 2011

DEM ALGORITHM

- Largest computational cost is collision detection.
- All objects need to be tested against each other O(N) complexity.

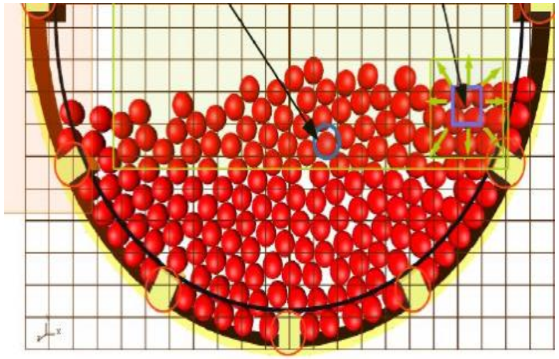
A common question in a number of areas.

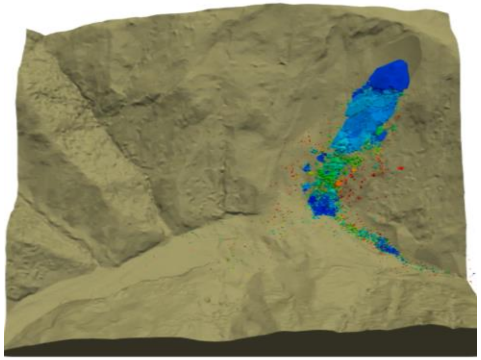


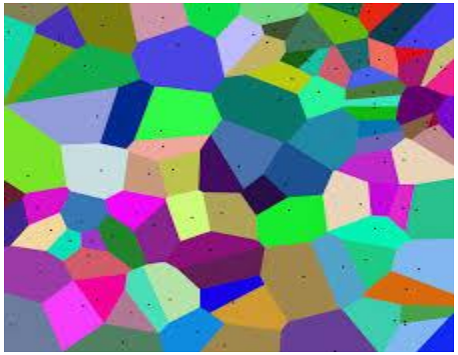
Who are my neighbors?

- Collision detection is a well known problem in computer science.
- Various spatial partitioning algorithms to reduce from O(N).

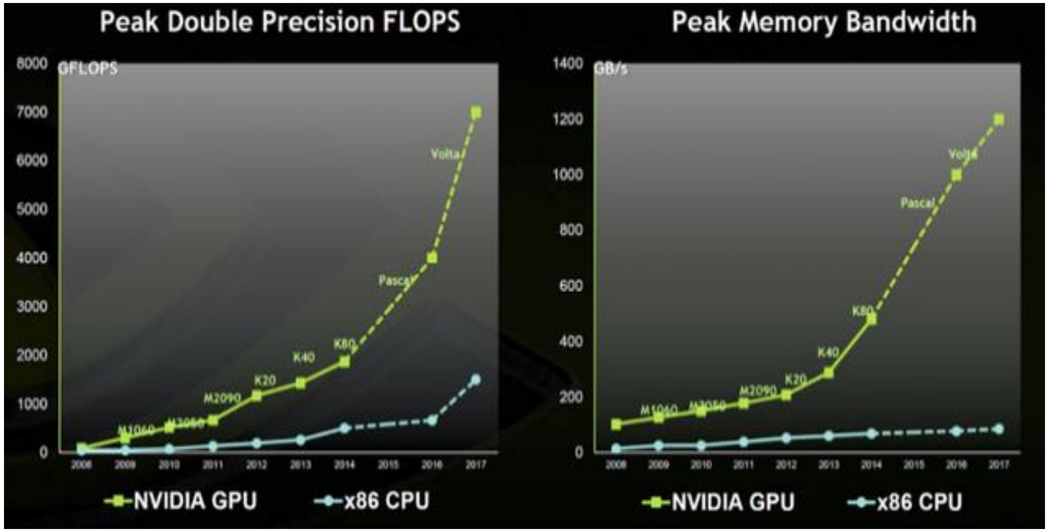
- Uniform grid and BVH are the most popular in DEM.
  - Uniform grid is the fastest when particles are similar sized.
  - Expensive in terms of memory when domain is dispersed.
  - BVH is ideal when objects move little relative to each other.







THE GAME CHANGER



2009: Talk at SC on using OpenGL for collision detection between points and geometric primitives for MC.

2010: Started with CUDA MD (emulated)

2011: Papers by Radake, Ge using GPUs for DEM with spheres.

2012: First DEM code for polyhedra on GPU, (100k to 32 million).

2013: CUDA research center and hosting on git of Blaze-DEM

2014: PhD and invited talk @ DEM 8

2015: ROCKY commercial DEM code

2017: EDEM OpenCL

2019: We still set the standard ☺

UNIVERSITY OF SURREY

GPU IMPLEMENTATION

- For spherical particles we are as fast as we can be. Bottle neck is with global memory access speed (task is SIMD). Force computation requires various values to be loaded from memory.
  MEMORY BOUND

- Using shared memory not possible as threads are run per particle so no data dependence on other particles (cannot be tiled). Even with the NN of each particle nothing is common.
  Shared Memory DOES NOT HELP

- Each particle needs to check if its current contact existed in the previous step. Within each thread loop over all previous particle contacts (History).
  Register Pressure

Benchmark for spherical particles

10 Million 1mm Particles, dt = 3.5E-6

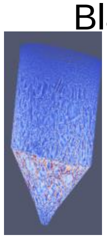
Liggghts-P: **60 Cores**: 1 second = **46 hours**

Cost $ 16000 for CPUs

*(Price at launch in 2013)= **$ 96000**

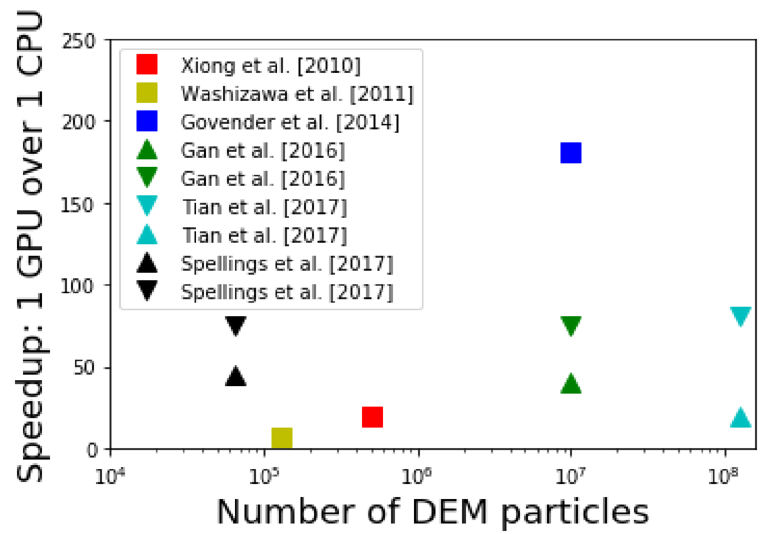
Reported 40x speed up over a commercial code

Blaze-DEM: 1 **GTX 980** : 1 second = **3.2 hours**



Cost $ 500

GPU 15X Faster, 30X Cheaper



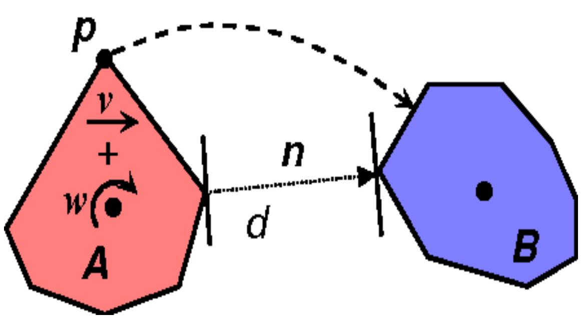
Gan et al. Needed 32 GPUS to get similar performance. Y He is 500x slower than us.

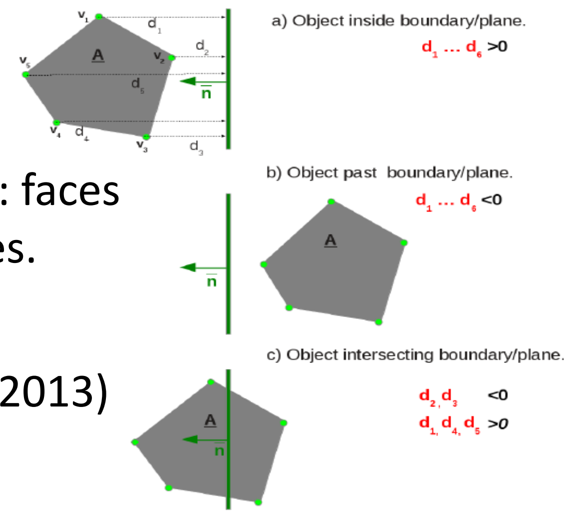
GPU IMPLEMENTATION

- In terms of spheres we are happy as we can be, as the compute per particle vs the memory transactions is low. Achieved goal of increasing particle number in a reasonable time.

- Polyhedra require a detailed contact check this takes 80% of the time. The NN search for spheres is used as the first check to prune neighbors.

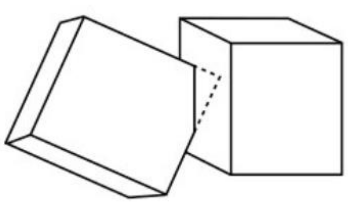
- Various methods for testing collision detection between polyhedra. Most popular is the common plane which is an iterative method, used by commercial codes.



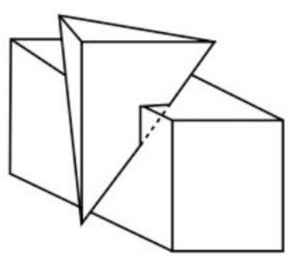


Finite number of planes: faces and cross between edges.

Re-formulated for GPU (Govender 2013) only face planes are tested.



Vertex inside a polyhedron

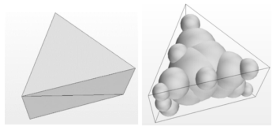


Object penetration without a vertex of one object contained in the other

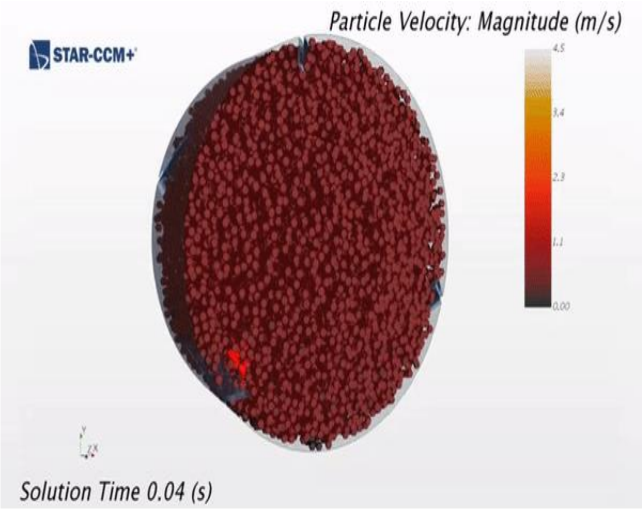
1. Problems when edges are involved.
2. Divergent threads
3. Normal is not uniquely defined!

POLYHEDRA IN COMMERCIAL SOFTWARE

Star CCM+: 4000 particles in 2018!

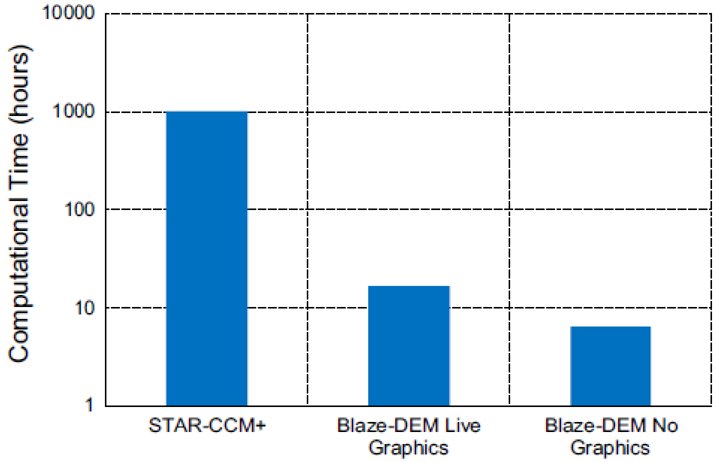


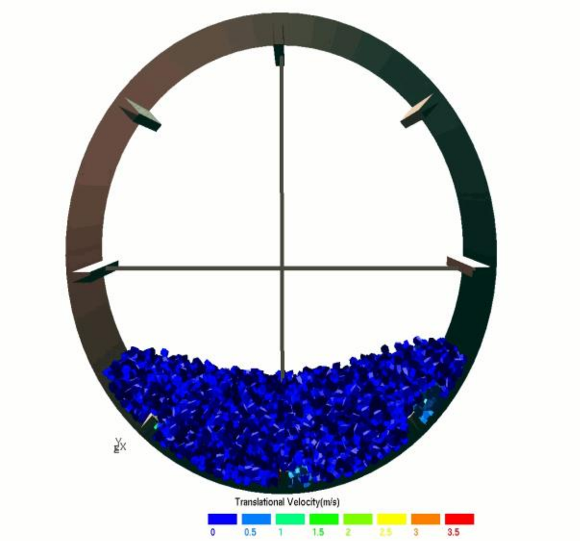
	Polyhedral	Composite
Time for 1s of simulated time, dt = 0.01s – 1 core, 4000 particles	424 s	3372 s



http://mdx2.plm.automation.siemens.com/blog/david-mann/star-ccm-v1204-preview-model-realistic-particle-shapes-polyhedral-dem-particles

I will use a dt of 1e-4 340s for 1s on GTX 1080 GPU. 1000X more steps and its correct!

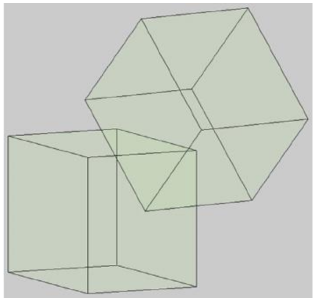




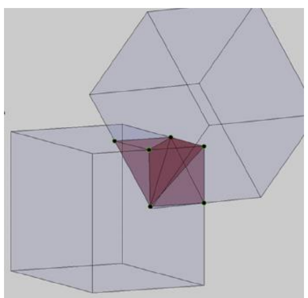
OUR APPROACH

Full accuracy using half the precision…

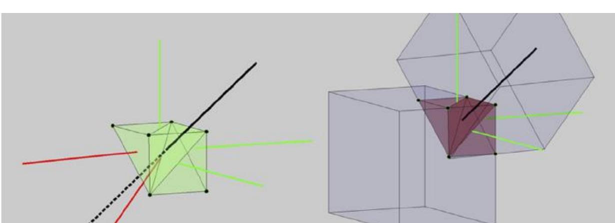
- Do it correct, when dealing with 3D object the contact region is a volume.

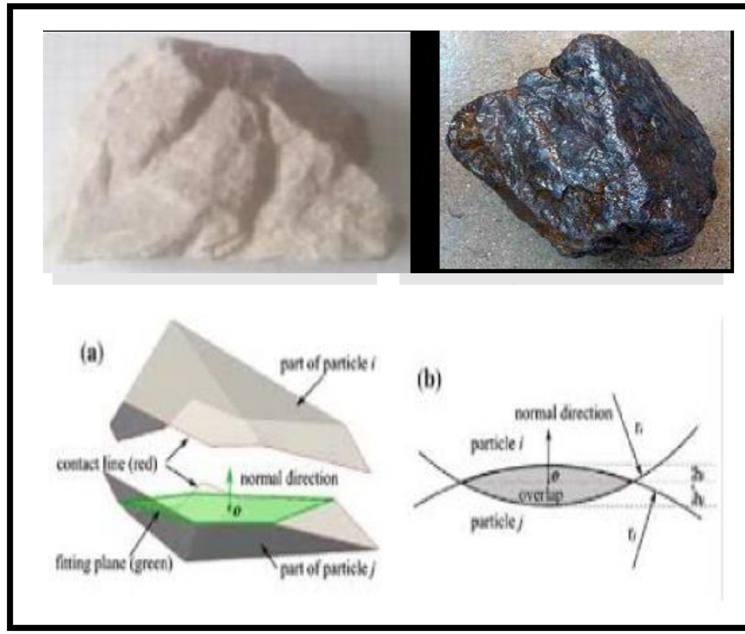


- Problem is cast in ray tracing form, resulting in a point cloud



- A convex hull is constructed to yield the resulting contact polyhedron.





Author	Shape	Physics Fidelity	Max particles	(Time N=5 x 10^5)
BLOCKS (2014, PhD thesis U Illinois)	*Poly	Highest	5000	186 days
iDEM (2014, PhD thesis U Illinois)	*Poly	Low	500000	2.8 days
BLAZE-DEM (2014)	Poly	High	32 x 10^6	28 min

Still around 5x faster than ROCKY DEM when using exact contact detection.

GPU IMPLEMENTATION

- Broad phase cannot eliminate enough neighbors cheaply, even if we use OABB determine intersection requires the polyhedron contact kernel which causes divergence.

- Adding a second pass on the output of broad phase does not reduce the computation time by much.

- Reason is that even in the case of a few NN the fact that we have to create a local array for the contact points as well as the faces of the resulting convex hull overflows registers and spills in global memory (any in kernel array spills).

- Occupancy is very low as we are memory bound. Reducing to FP16 increases the speed but that is due to the reduced memory overhead.

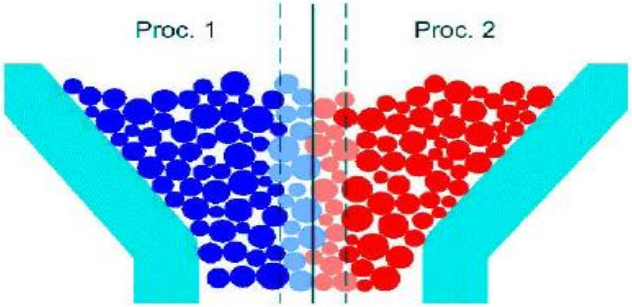
- Have to find a way to eliminate the use of local arrays for the storage of computed contact points.
  - Since each particle pair has to do this having it directly in global memory and then splitting the computation does reduce divergence and increase speed but the memory cost is far to great.

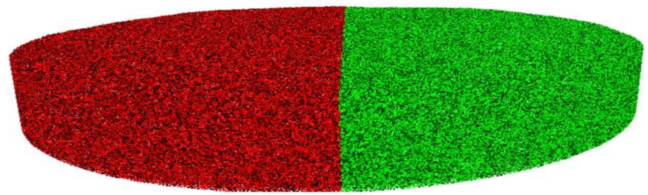
- Since occupancy is already low, we can manually launch the waves of blocks on the GPU.

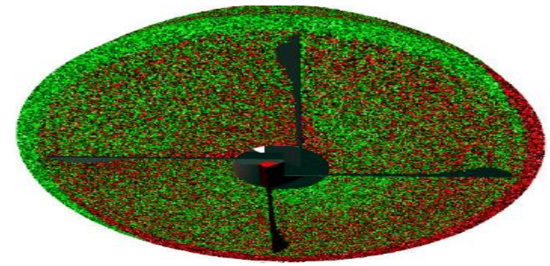
Govender et al. (2018) FD Jacobian solver for heat transfer between bodies.

MULTI GPU

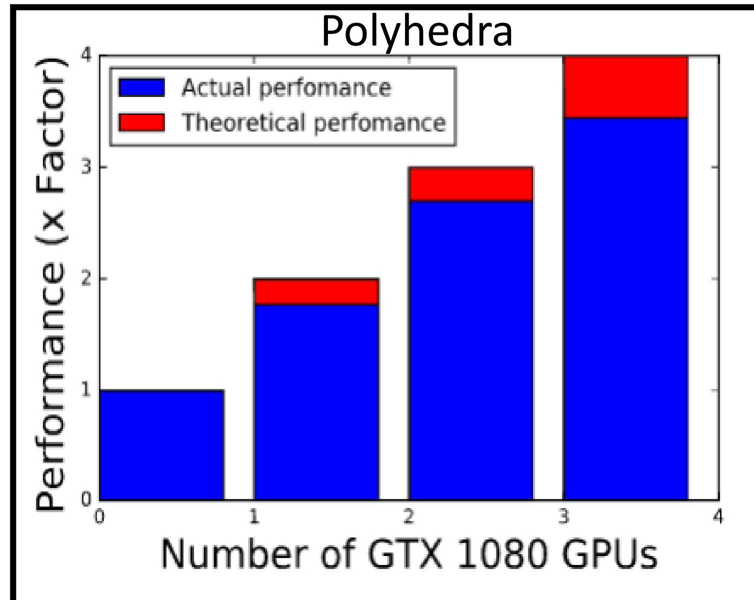
- Classical domain decomposition is not general enough for DEM as particles are dynamic creating load balancing issues.







- On a single node don’t need OpenMP, cudaPeer is sufficient.
- Polyhedra have sufficient compute to hide data transfer even when all data is transferred.
- Bi-direction bandwidth can be exploited.
- Compute for spheres is faster than hardware bandwidth. Such an approach cannot work.
- Rocky for example uses domain decomp for spheres with scaling > 1mil. However, they are 5x slower than us so scaling is apparent due to a slower compute…



Coming soon a novel order and bucket multi-gpu approach for arbitrary domain's and particle shapes.

THE UNIVERSITY OF UTAH™

UNIVERSITY OF SURREY

UNIVERSITY OF JOHANNESBURG

University of Pretoria

MINES Douai

LGCgE Laboratoire Génie Civil et géo-Environnement Lille Nord de France

UTCB Technical University of Civil Engineering Bucharest

CSIR our future through science

BLAZE

POWERED BY NVIDIA

UNIVERSITY OF SPLIT

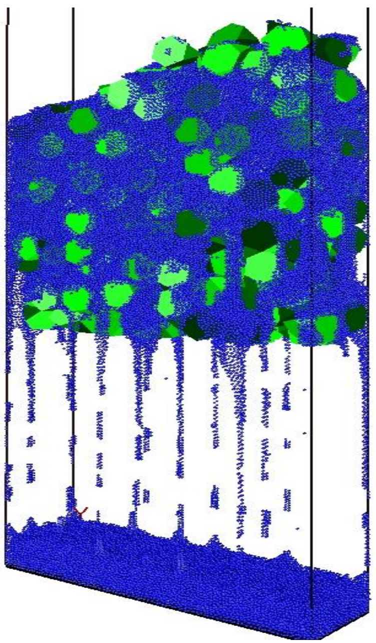
MONASH University

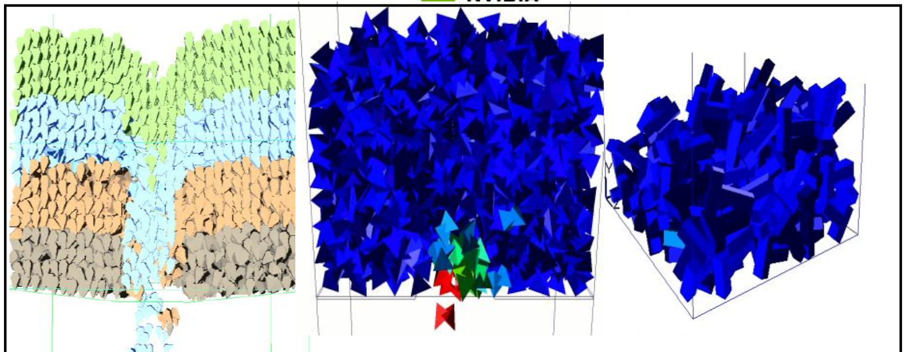
THE UNIVERSITY OF QUEENSLAND AUSTRALIA

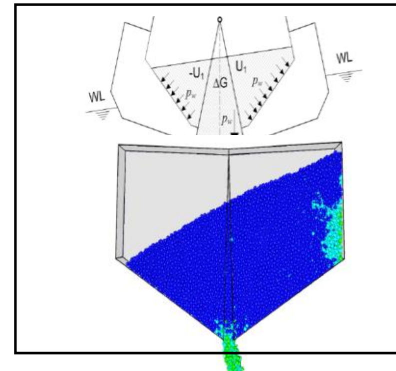
RUTGERS THE STATE UNIVERSITY OF NEW JERSEY

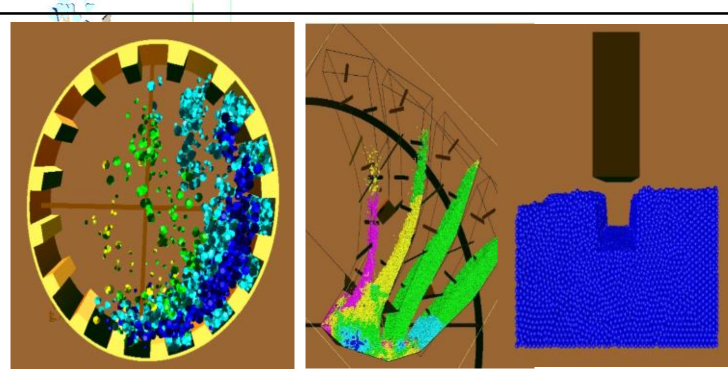
UNIVERSITY OF BRITISH COLUMBIA

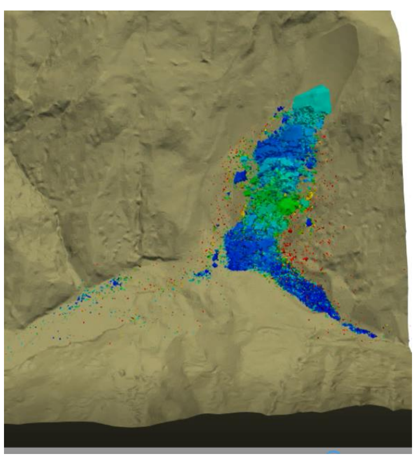
LUT Lappeenranta University of Technology

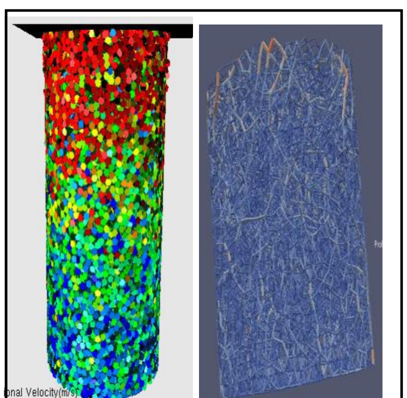


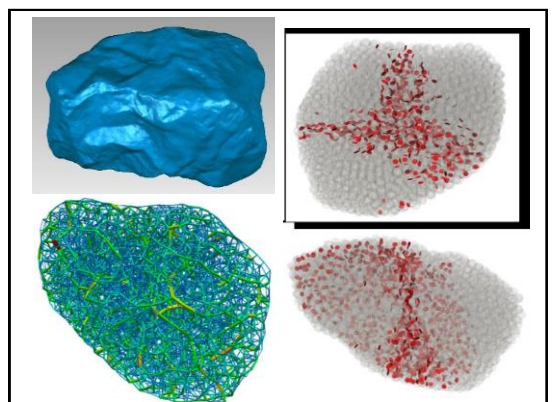


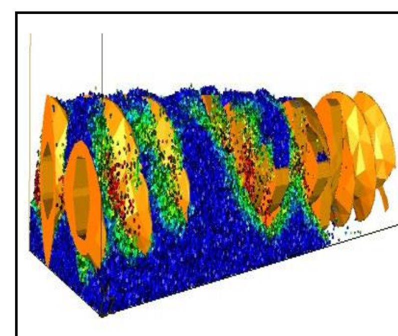






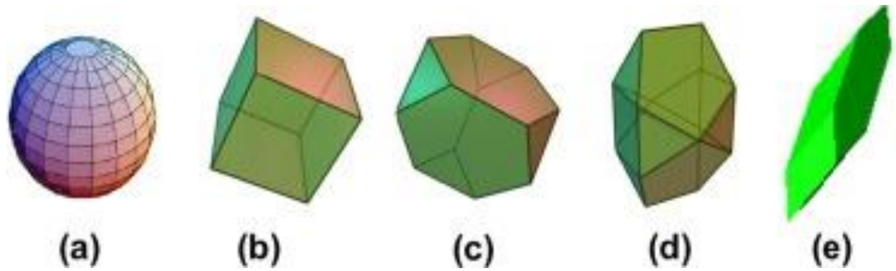


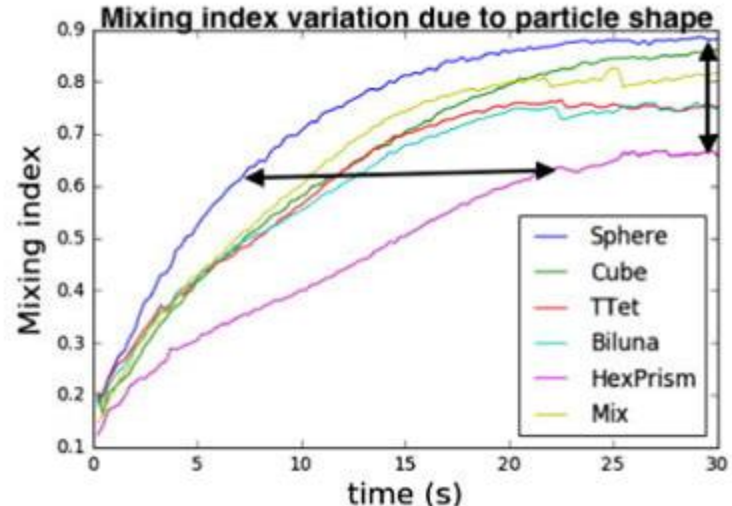


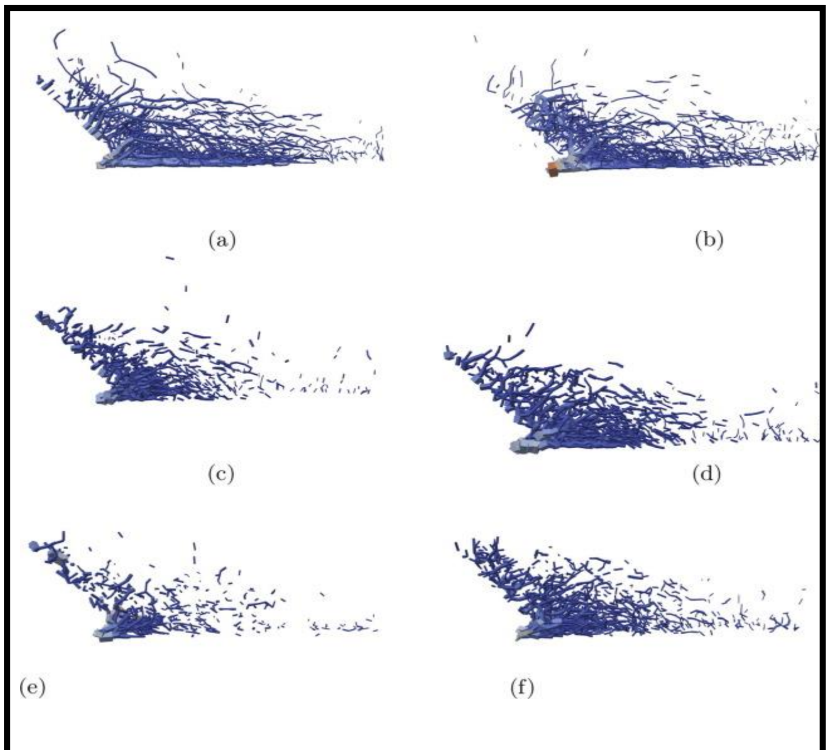


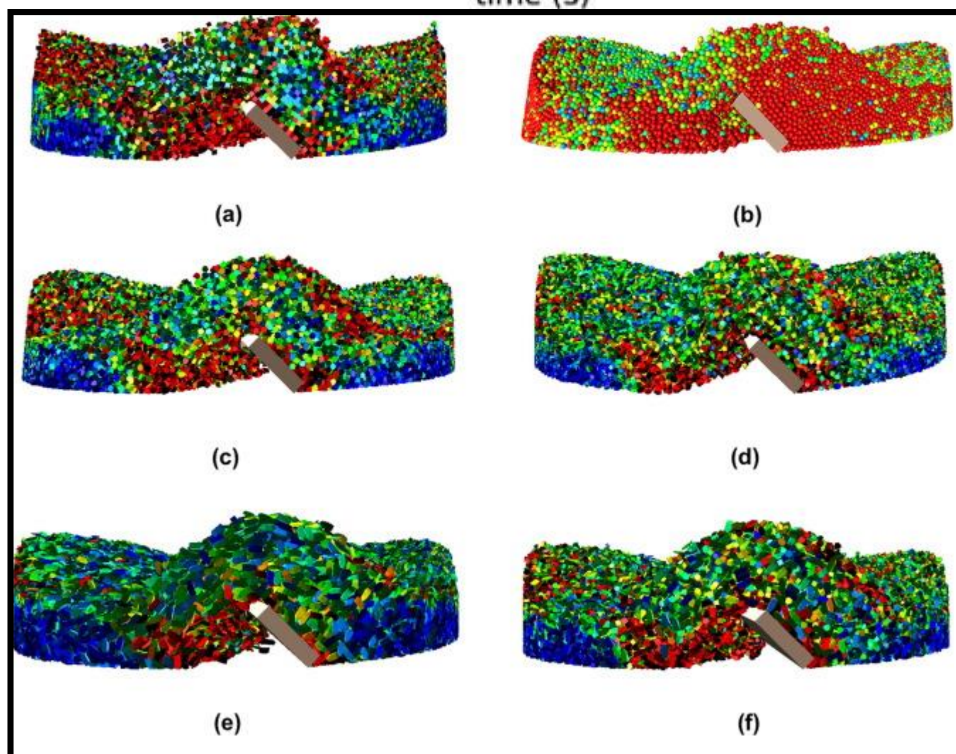
Assumption 1:
Do we really need shape

Granular Mixing







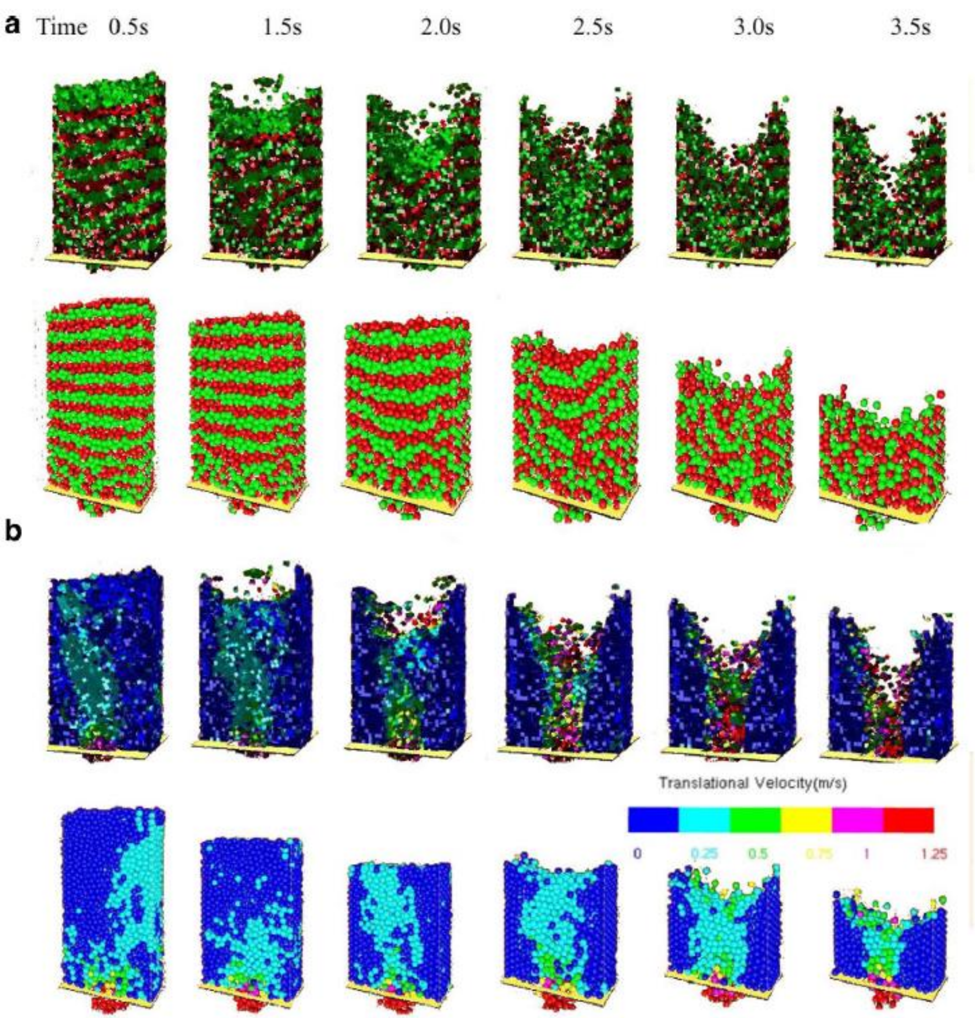


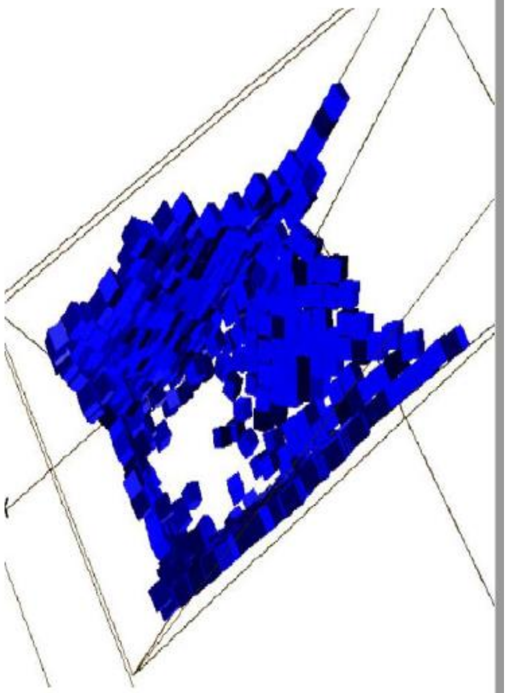
[1] Large-scale GPU based DEM modeling of mixing using irregularly shaped particles, Advanced Powder Tech. (2018)

Spheres are fine, we add "rolling friction"

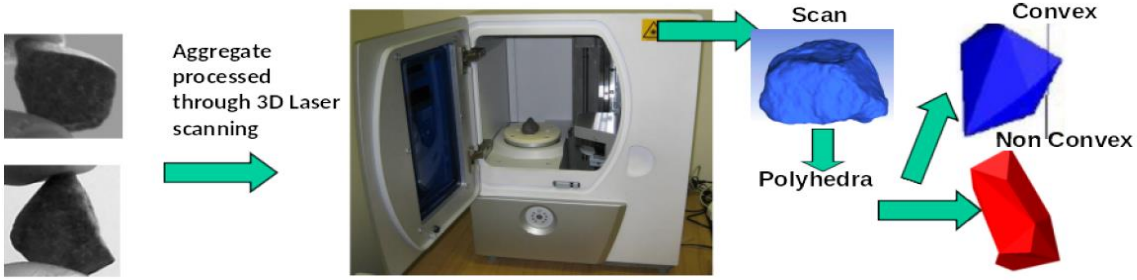
Still

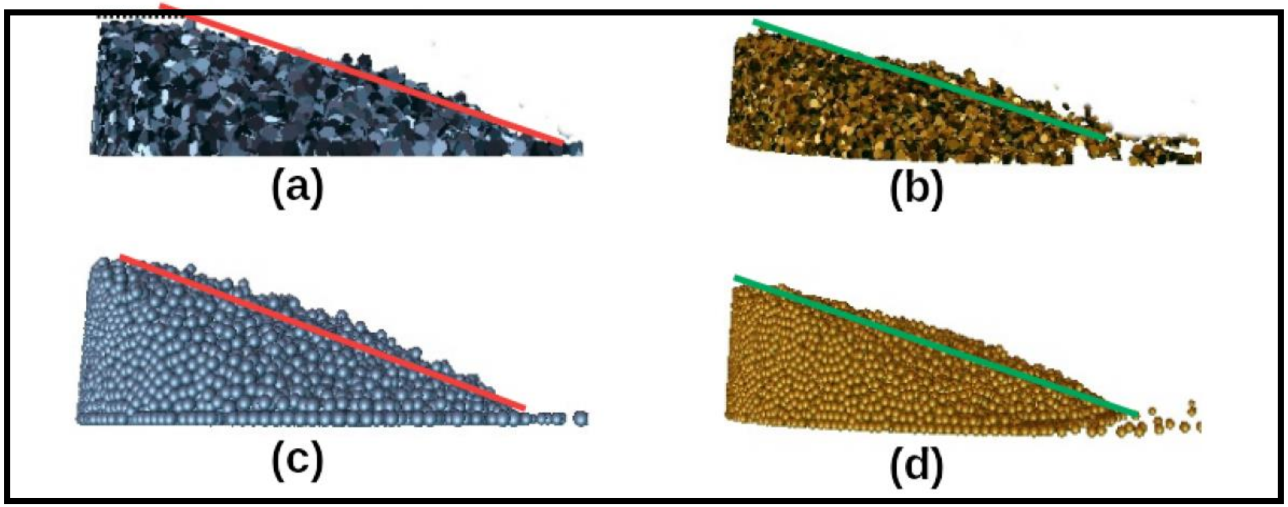
Can rolling friction with spheres capture complex behavior such as arching ?

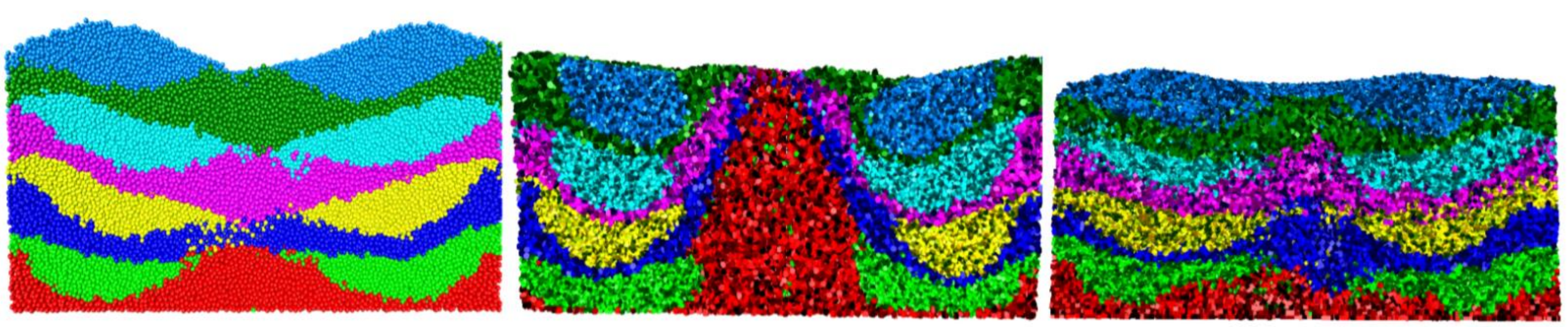




To what extend does rolling friction mimic shape?

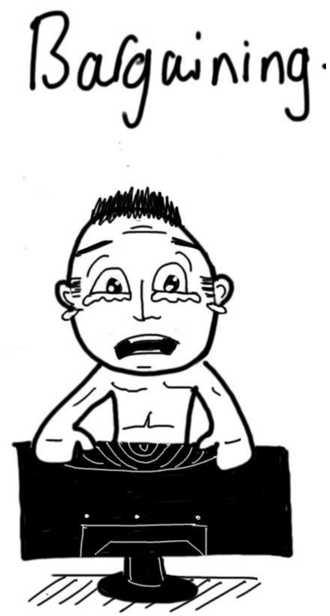




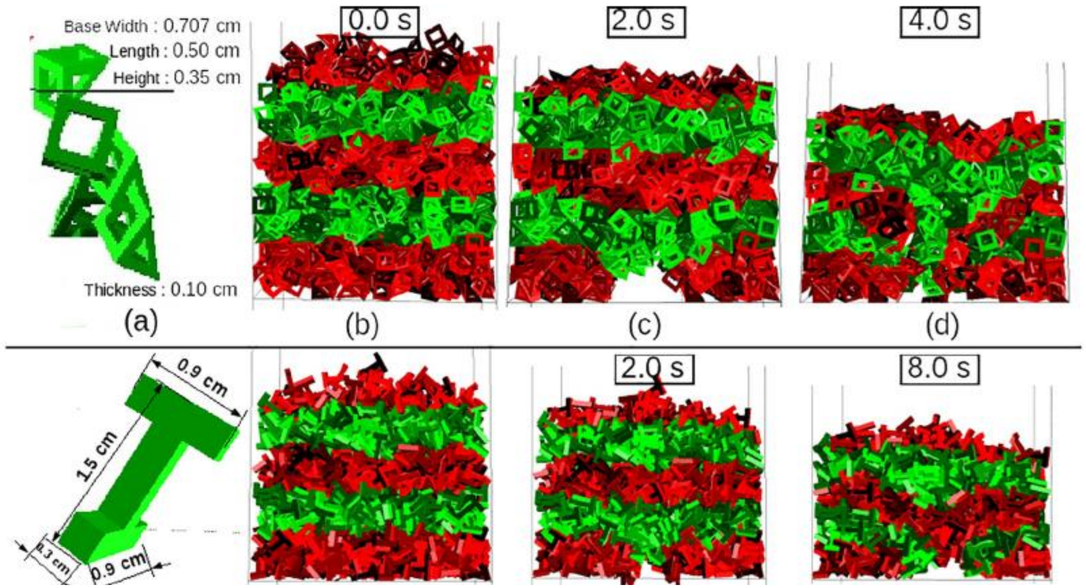


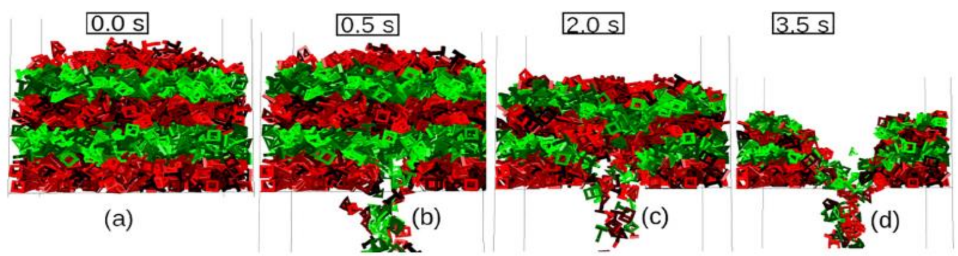
(a) (b) (c)

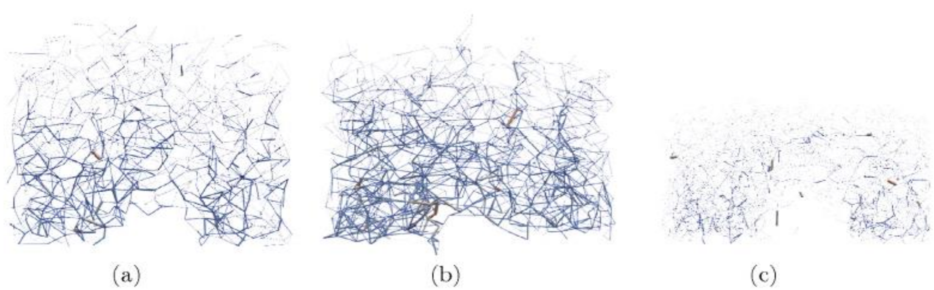
Assumption 2:
Ok we can stick our spheres to get non-spherical shapes.



Can we do this with spheres or clumped spheres ?

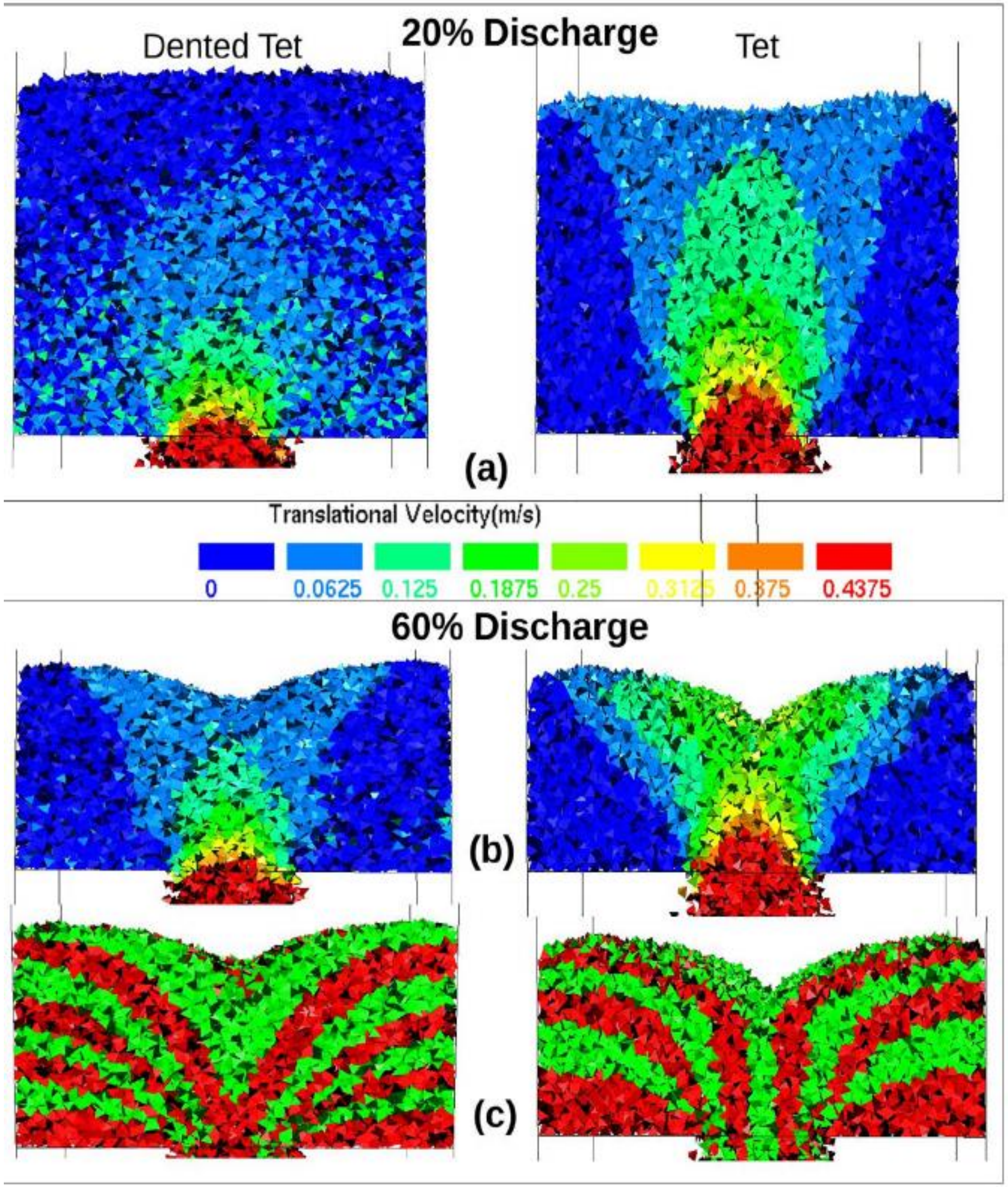






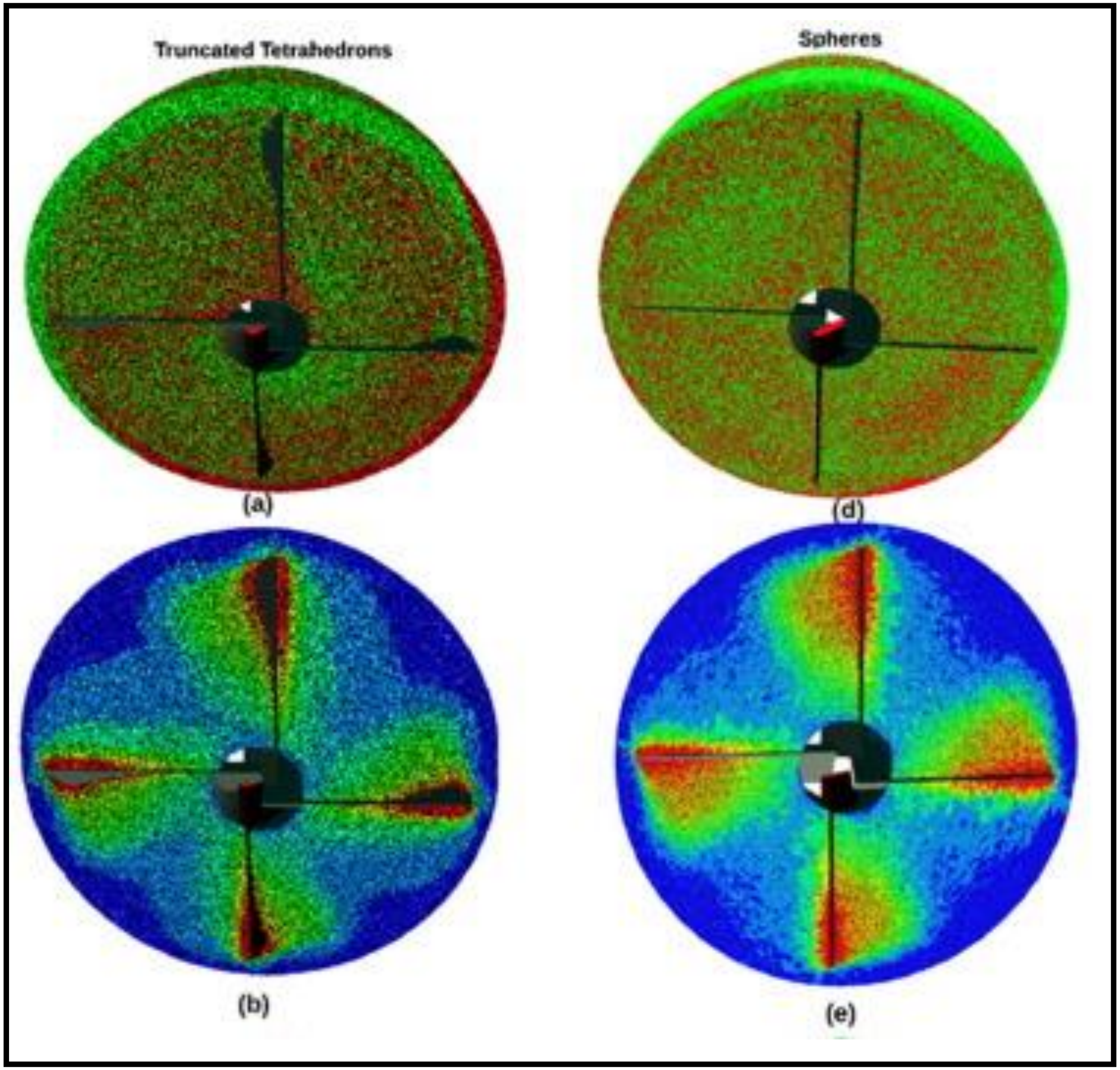
**Assumption 3:
Ok but it does not matter on the larger scale.**

Do we still get shape effects for large scale ?



UNIVERSITY OF SURREY

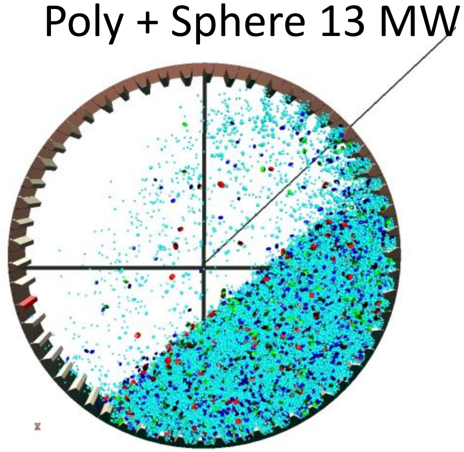
Do we still get shape effects for large scale ?



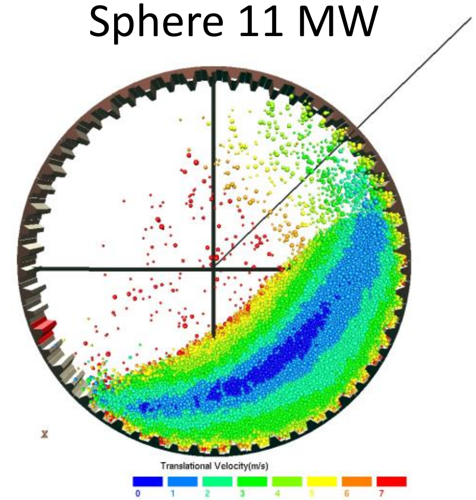
UNIVERSITY OF SURREY

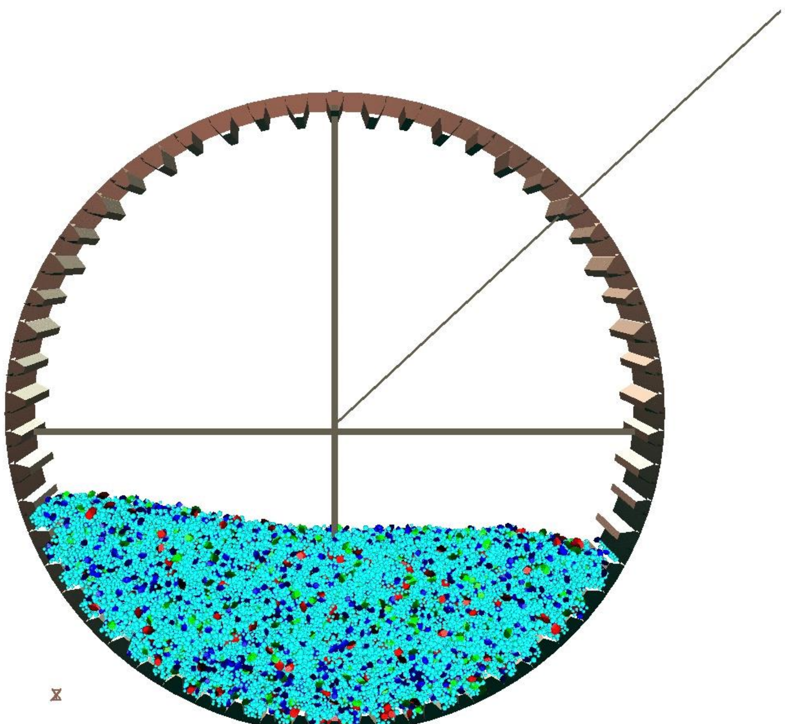
Do we still get shape effects for large scale ?

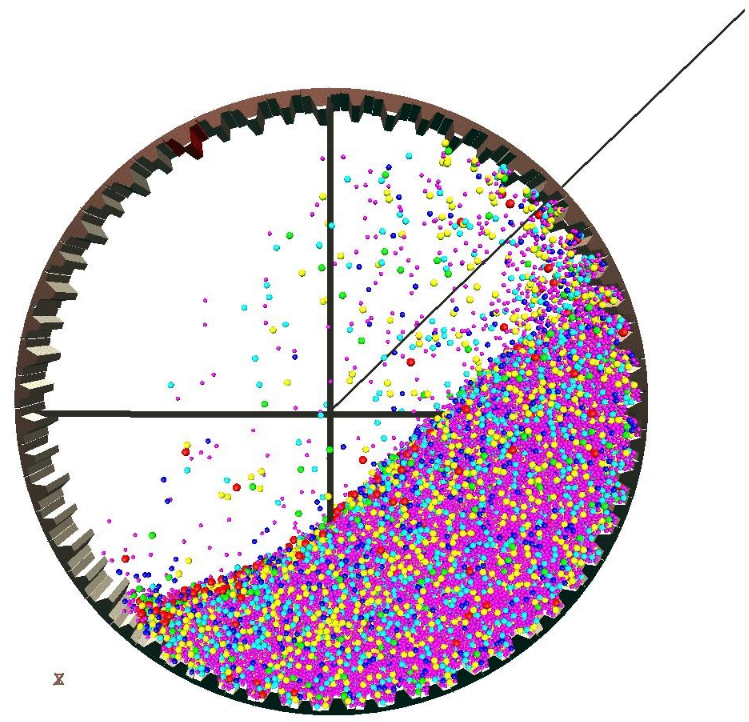
Poly + Sphere 13 MW



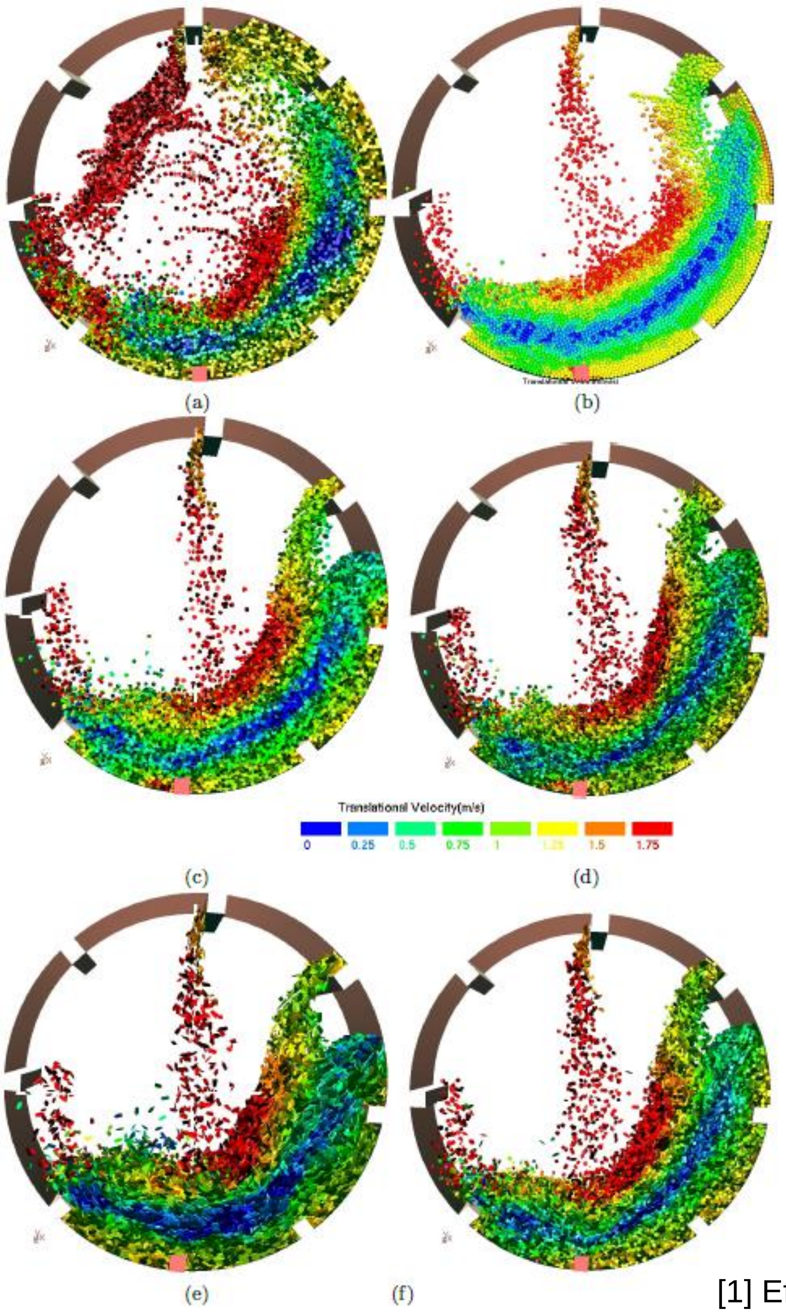
Sphere 11 MW







Milling



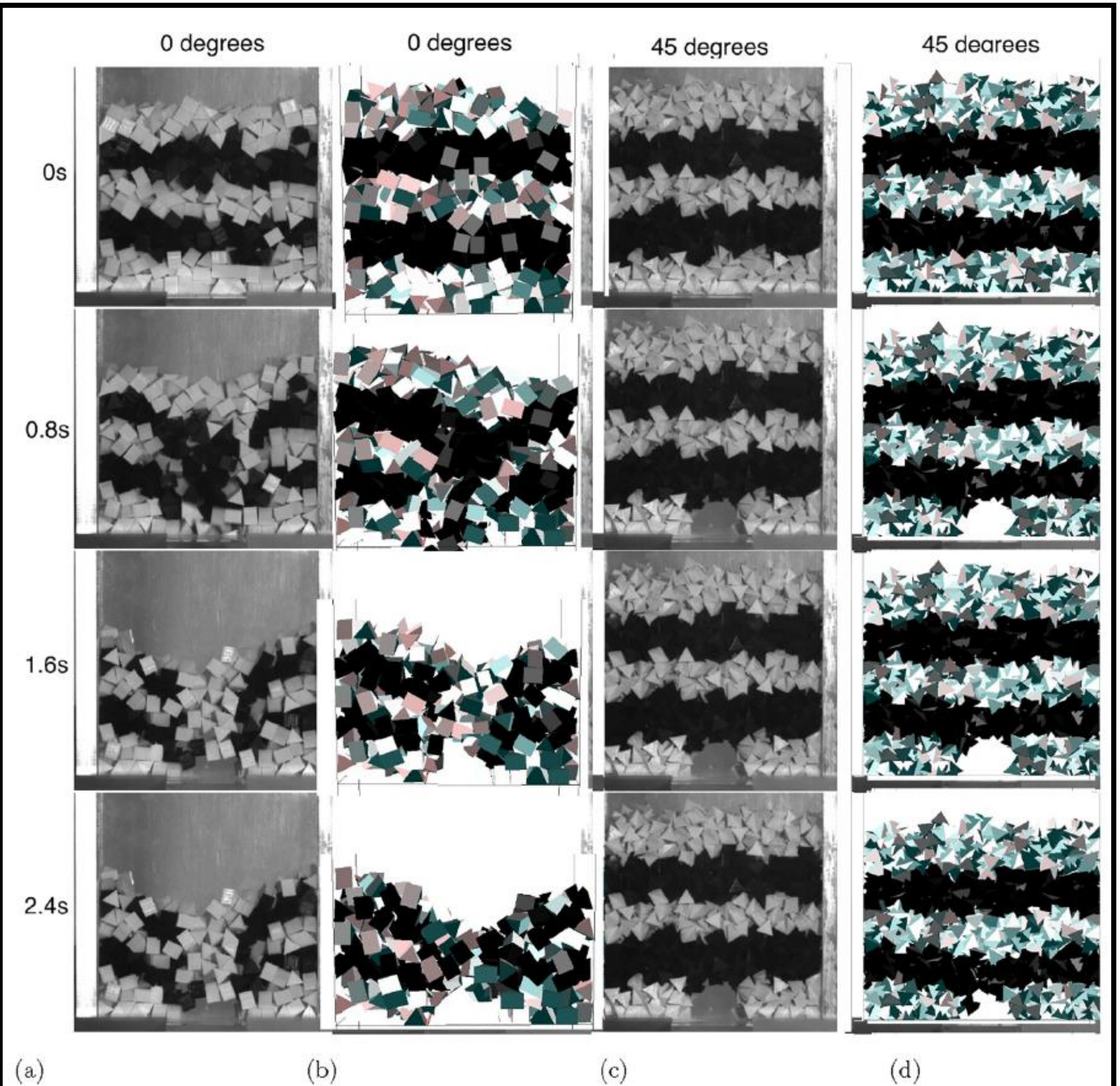
Flow Profile and Energy consumption

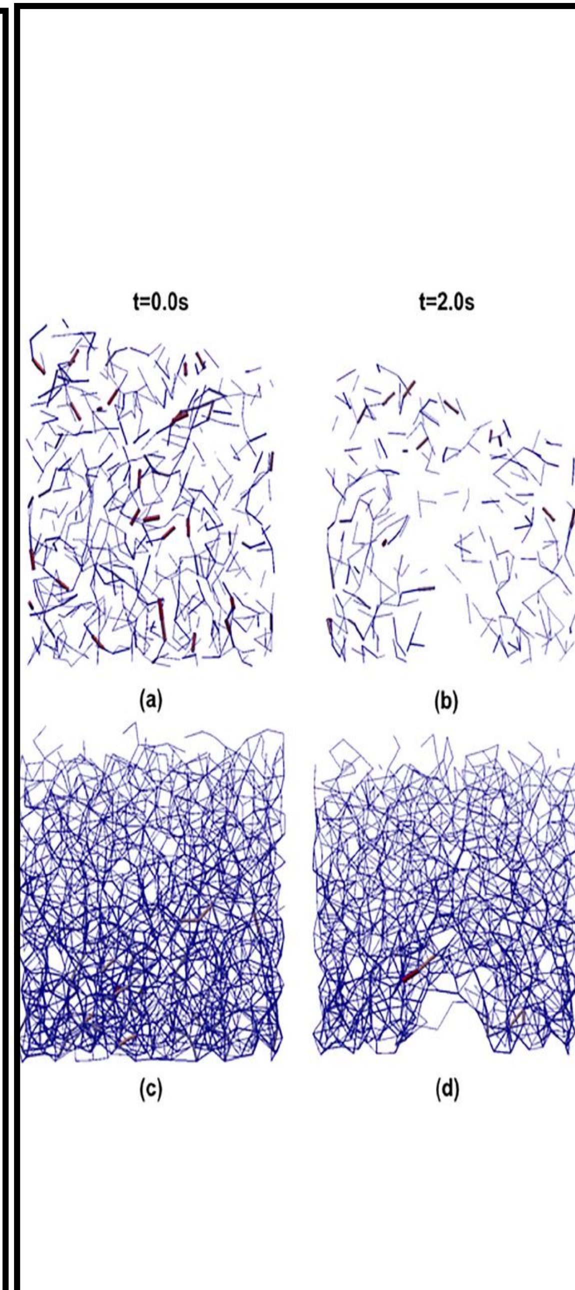
Particle Shape \ Units (%)	% PartN	% PartS	% LiftN	% ShellN
Sphere	32	**38**	**21**	9
Cube	**68**	7	8	**15**
TTet	57	23	11	7
Biluna	51	20	16	11
HexPrism	60	19	12	8
Mix	57	20	12	9

[1] Effect of particle shape on milling, Minerals Engineering (2018)

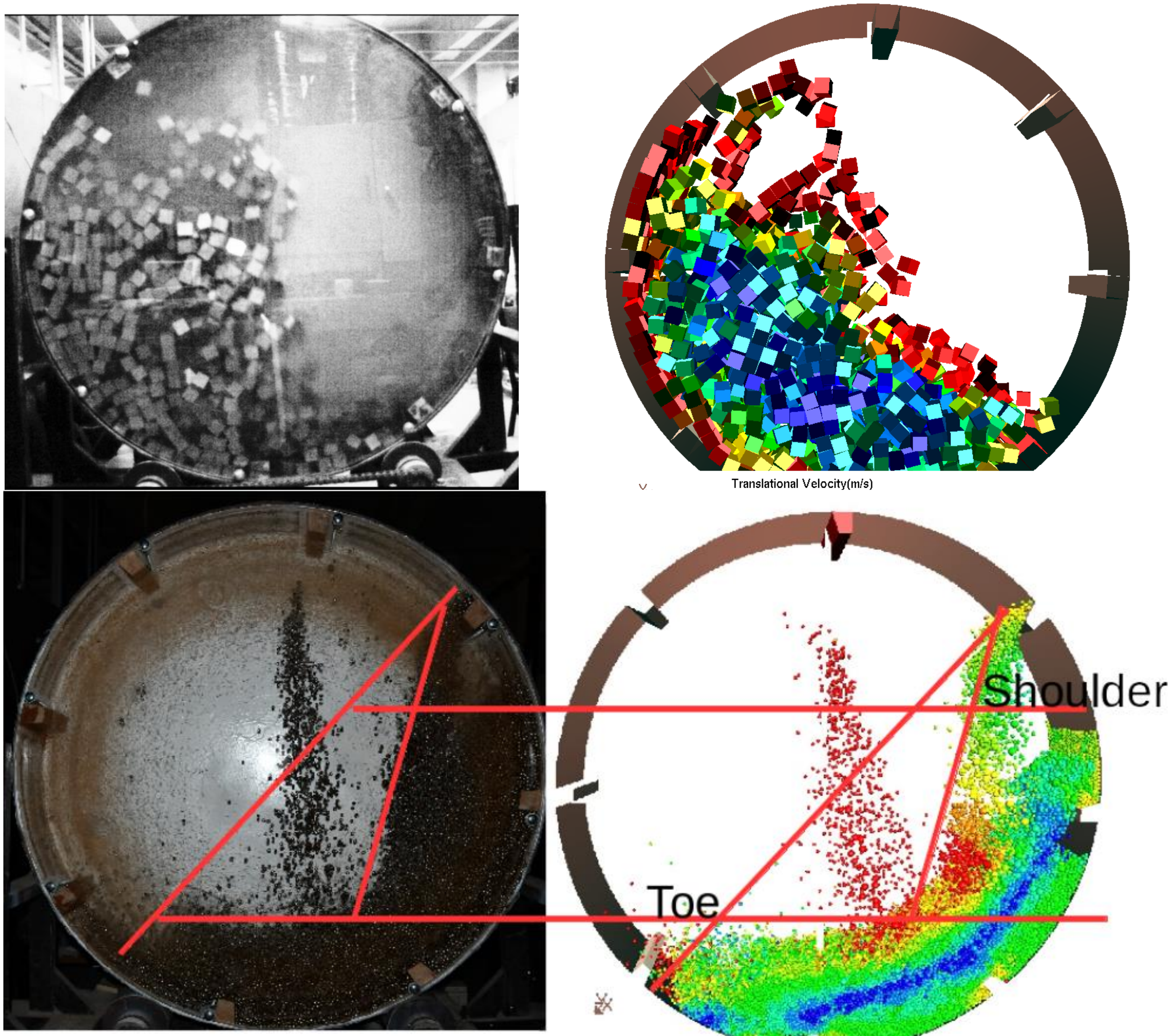
Ok this GPU thing its for games not real science right ?

Test 1: Contact stability



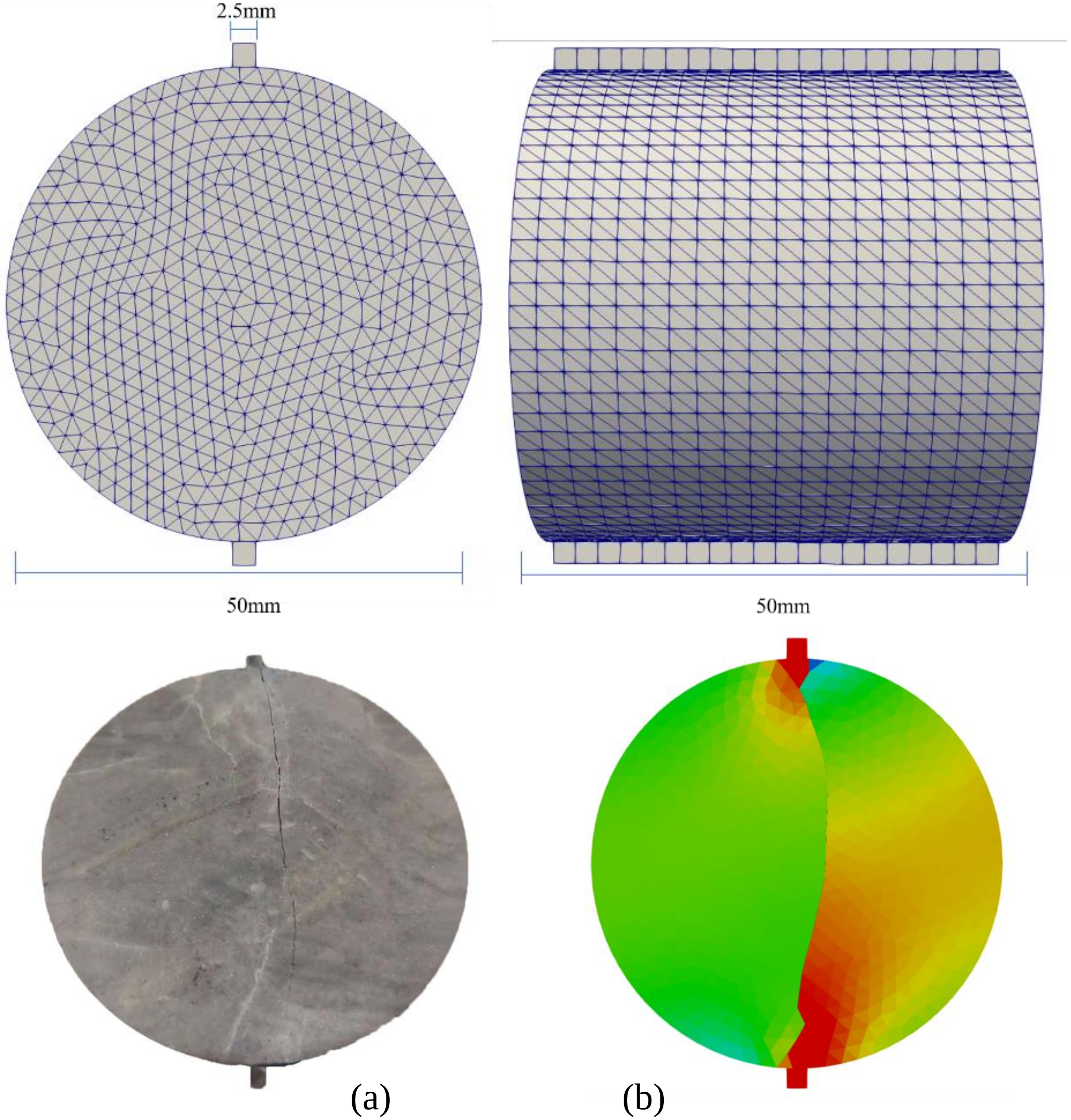


Test 2: Dynamic Motion

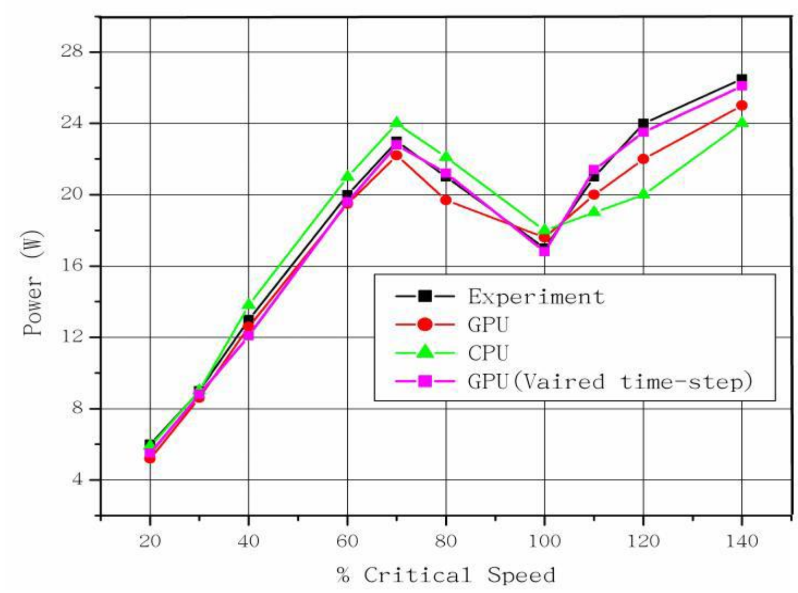


Test 3: For good measure typical FEM problem

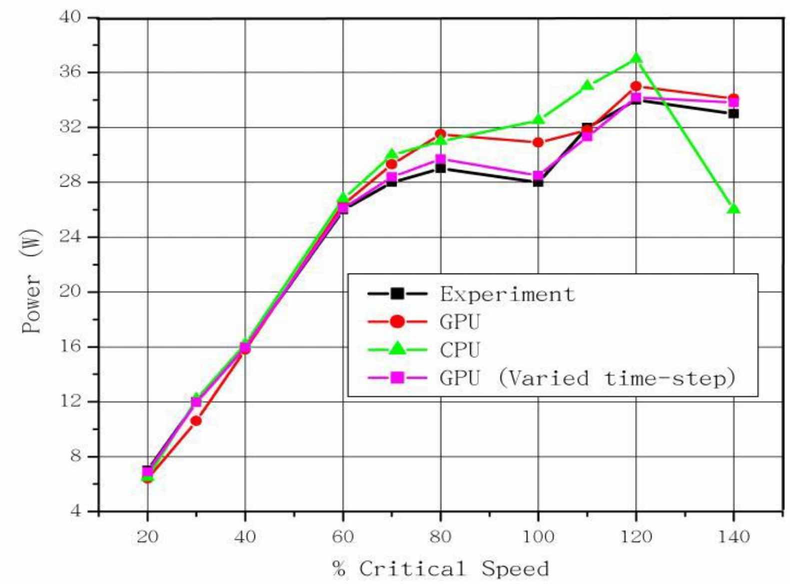
Modeled in Blaze-DEM as bonded polyhedra



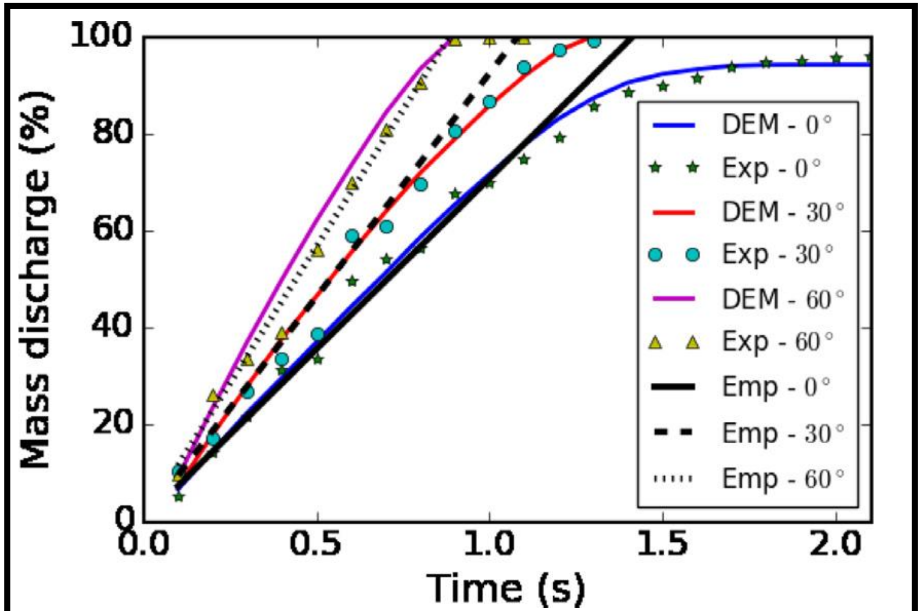
Test 4: Not just pretty pictures..

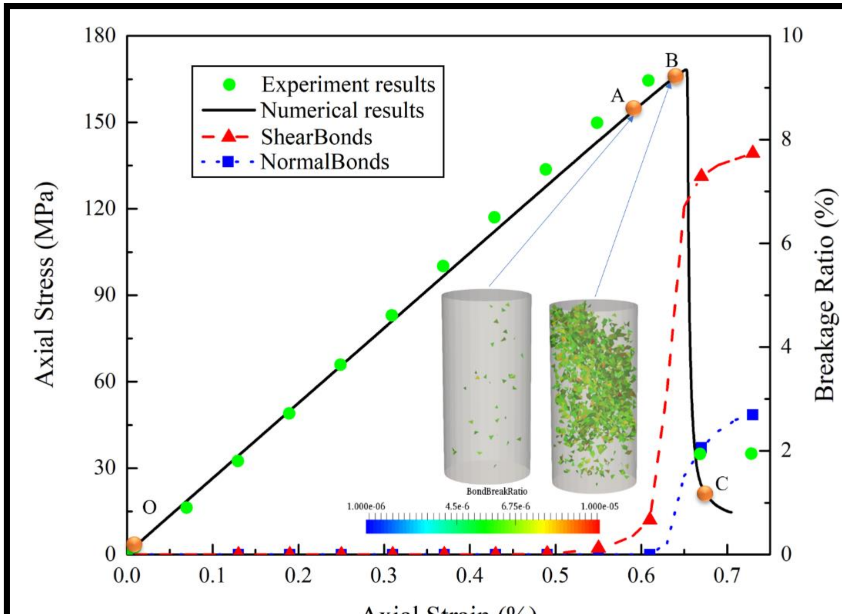


(a)



(b)



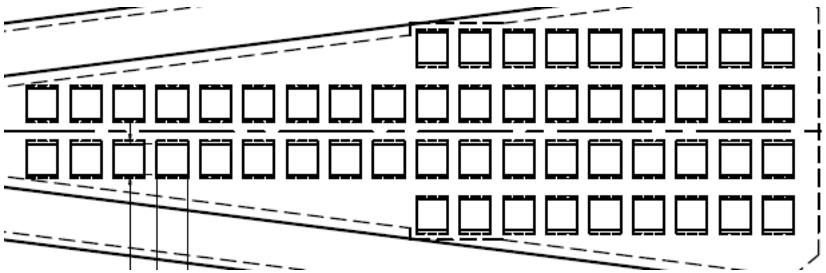


Finally

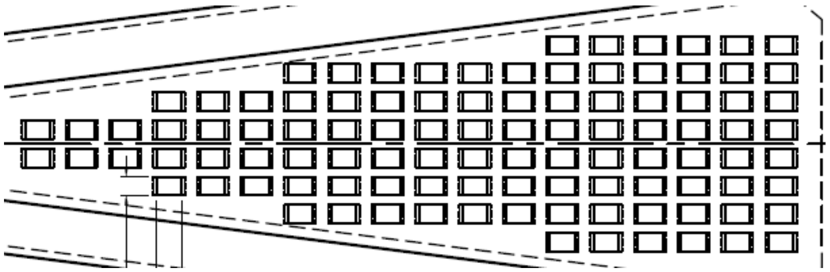
Disclaimer: No CPU programmers where harmed during the making of these slides.

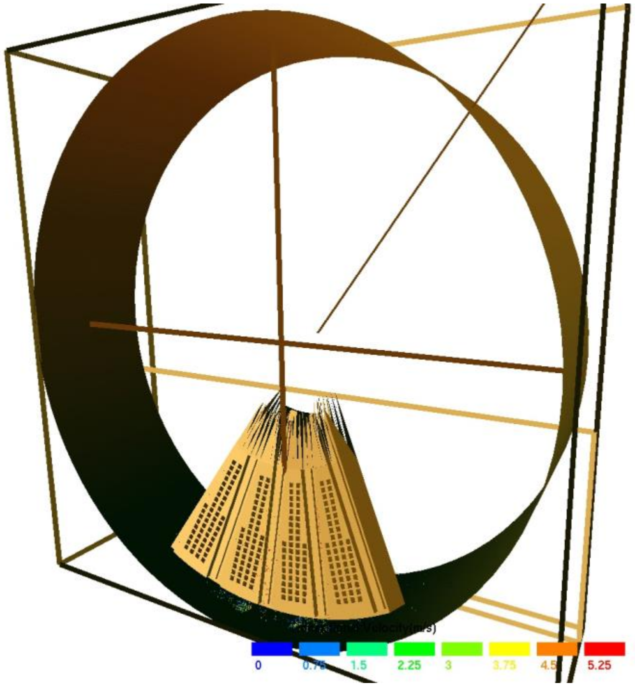
DESIGN EVALUATION

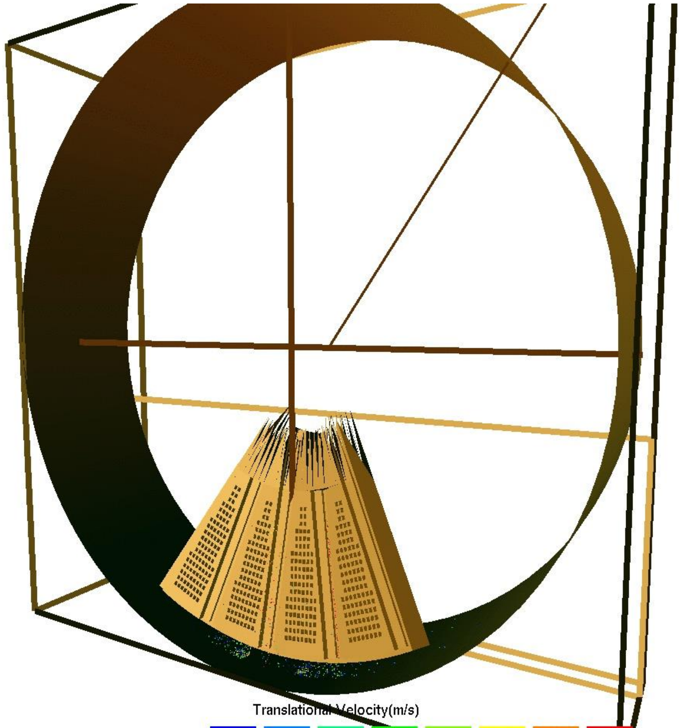
A



B







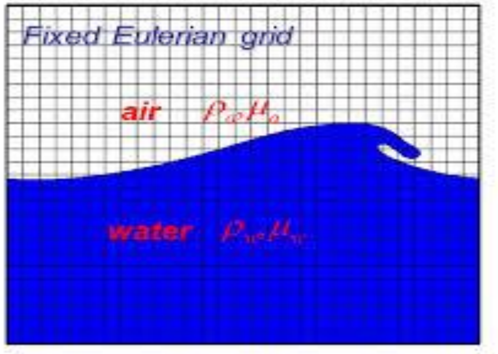
- 30x40 grate slots give a 10% higher flow rate through the discharger. 8% less backflow and 5% less carry over flow

Coupling with Fluid

- A large number of industrial processes requires both particulate matter and liquid/air to be simulated.

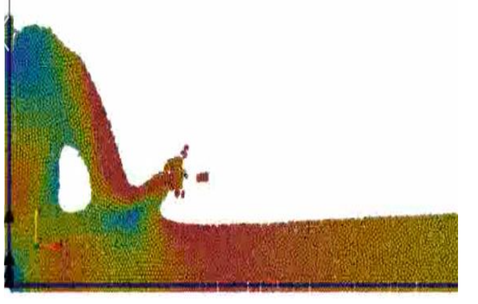
- CFD(VOF) is the most common method for the simulation of fluid, unfortunately apart from a few specific cases it does not fit the GPU model.

- LBM is similar in spirit to CFD however it has a fixed number of propagation directions in each node making it well suited to GPU implementations.



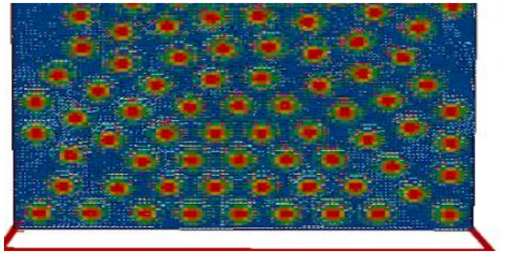
- A weakness of LBM/CFD and grid based methods in general is that free surfaces requires additional computation and memory.

- Mesh free methods like SPH are by far the most suited to the GPU as the fluid is represented by particles. The free surface is also "free". Most popular for games/animations.



- However SPH is oth order accurate making its use in scientific applications limited.

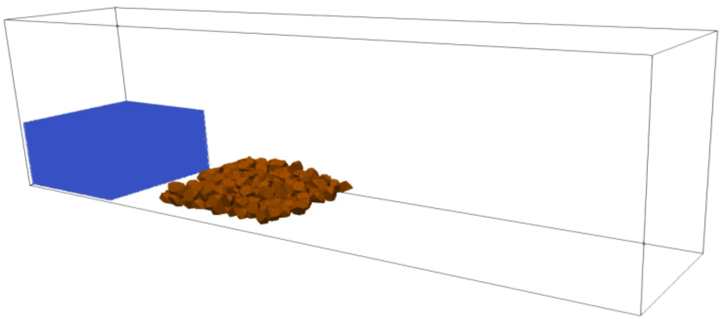
- Particles treated as a porous medium, unresolved flow around the particles/structure. Drag models are needed, which still do not capture shape effects correctly.

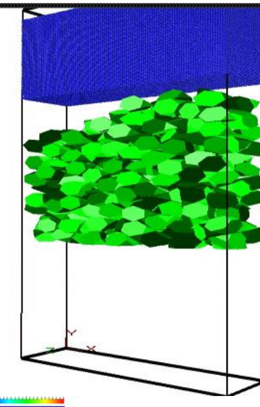


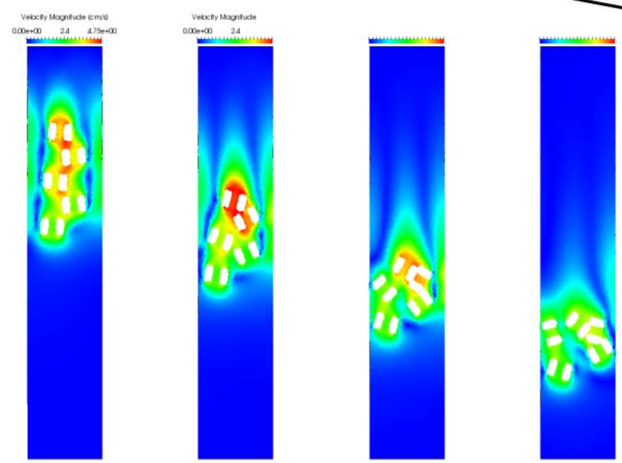
MULTI-PHYSICS COUPLINGS

DualSPHysics : Unresolved

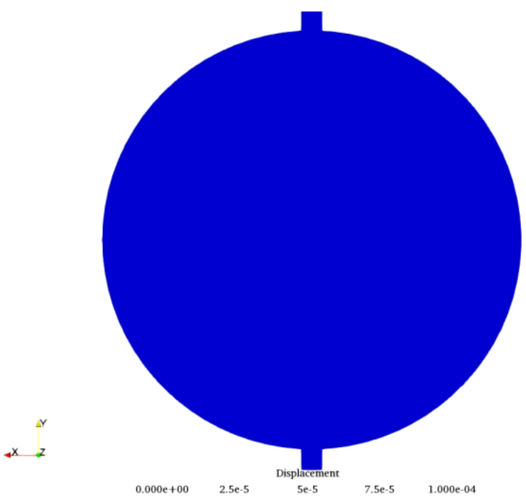
Blaze SPH : Resolved 1st order gradient correction

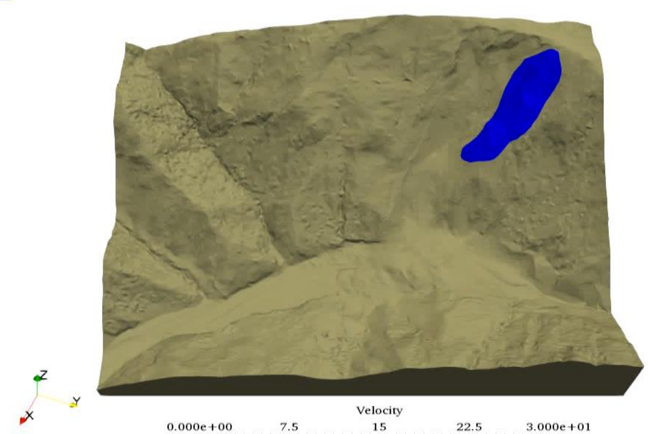






(a) (b) (c) (d)





CONCLUSIONS

- DEM simulations using the GPU computing is at the same physics fidelity as CPU based codes.

- The increase in computational power gives us a large number of spherical particles many times faster than CPU codes.

- The increase in computational power is used to do shape more accurately than CPU based codes while being faster and allowing for millions of particles.

- The effect of particle shape is evident.

- Blaze-DEM is open-source to collaborators, have a look at researchgate.

- Submit an abstract for DEM 8, Sessions on particle shape and GPU/HPC .

- Always welcome GPU donations.

A man's reach should exceed his grasp, or what are GPUs for…

UNIVERSITY OF SURREY