6th DualSPHysics Workshop

UPC – Barcelona, Spain

October 26, 2022

Support in Project Chrono for simulating automation, robots, and autonomous vehicles

Dan Negrut

NVIDIA CUDA Fellow

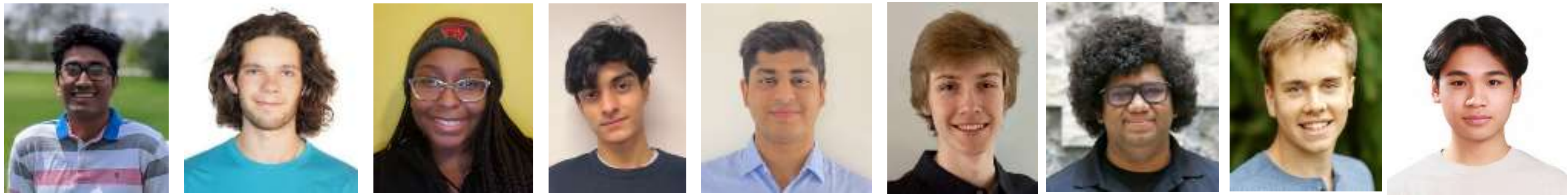
Bernard A. and Frances M. Weideman Professor of Mechanical Engineering

University of Wisconsin-Madison

Simulation-Based Engineering Lab members & collaborators

(left to right, based on how long in the lab)

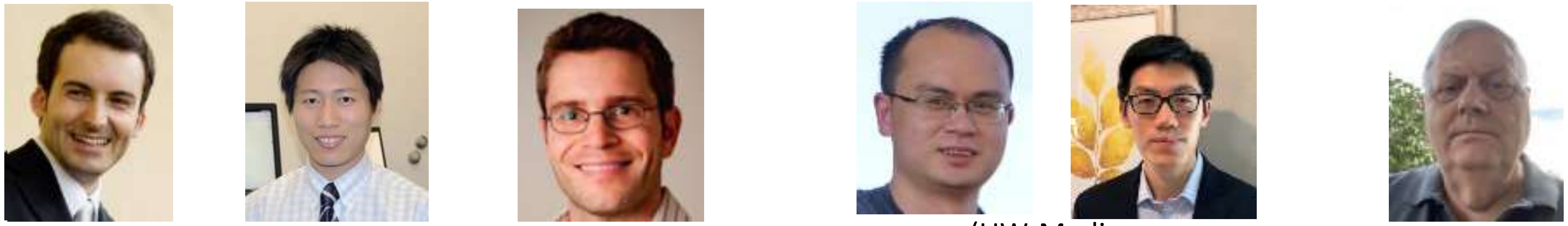
Undergrad Students



Grad Students

Non-students

Collaborators, from outside lab



(Italy, for Chrono) (CSULA, Robotics) (MIT, Terradynamics) (UW-Madison, Controls & Data Science) (Germany, for Chrono)

Funding sources, ongoing

- US Army Research Office
  - Basic research in terradynamics
- US Army DURIP
  - Instrumentation, for HPC & GPU computing
- National Science Foundation
  - Simulation Engine development
- National Science Foundation
  - AV simulation + Human-in-the-loop simulation
- DoD HPC Modernization Program
  - Vehicle Dynamics simulation
- NASA
  - 2023 VIPER lunar mission
  - Lunar human habitat
  - Perception in harsh lunar environments
- Hexagon/MSC.Software
- Disney Research
- Blue River/John Deere

engineering innovation & scientific discovery → fueled by good quality data

- Data: by and large, comes from measurement/sensing

- Our lab's research goal: increase the % of data that has simulation as its provenance

- Simulation → our focus is on automation, robotics , AVs, human-robot interaction

- We are not developing next gen robots or AVs

- We seek to produce models & numerical methods & software for computer simulation to be instrumental in designing the next gen robots or AVs

- Robotics: Better engineering through better simulation

Similar outfits…

- Gazebo
- IsaacSim (NVIDIA)
- [CARLA]

What are we interested in simulating?

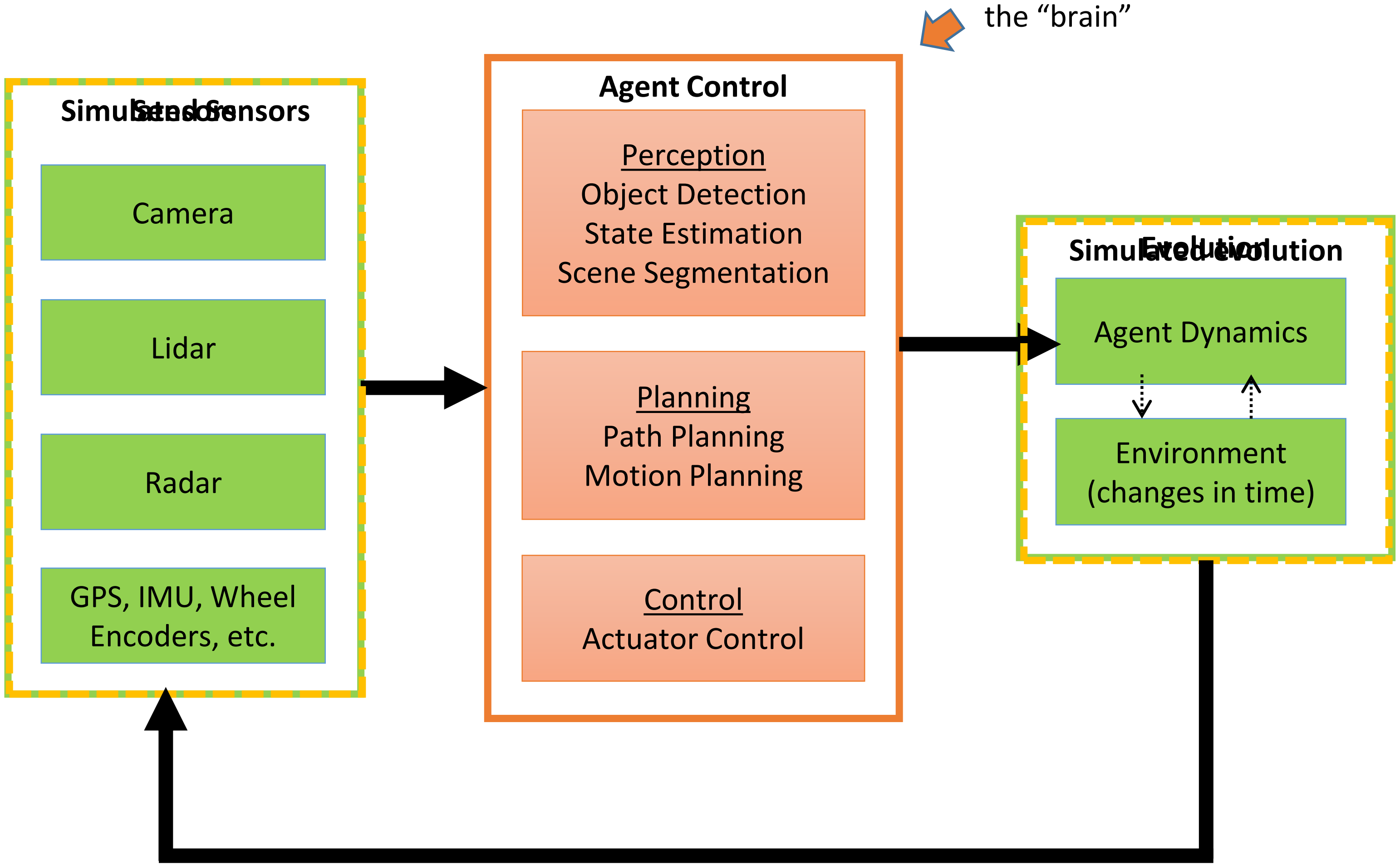
- Simulate the process of sensing
- Simulate the robot/rover/autonomous vehicle
- Simulate the world in which the robot/rover/autonomous vehicle operates

Why are we interested in these things?

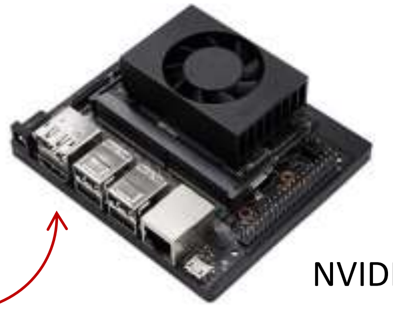
Questions that can be answered quickly in simulation

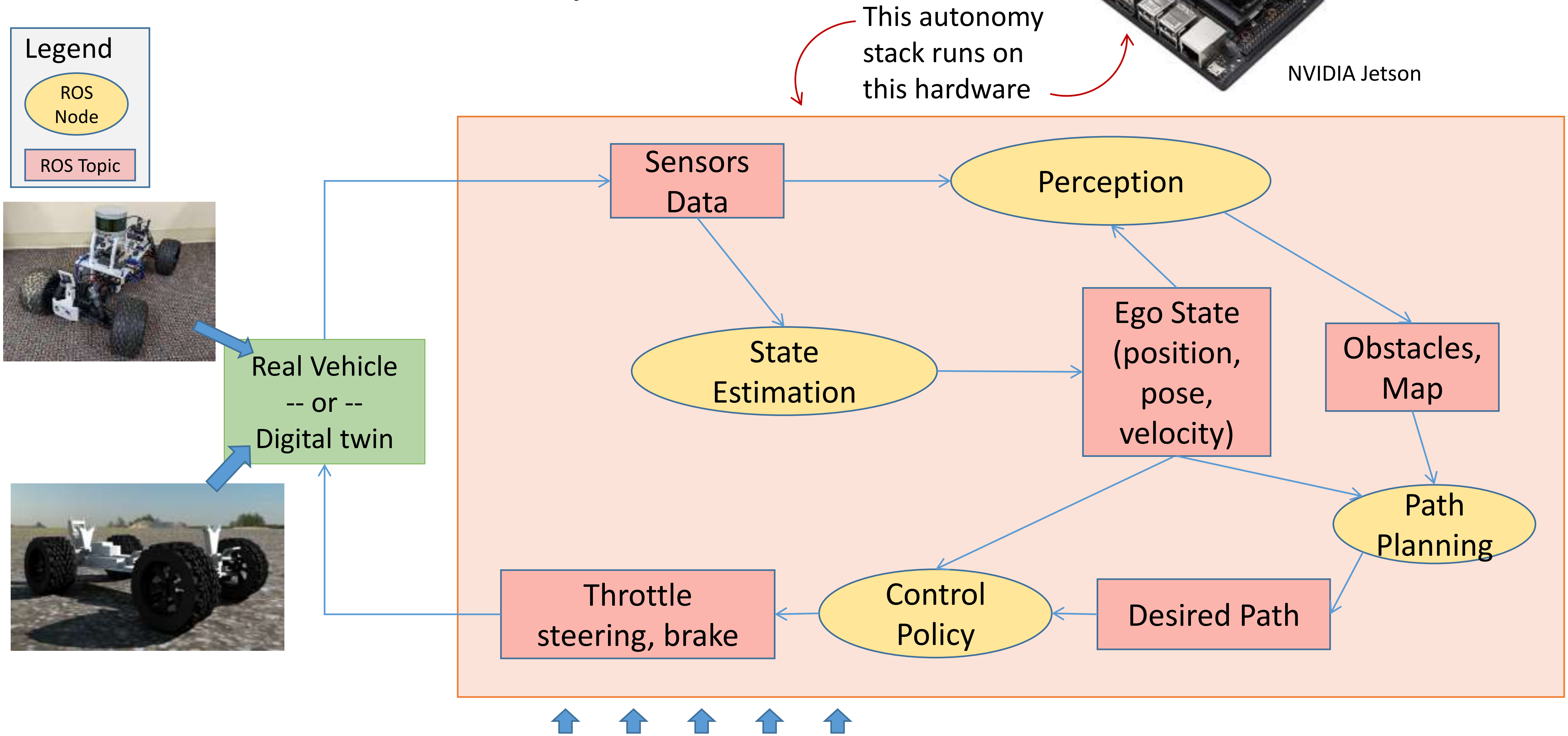
- Is this control policy better than the other one?
- Is this state estimation better than the other one?
- Is SLAM algorithm better than the other one?
- How long does it take a chip (Intel Nuc, Jetson, R-Pie) to handle the ROS2 autonomy stack?
- How good is this visual odometry algorithm?
- Etc.

Simulation of robots/AV: 30,000 feet picture



The ART/ATK Autonomy Stack





This autonomy stack is the same, regardless of whether actual or virtual vehicle used

Going on a tangent: The chip on *Perseverance* → RAD750

- Radiation-hardened single-board computer manufactured by BAE Systems (approx. $300,000 apiece)
- Launched in 2001
- 110-200 MHz
- ISA: PowerPC 1.1
- Technology: 200 nm or so
- Number of transistors: about 10.5 millions
- Caches: L1 only (I$ - 32KB; D$ - 32KB)

Credit: https://en.wikipedia.org/wiki/RAD750

SBEL SIMULATION BASED ENGINEERING LAB
UNIVERSITY OF WISCONSIN - MADISON

Autonomy Research Testbed

So again, what are we interested in simulating?

- Simulate the process of sensing
- Simulate the robot/rover/autonomous vehicle
- Simulate the world in which the robot/rover/autonomous vehicle operates

The world that we simulate doesn't exist. We have to create it.

Another place where folks create fictitious worlds: computer graphics

One slide side trip: Computer Graphics vs. Computer Vision

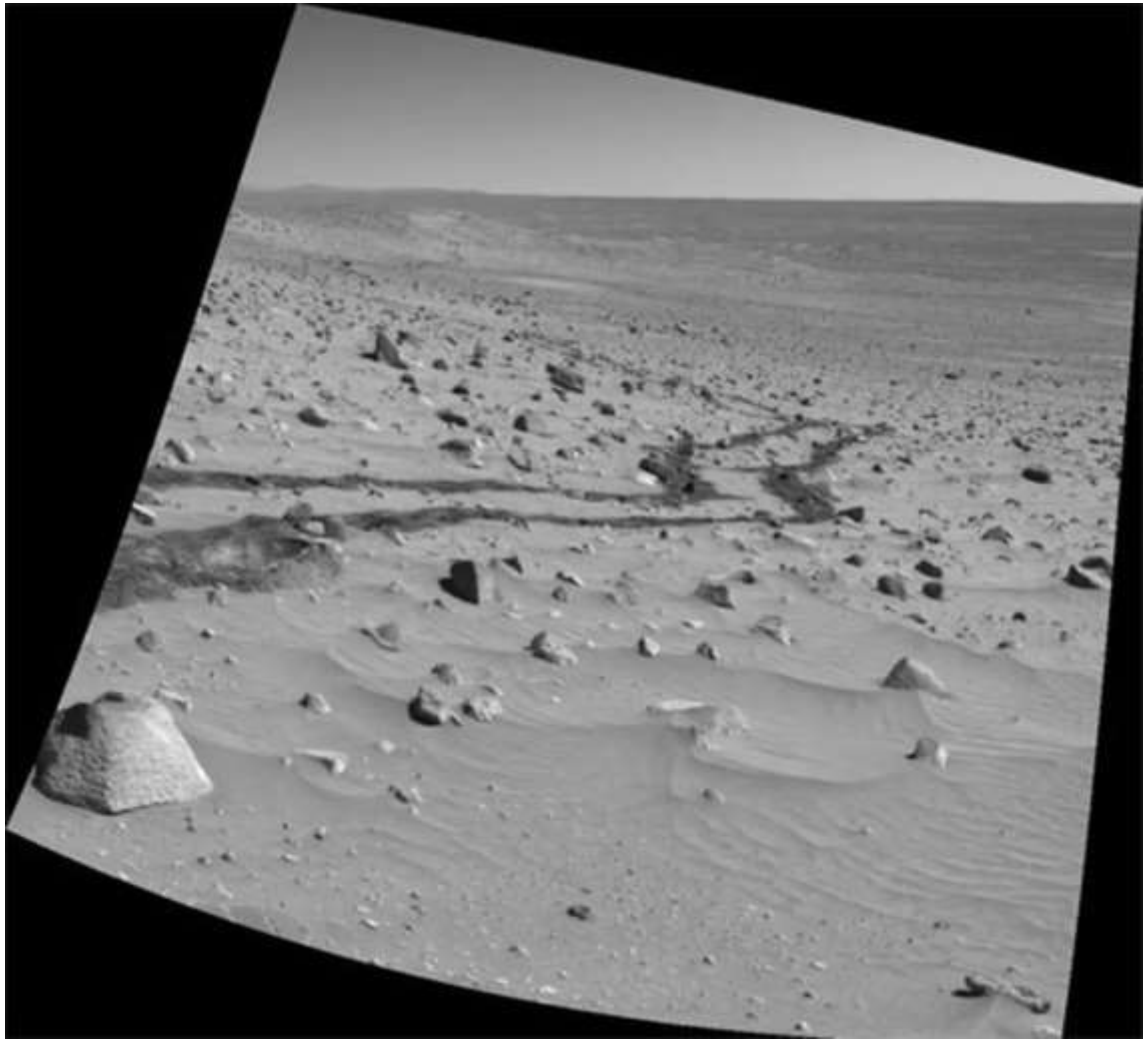
- Computer Graphics:
  - Input: a bunch of virtual assets (mesh of table, mesh of chairs, etc.) that make up virtual world
  - Output: an image
- Computer Vision:
  - Input: an image
  - Output: a bunch of objects & features picked up by looking at the image
- Computer Graphics & Computer Vision: are complementary

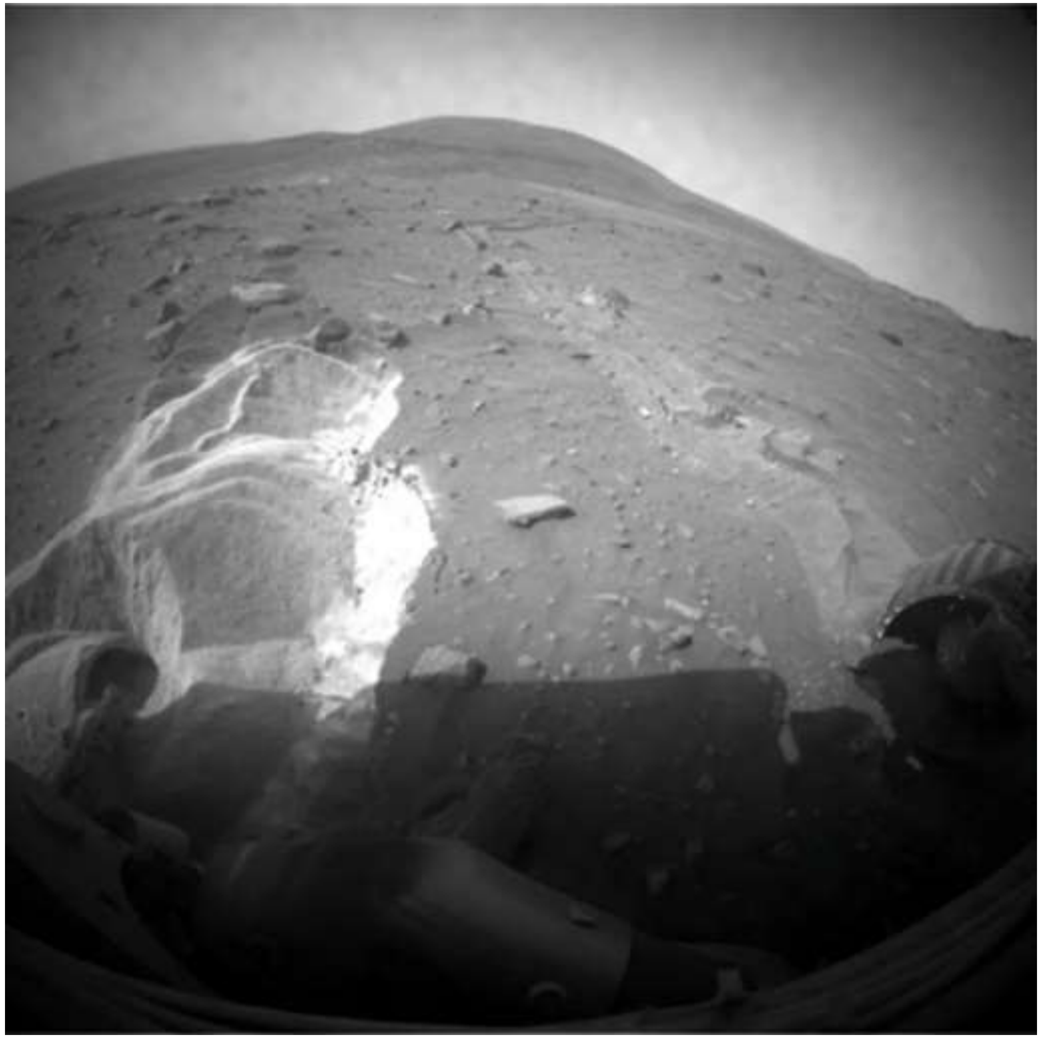
For full details, see https://hof.povray.org/

Simulate what a camera sees under water…

robot plants heat-resistant corals to save endangered reefs
[https://www.popsci.com/heat-resistant-corals-robot]→

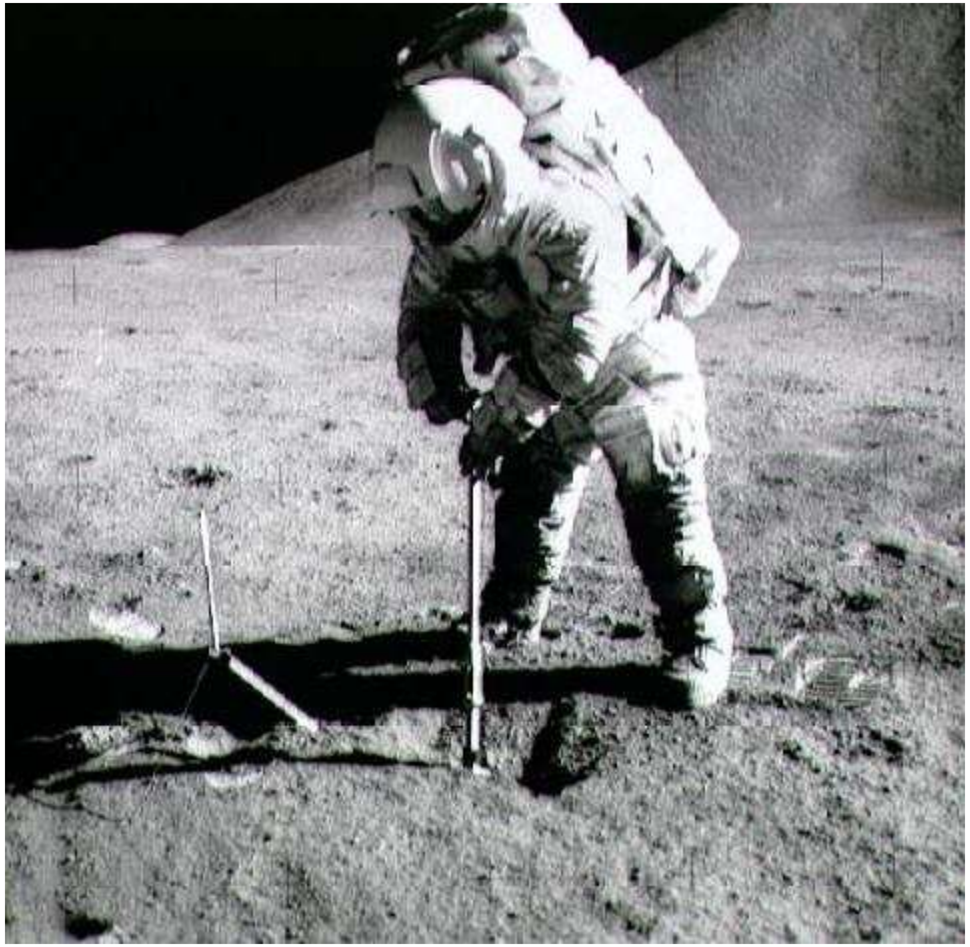
Movie, Spirit trapped



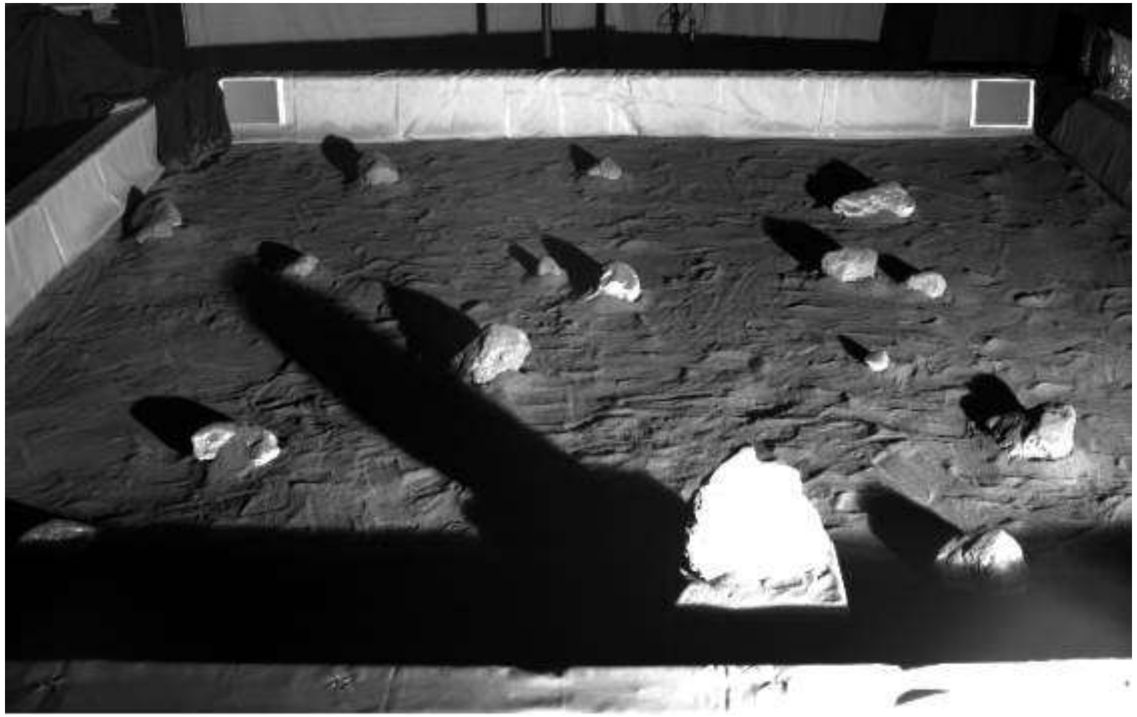


Credit: NASA

No clear idea what a camera should see on the moon…

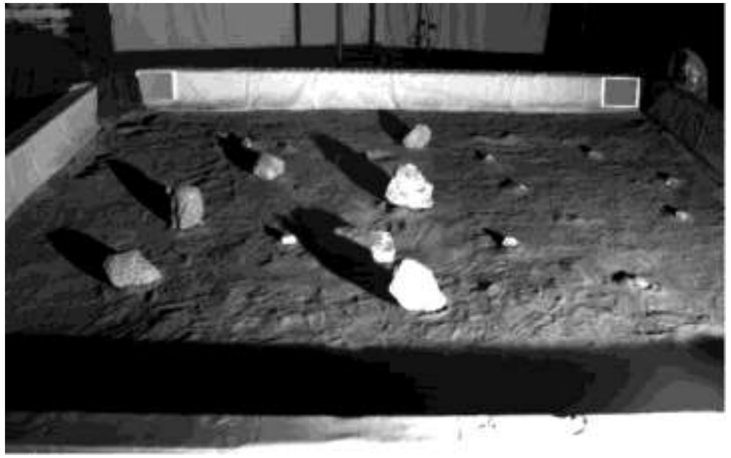


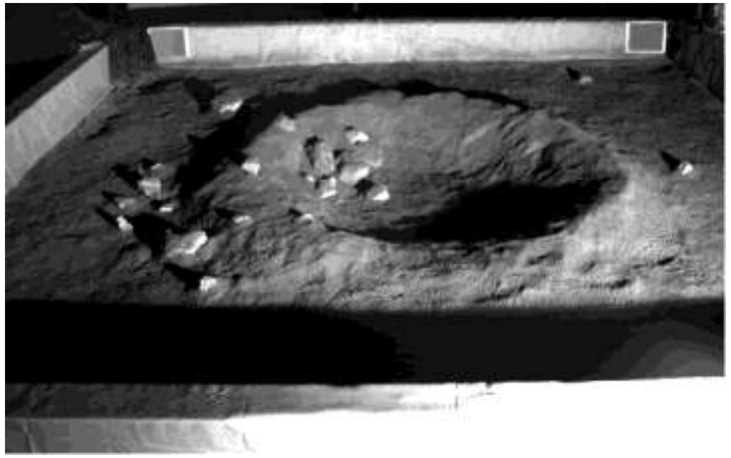
https://nssdc.gsfc.nasa.gov/image/spacecraft/alsep_soil_mech.jpg



NASA's Polar Stereo Dataset

NASA's Polar Stereo Dataset





A frame of what the camera "sees"

NASA's Polar Stereo Dataset

Another frame of what the camera "sees"

NASA's [Polar Stereo Dataset]

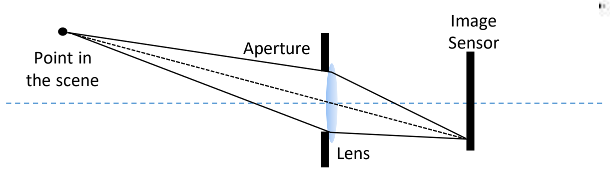
Quick comments, on camera simulation

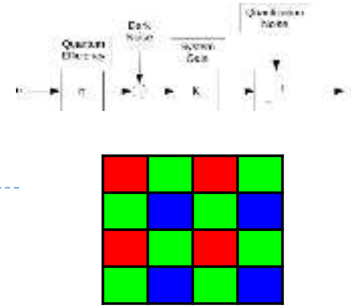
- Use computer graphics to generate in simulation what the camera sees in reality
- We care about what a camera sees, not what a human expects to see
- Camera simulator good if Computer Vision can't discern between real of simulated data

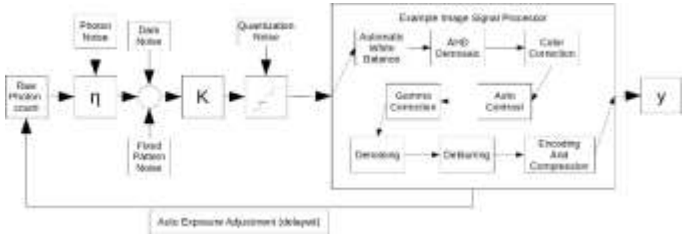
Camera modeling and simulation

- Understanding how light forms an image
- Many steps have unknown features and parameters

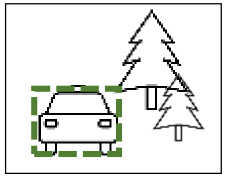
Scene → Optical System → Image Sensor → Image Signal Processor → Perception







[Choi, 2014] →

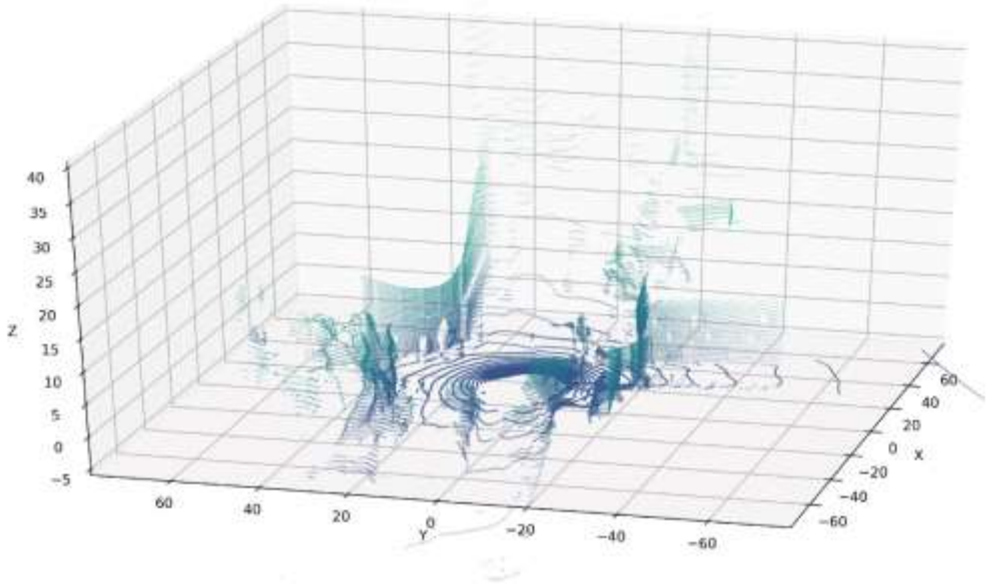


Scene	Optical System	Image Sensor	Image Signal Processor	Perception
Weather Materials Lighting	Lens Distortion Lens Flare Vignetting	Measurement Noise	Demosaicing Color Balance Compression	Application context

Chrono::Sensor – supported sensors

- Dynamics-based sensors
  - IMU (gyroscope, accelerometer, magnetometer)
  - GPS
- Light-based sensors leveraging ray tracing
  - Camera
  - Lidar
  - Radar (early prototype)

Simulated LiDAR point cloud using Chrono::Sensor



Simulation-Based Engineering Lab

SBEL

So again, what are we interested in simulating?

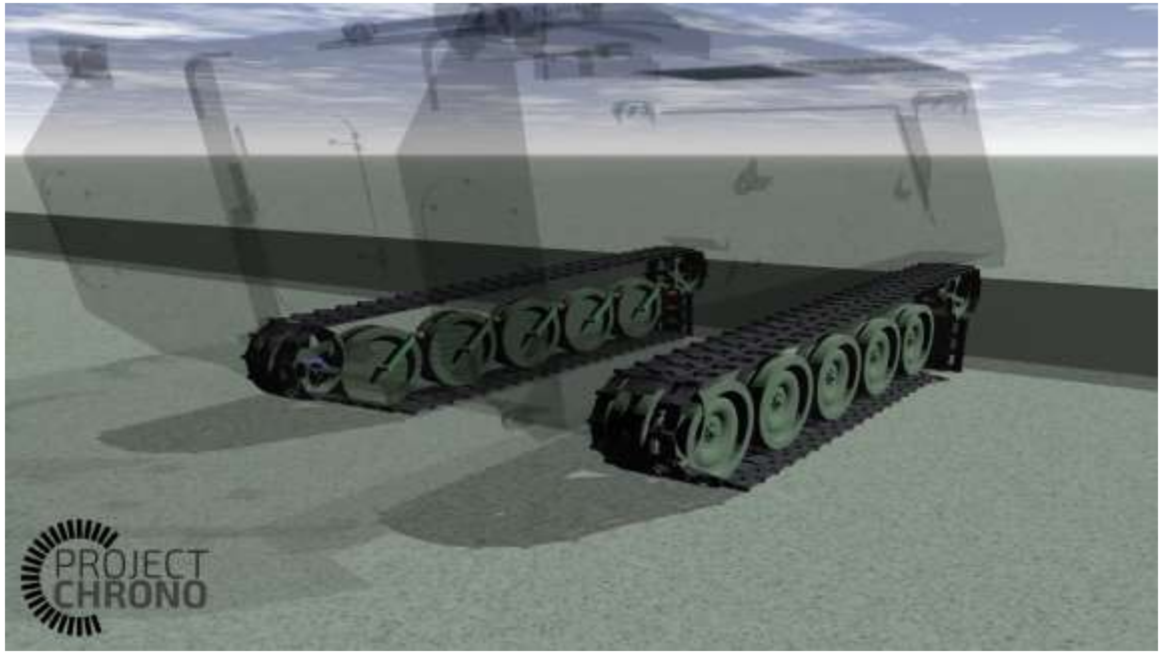
- Simulate the process of sensing
- Simulate the robot/rover/autonomous vehicle
- Simulate the world in which the robot/rover/autonomous vehicle operates

Multibody Dynamics w/ Friction and Contact: The Math

$$
\begin{aligned}
\dot{\mathbf{q}} &= \mathbf{L}(\mathbf{q})\mathbf{v} \\
\mathbf{M}(\mathbf{q})\dot{\mathbf{v}} &= \mathbf{f}\left(t, \mathbf{q}, \mathbf{v}\right) - \mathbf{g}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}, t)\lambda \\
\mathbf{0} &= \mathbf{g}(\mathbf{q}, t)
\end{aligned}
$$

Chrono::Vehicle

Chrono::Vehicle – Tracked system example



DEM example applications





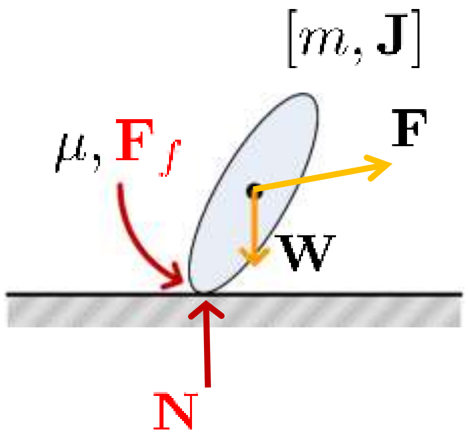

So again, what are we interested in simulating?

- Simulate the process of sensing
- Simulate the robot/rover/autonomous vehicle
- Simulate the world in which the robot/rover/autonomous vehicle operates

Mass × Acceleration = Force

Mass × Acceleration = Force

- Coulomb friction, w/ friction coefficient μ

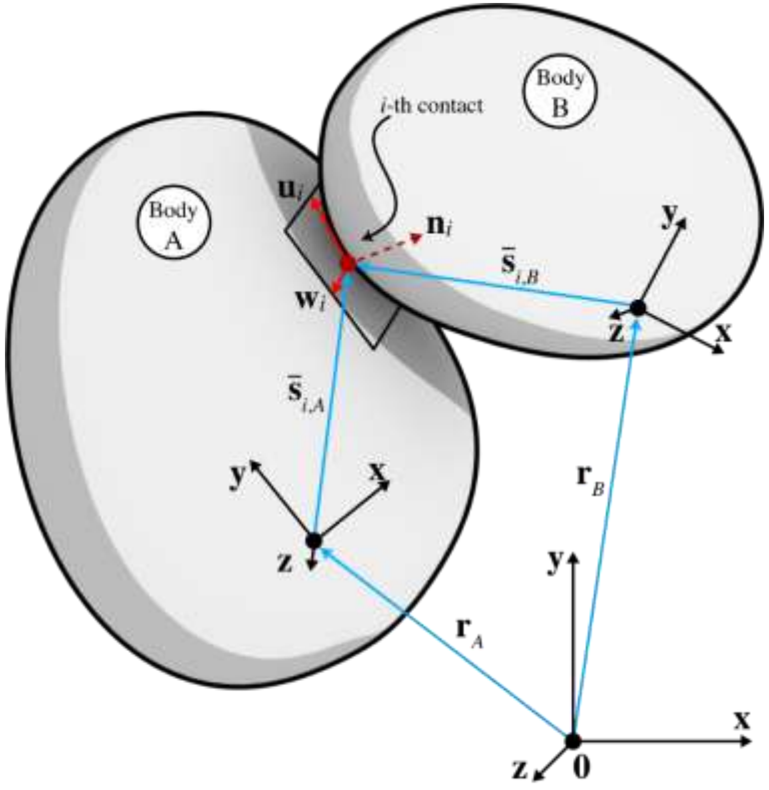


$$m\dot{\mathbf{v}} = \mathbf{W} + \mathbf{F} + \mathbf{N} + \mathbf{F}_f$$

$$||\mathbf{F}_f|| \leq \mu ||\mathbf{N}||$$

Terradynamics: 1,000 million bodies

Two main approaches: penalty & complementarity



Problem

Computational many-body dynamics
Handling frictional contact

Modelling approach

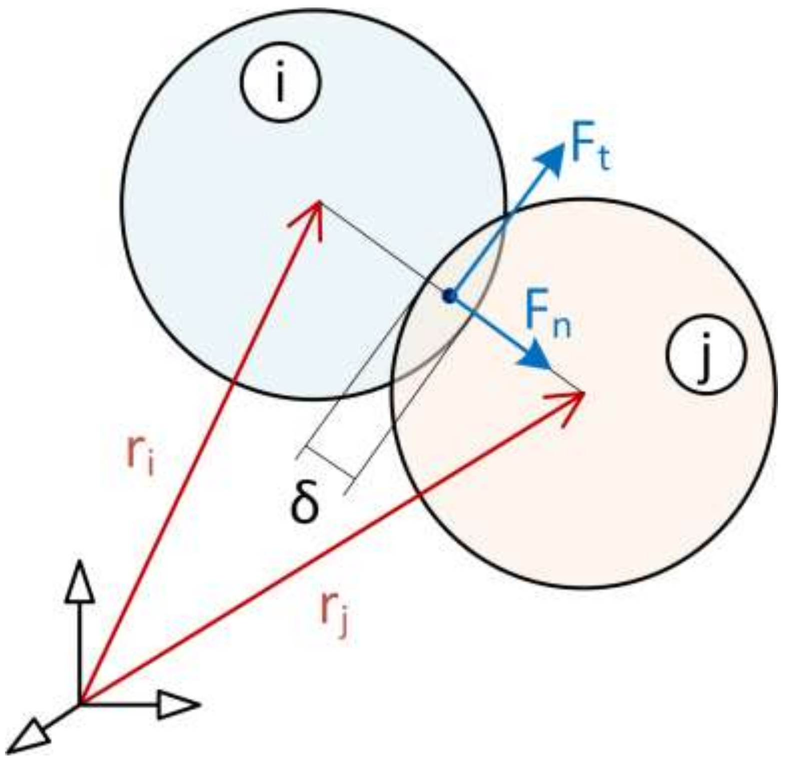
- Penalty-based approach
- Complementarity approach

Numerical techniques

- Collision detection (Penalty-based approach; Complementarity approach)
- Optimization techniques (Complementarity approach)

Most common frictional contact model: The PENALTY model

Kinematic/Geometry Aspects



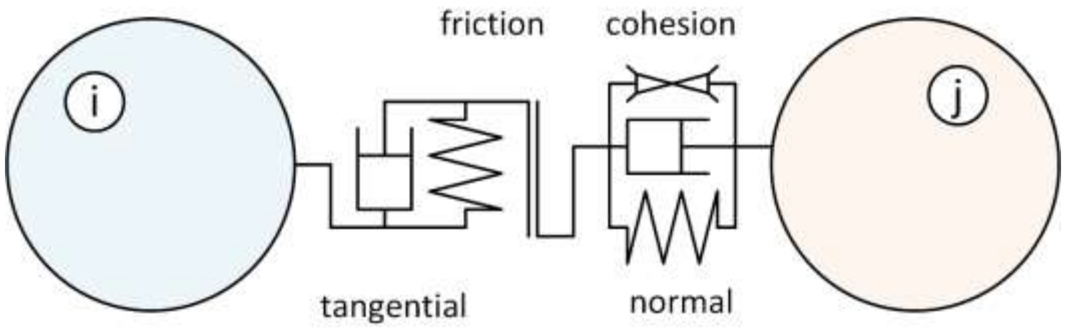
- Cundall & Strack: *A discrete numerical model for granular assemblies*, Geotechnique 29.1 (1979): 47-65.
- Almost 18,000+ citations
- “This overlapping behavior takes the place of the deformation of the individual particles.”

PENALTY model: The contact (normal) component

- Kelvin-Voigt: $F_N = K\delta_n + D\dot{\delta}_n$
  - linear spring, linear damper
  - contact force at zero deformation is not continuous
- Hunt-Crossley: $F_N = K\delta_n^\alpha + \chi\delta_n^\alpha\dot{\delta}_n$ with $\chi = \frac{3K(1-c_r)}{2\dot{\delta}_n^-}$
  - express damping as function of deformation (disspation as heat)
  - typically, $\alpha = 3/2$
  - force is zero at start and end of contact
- Lankarani-Nikravesh: $\chi = \frac{3K(1-c_r^2)}{4\dot{\delta}_n^-}$
  - valid only for c_r close to unity (dissipated energy small compared to maximum absorbed elastic energy)
- Machado-Flores: $\chi = \frac{8K(1-c_r)}{5c_r\dot{\delta}_n^-}$
  - derived from energy balance and conservation of linear momentum
  - correctly captures $\chi \to \infty$ as $c_r \to 0$

1. W. Goldsmith, Impact, The Theory and Physical Behaviour of Colliding Solids, Edward Arnold Ltd, London, 1960
2. K.H. Hunt and F.R.E. Crossley, Coefficient of Restitution Interpreted as Damping in Vibroimpact, J. Appl. Mech., 42, 1975
3. H.M. Lankarani and P.E. Nikravesh, A Contact force model with hysteresis damping for impact analysis of multibody systems, J. Mech. Design, 112 (1990)
4. M. Machado, P. Moreira, P. Flores, H.M. Lankarani, Compliant contact force models in multibody dynamics: evolution of the Hertz contact theory, Mech. Mach. Theory, 53 (2012)

PENALTY model: the frictional component



- The model:
  - "The resultant forces on any disc are determined exclusively by its interaction with the discs with which it is in contact."
  - One normal/contact force
    - "The force-displacement law is used to find contact forces from displacement"
  - One tangential/friction force

Cundall & Strack: *A discrete numerical model for granular assemblies*, Geotechnique 29.1 (1979): 47-65.

PENALTY model: the frictional component (Cnt'd)

Common accepted model: Cundall and Strack (1979) - virtual tangential spring

- Include Coulomb law of attrition: $|F_T| \leq \mu |F_N|$
- Refinements: viscosity, non-linearity, hysteresis

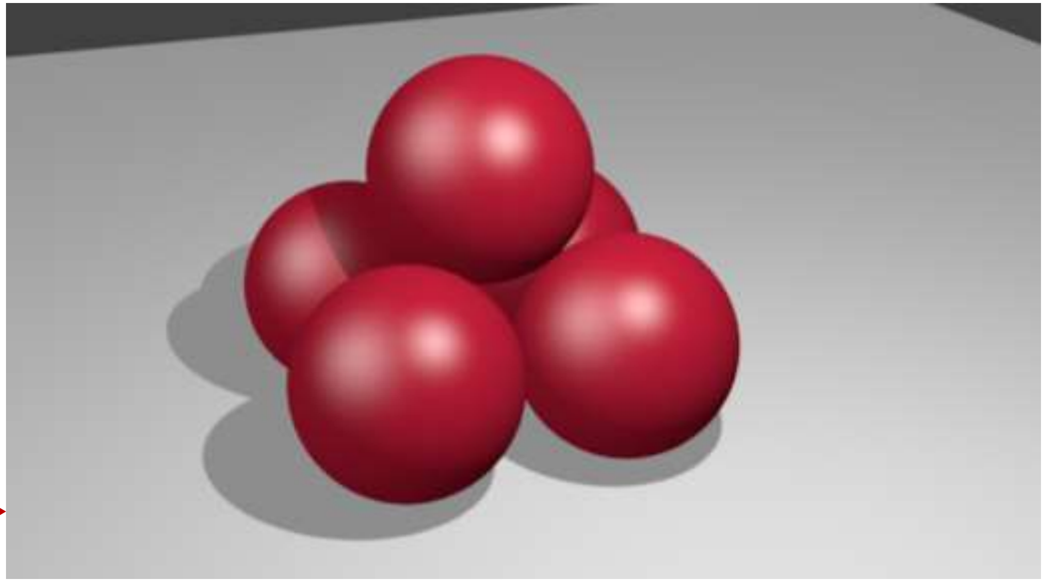
Viscoelastic model (general form): $F_T = -K_T(\delta_n)\delta_t - D_T(\delta_n)v_t$

- linear (K_T, D_T constant) or nonlinear ($K_T, D_T \propto \delta_n^{1/2}$)
- tangential displacement: $\delta_t = \int_{t_c} v_t(\tau) d\tau$
  approximation: $\delta_t \approx h v_t$

1. P.A. Cundall and O.D.L. Strack, A discrete numerical model for granular assemblies, Geotechnique, 29(1), 1979
2. H. Kruggel-Emden, S. Wirtz, V. Scherer, A study on tangential force laws applicable to the discrete element method for materials with viscoelastic or plastic behavior, Chem. Eng. Sci., 63, 2008
3. A. Di Renzo and F.P. Di Maio, An improved integral non-linear model for the contact of particles in distinct element simulations, Chem. Eng. Sci., 60, 2005

Terradynamics: a multi-scale problem

Vehicle ↔1 [m] versus Sand grain ↔1E-3 – 1E-4 [m]

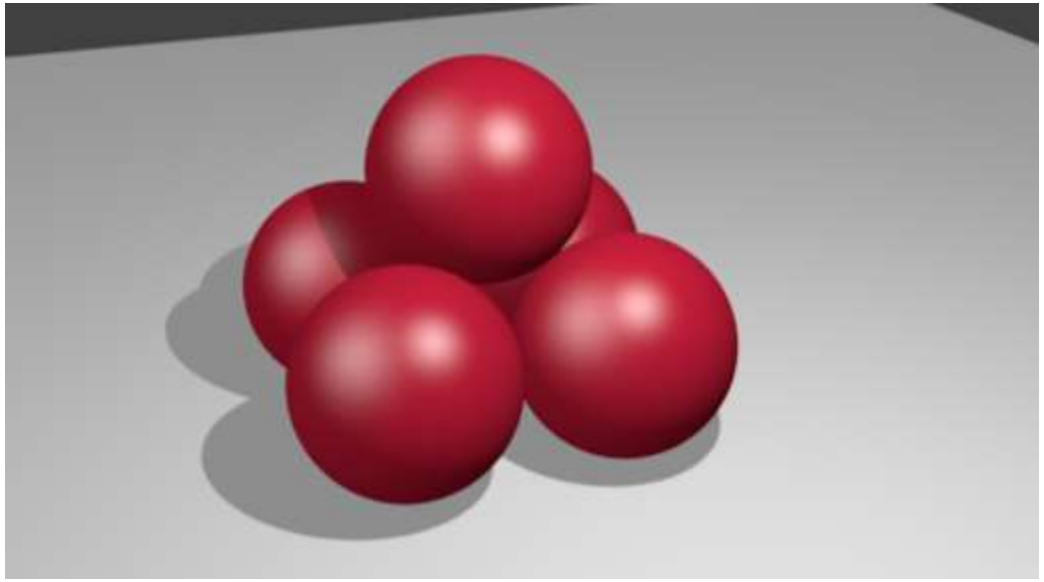


Macroscale emergent behavior dictated by microscale dynamics

What's "microscale"? It gets even more interesting…

- Sphere sitting on top of another sphere
  - Penalty method, all units SI
    - K = 10^{10}
    - $\rho = 2600$
    - Gravitational acceleration (Europa): 1.3
    - Radius: 10^{-4}
    - Deformation order $\approx 10^{-18}$
  - Phobos and Deimos gravity: 1700X lower than Earth

Sand grains ↔1E-3 – 1E-4 [m]



A look around, problem size and element morphology

Year	# Bodies	Geometry	?D	# References	Link
1998	3,960	Polydisperse spheres	3D	618	Source
2001	8,000	Monodisperse spheres	3D	773	Source
2003	1,000	Monodisperse disks	2D	534	Source
2005	5,000	Range of radii disks	2D	711	Source
2006	440,000	Spheres, of 3 radii	3D	281	Source
2007	~100s	Monodisperse Spheres	3D	58	Source
2014	260,000	Monodisperse spheres	3D	11	Source
2014	18,000	Polydisperse spheres	3D	23	Source
2015	~1,000s	Polydisperse spheres	3D	59	Source
2015	?	Polydisperse ellipsoids	3D	1	Source
2015	21,812	Polydisperse spheres	3D	34	Source
2016	20,000	Crushable unions of spheres of different radii	3D	27	Source
2016	90,905	Polydisperse spheres	3D	?	Source
2016	5761 particles	Unions of spheres (24771 spheres)	3D	66	Source
2017	16,000	Polydisperse spheres, cubes	3D	?	Source
2017	33,600	Polydisperse spheres	3D	15	Source
2017	300,000	Monodisperse spheres	3D	2	Source
2018	46,280	Spheres	2D	1	Source

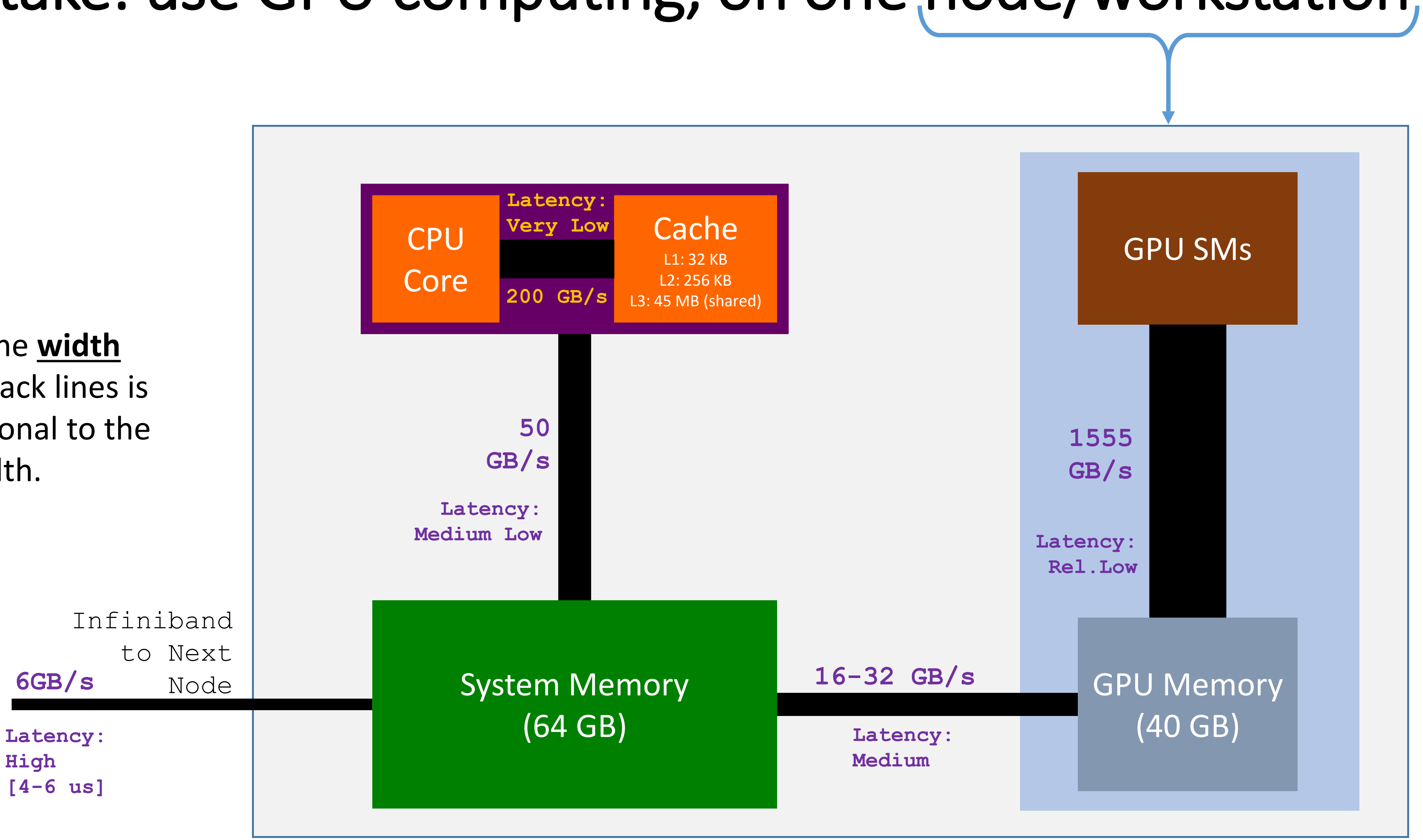
State of the Art, 3D Friction and Contact

- Japan’s K-Supercomputer (World’s fastest in 2012)
- Frictional contact problem: 2.4 billion elements (about 18 billion DOFs)
- 131,072 cores (MPI processes)
- Run in 2017-2018

M. Furuichi, D. Nishiura, O. Kuwano, A. Bauville, T. Hori, and H. Sakaguchi, “Arcuate stress state in accretionary prisms from real-scale numerical sandbox experiments,” Scientific Reports - www.nature.com/scientificreports, vol. 8, 12 2018.

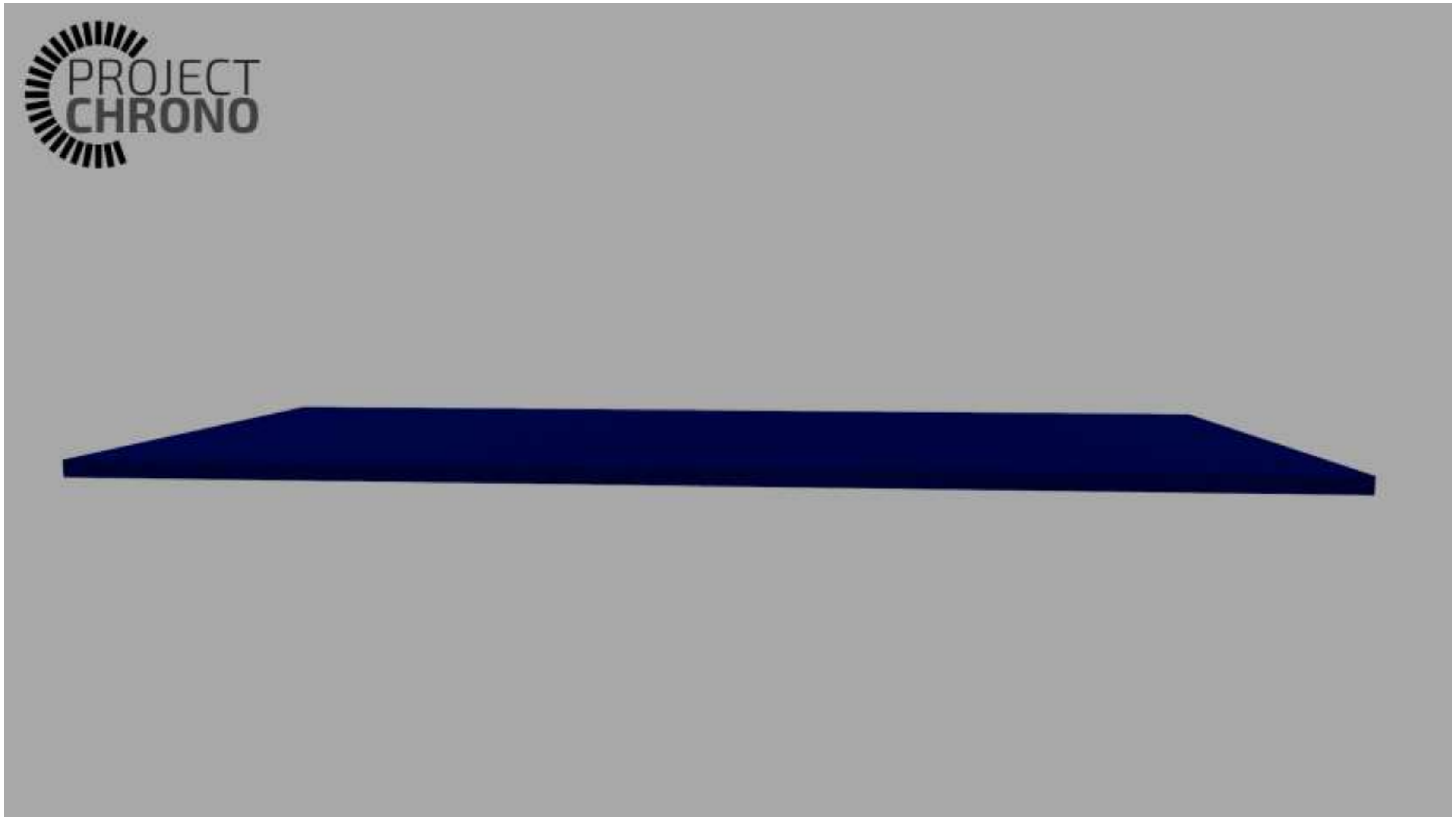
Our take: use GPU computing, on one node/workstation

NOTE: The **width** of the black lines is proportional to the bandwidth.

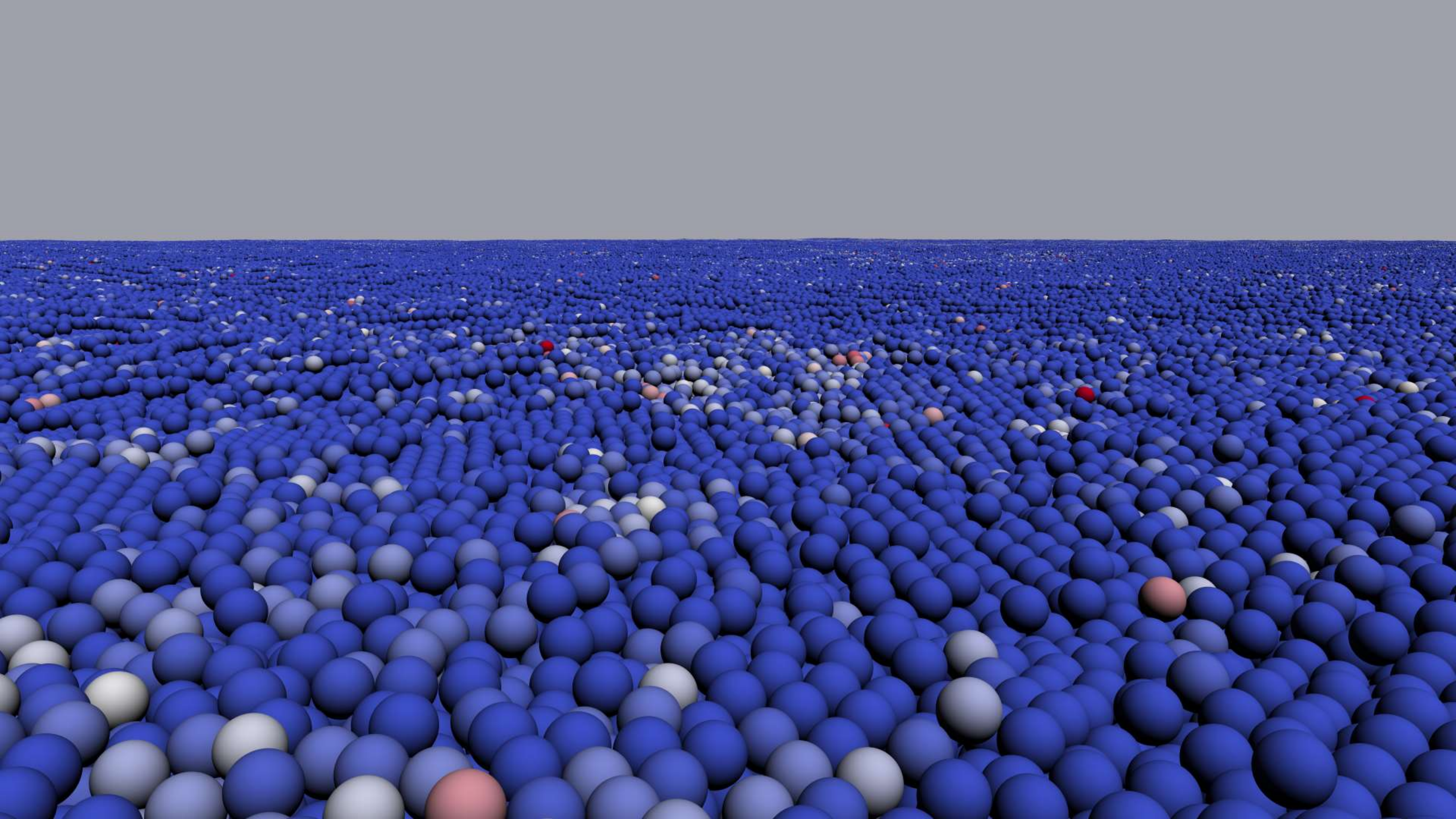


201,898,800 Body Problem

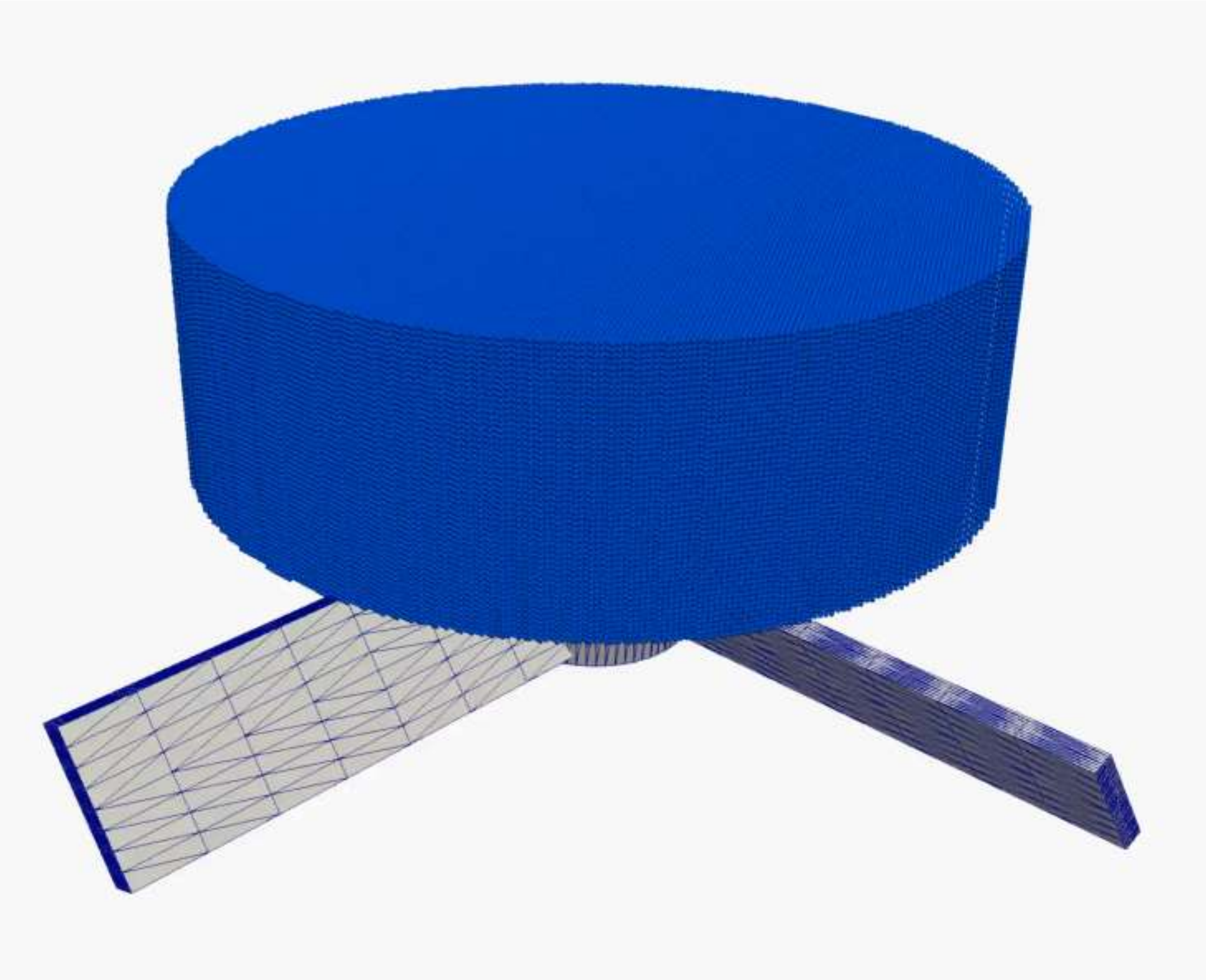
[2.8 seconds of settling: 24 hours of compute time]



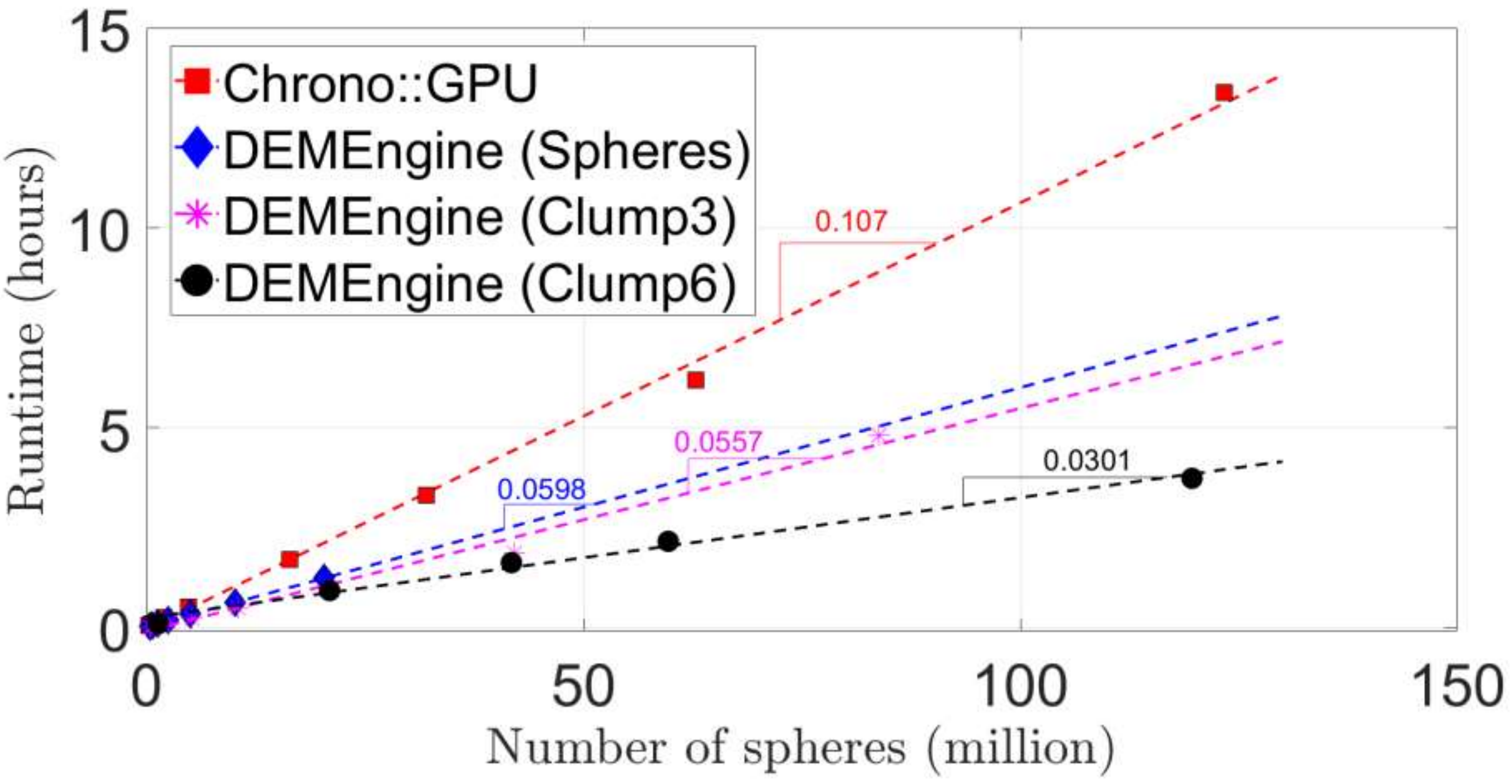
- Patch of granular material
  - 35 m long
  - 15 m wide
  - 0.5 m deep
  - Element radius: 0.5 cm
  - Cohesion force but no friction force



Scaling Analysis
Test problem: Granular material mixing…



Scaling analysis: 1 second of mixer simulation on A100s



Clump3

Clump6

Being mindful of storing data on the GPU (done parsimoniously)

Data Type	Variable	Memory Type
`uint64_t`	Voxel index	Global
`uint16_t`	Sub-voxel index	Global
`int32_t` or `float`	Kinematic quantities, friction history etc.	Global
`double`	Penetration	Register
`double`	Integration (time marching)	Register
`float`	Particle shape information	Shared

- Defining our data representation makes more effective use of memory bandwidth usage
- We do not compromise the physics (such as penetration calculation)

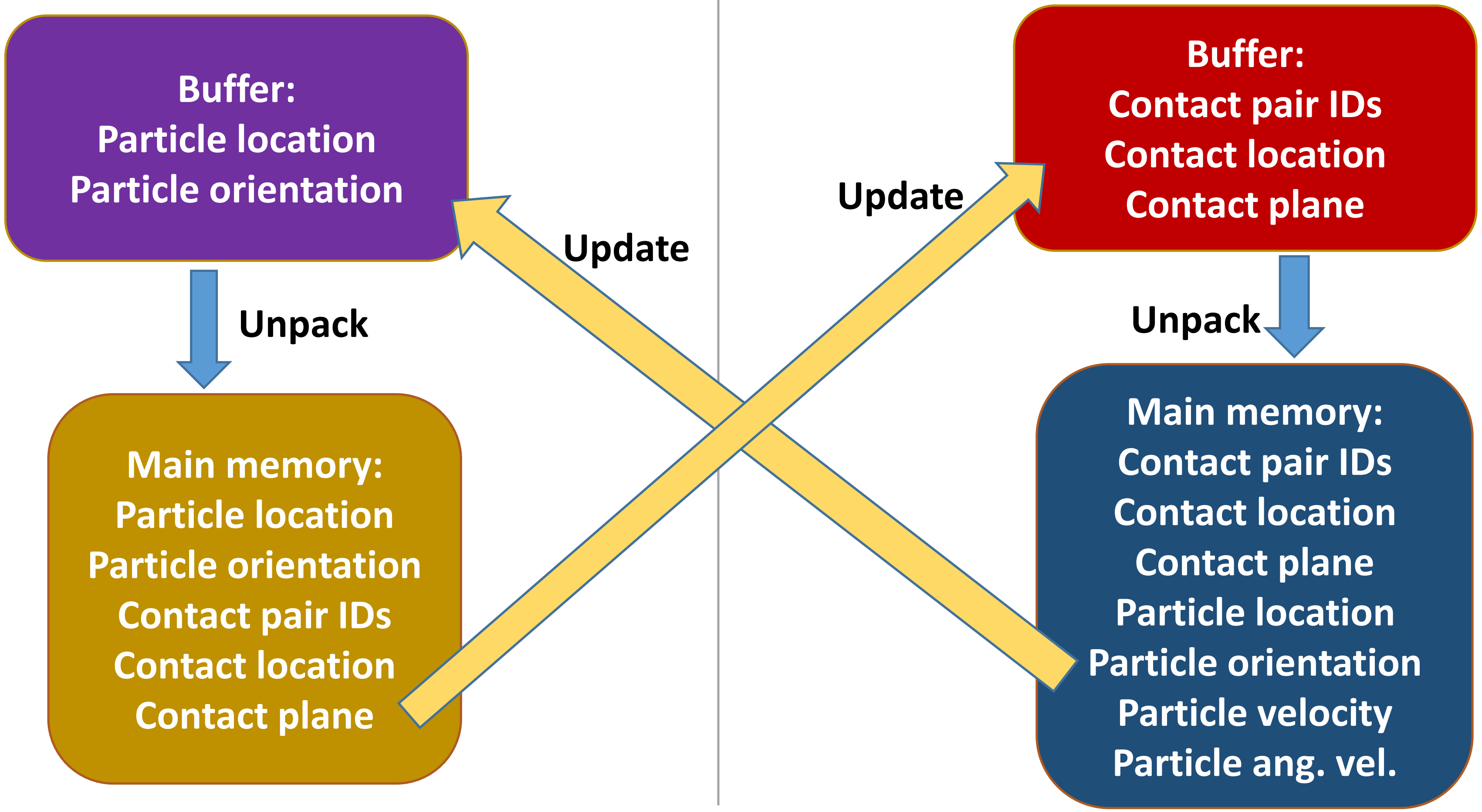
Dual-GPU solution

Kinematic Thread (GPU1)

- Tasked with contact detection
- And misc. issues that need not to be in sync with physics

Dynamic Thread (GPU2)

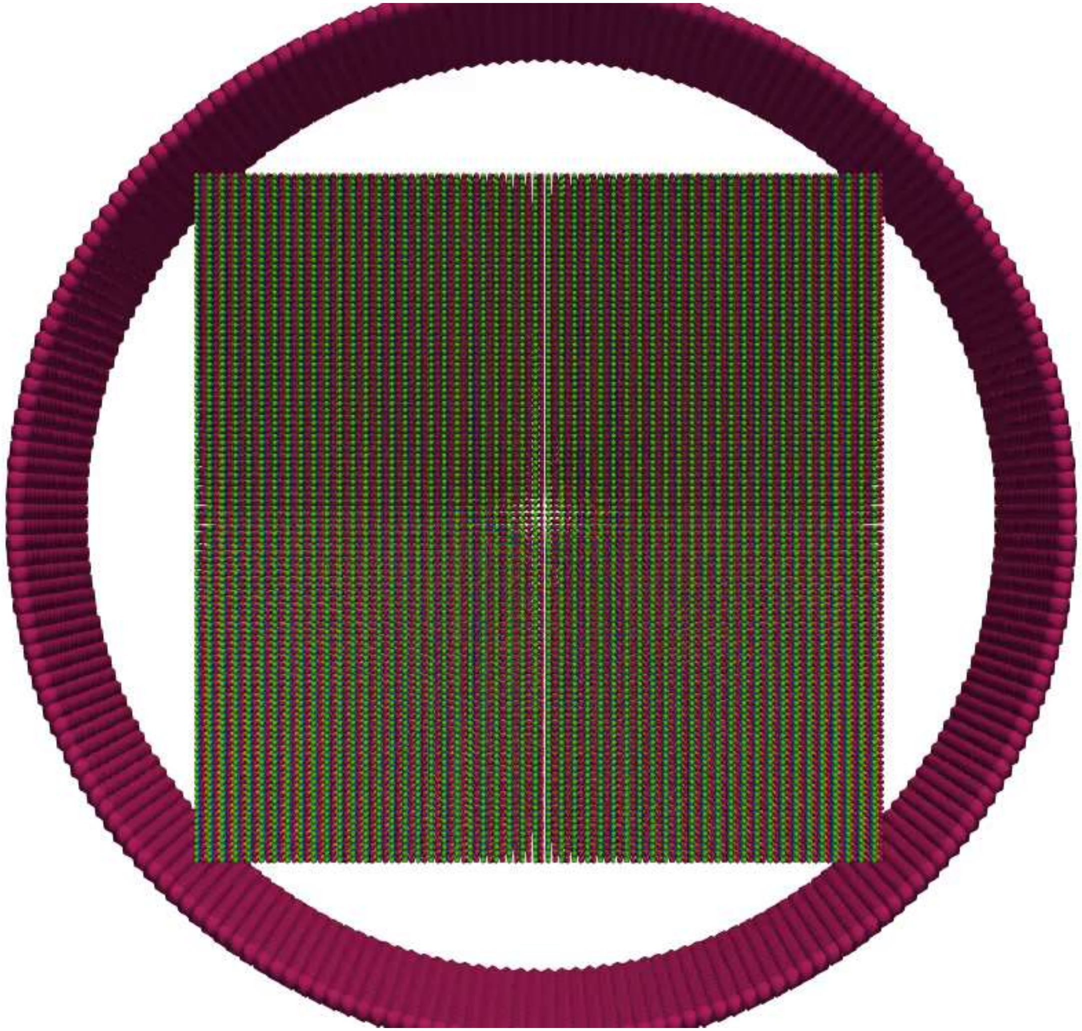
- Tasked with advancing the physics
  - Contact force calculation
  - Integration



Larger problem size examples [Updated]

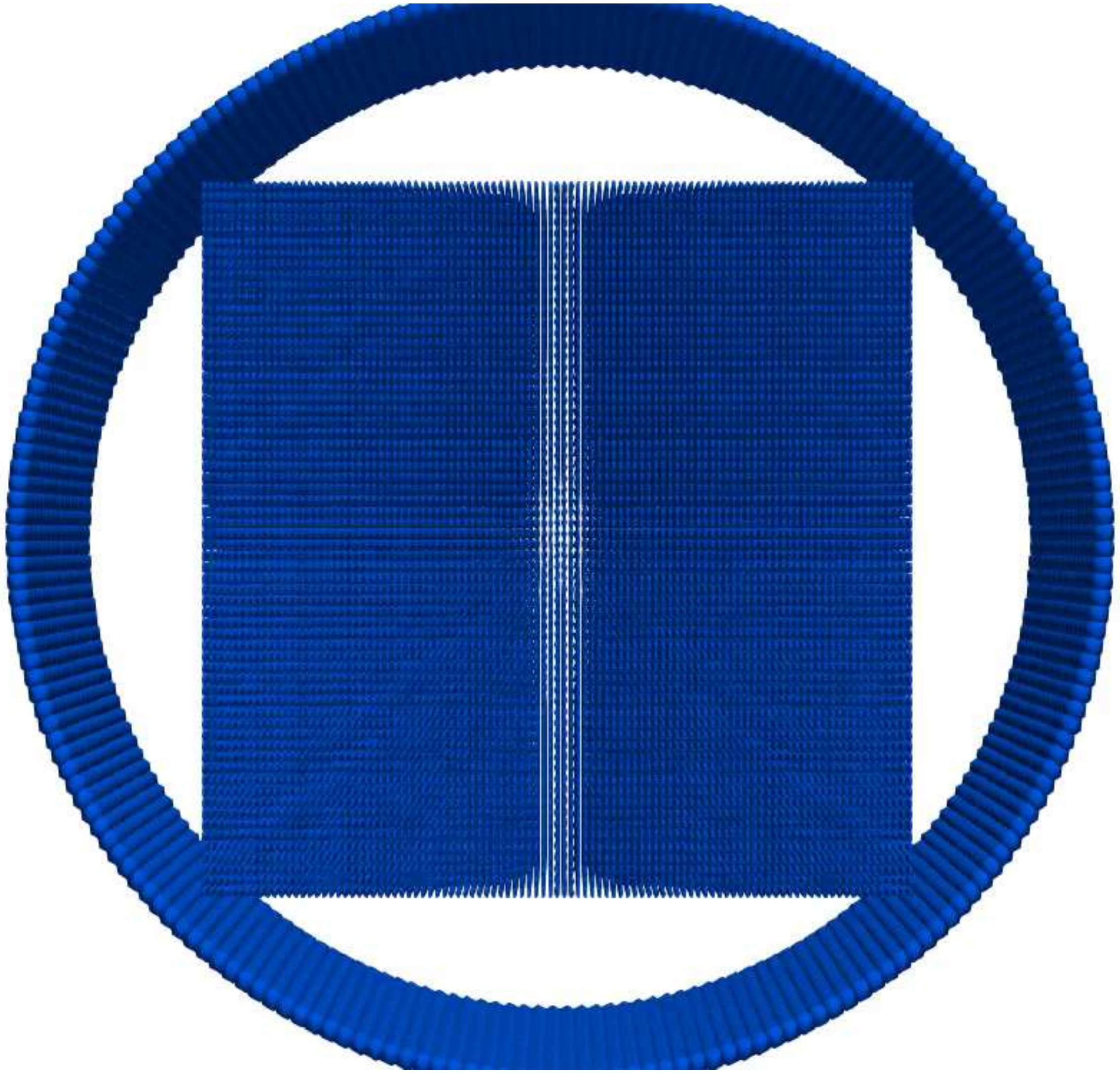
Year + Link	Problem size [millions]	2D/3D + geometry	Simulated time	Simulation time	Hardware used	Citations	Comments
2002	0.2	2D, mono	1 [s]	200 [s]	Swiss-T1, 64-node supercomputer	578	$1 million, in 2001 dollars
2005	1	3D spheres, 8 mm radius	1.5 [s]	1 week	32 processors	234	
2012	0.4	3D, spheres, 1 mm radius			36 processors, MPI	586	
2013	0.15	3D, bidsiperse spheres, 2/4 mm	10 [s]	375 hours	16 processors	18	
2013	0.256	3D, composete macrospheres			GPU	37	
2014	0.08	3D, spheres, 4 different radii	120 [s]	35 days	32 cores, MPI	66	
2014	0.13	3D, mono, 2.5 mm	?	?	?	89	
2015	0.392	3D, 3-macrospheres (4 mm/sphere)	?	?	?	35	
2016	1	3D, mono, 2.5 mm	?	?	1 GPU	42	
2017	0.9	3D, mono, 12 mm	7.64 [s]	5.5 days	1 CPU, multi-core	78	**Chrono**
2017	0.09	3D, poly	5 [s]	2 days	Intel Core i3-2100 1.58 GHz	13	
2018	1	3D, spheres, three radii	?	?	up to 20 GPUs	8	
2018	2.8	3D, spheres, polydisperse	?	(**32 days**; stopped early)	GPU (GTX1080)	5	up to 4 million elems
2018	0.25	3D, quad-disperse, spheres	?	?	4-core on one CPU	10	commercial code
2018	**2400**	**3D, poly, spheres**	**?**	**?**	**131,072 cores**	**2**	**Japan's K-supercomputer**
2022	**120 [20]**	**Six-sphere clumps**	**1 [s]**	**4 hours**	**2 GPUs**	**WIP**	**scales linearly**

Example — Centrifuging 1



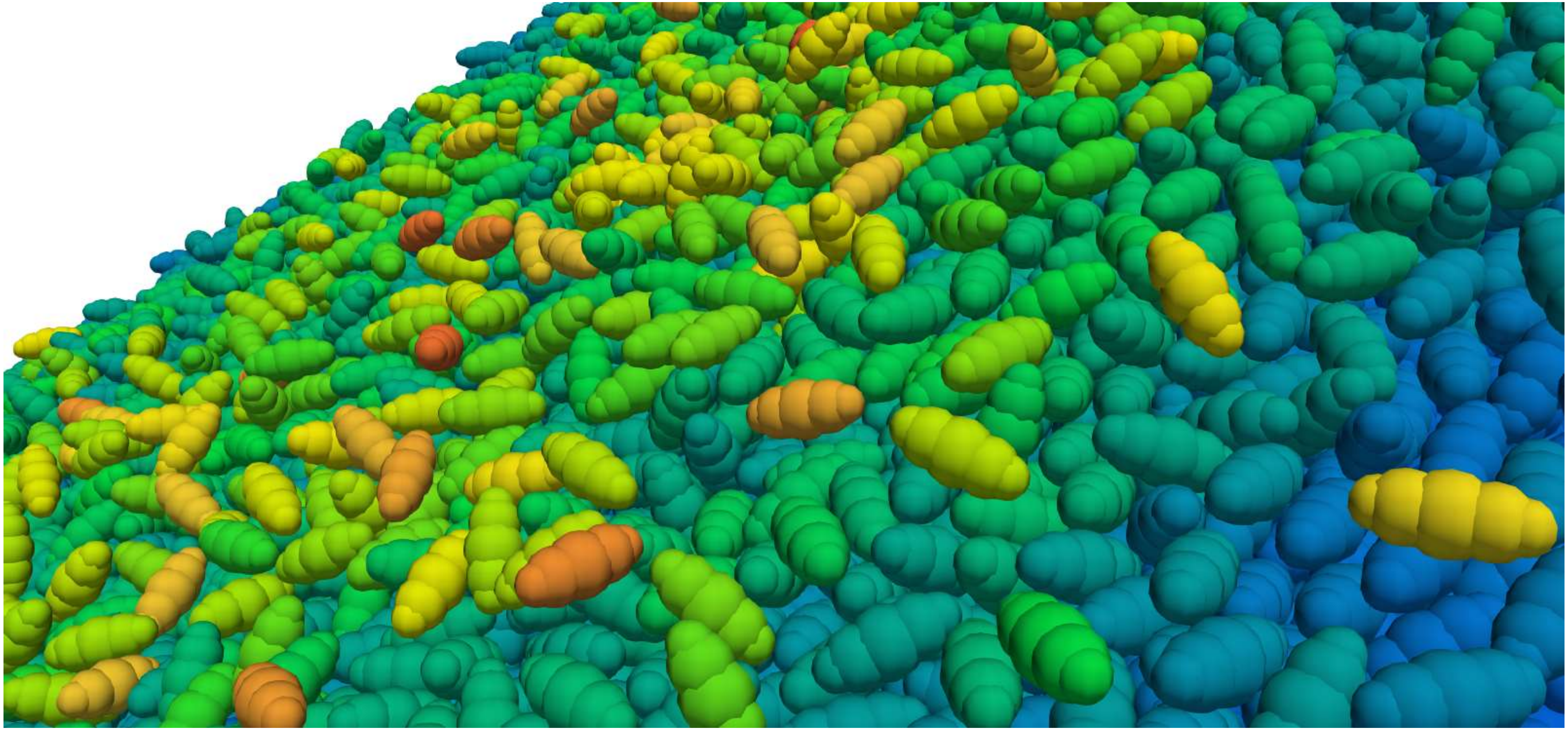
All particles are spherical, same radius. Cold color means lighter particles and warm color means heavier ones. The drum rotates at around 2π rad/s. Gravity parallel to axis of rotation.

Example — Drum (ellipsoidal particles)



Simulation scale is about 200k clumps (1M spheres). The drum rotates at 0.1 rad/s. The final incline is about 33°.

Drum (ellipsoidal particles)



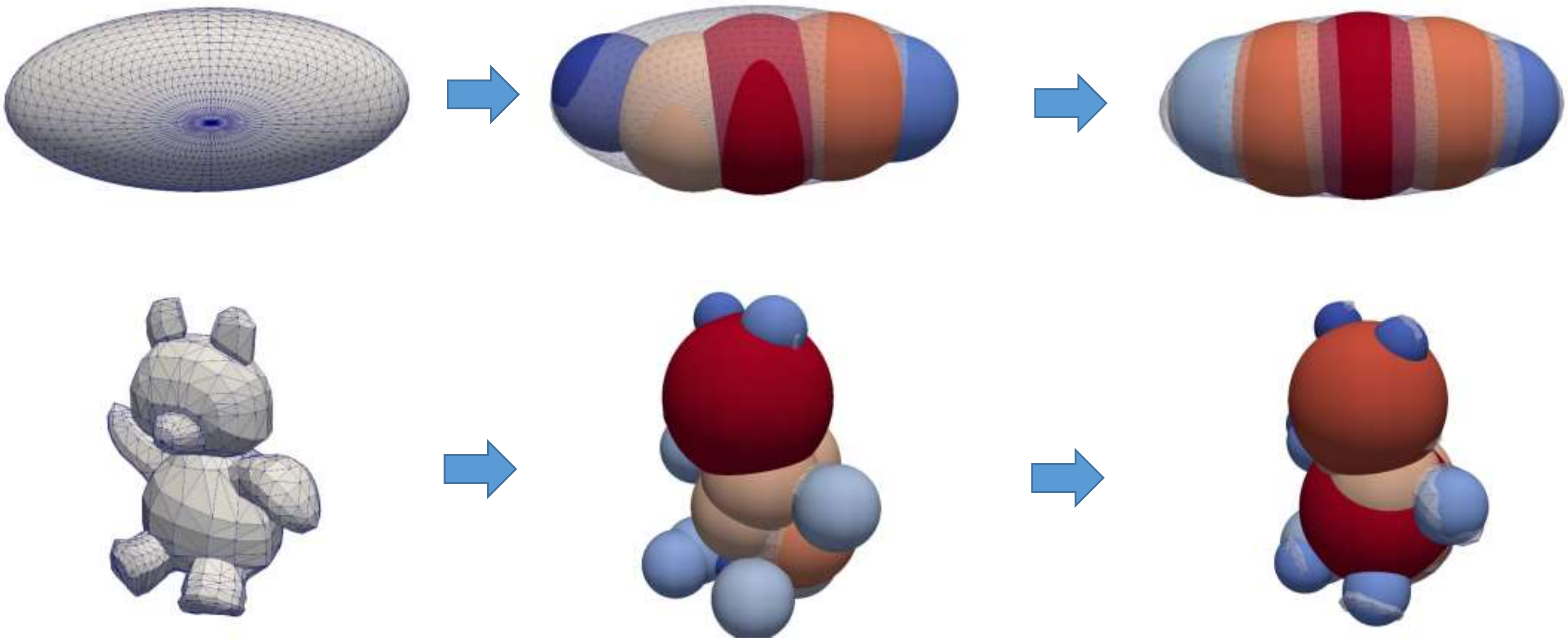
Close-up of the ellipsoidal particles

Clump Shape Generator

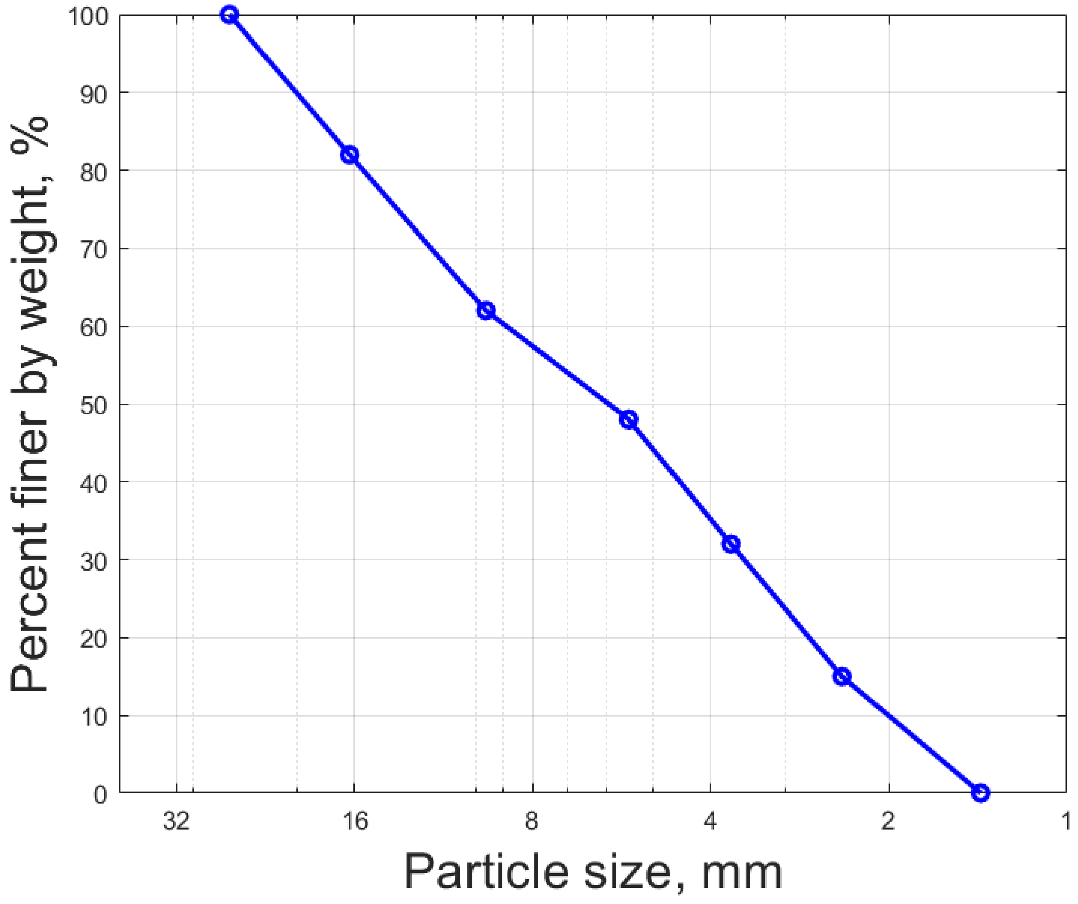
Original meshed object

Convex-decomp. then fit spheres

Optimize spheres to fit mesh



GRC-1 terrain, particle shape & size (scaled up by ≈10×)



Particle size distribution used in this simulation

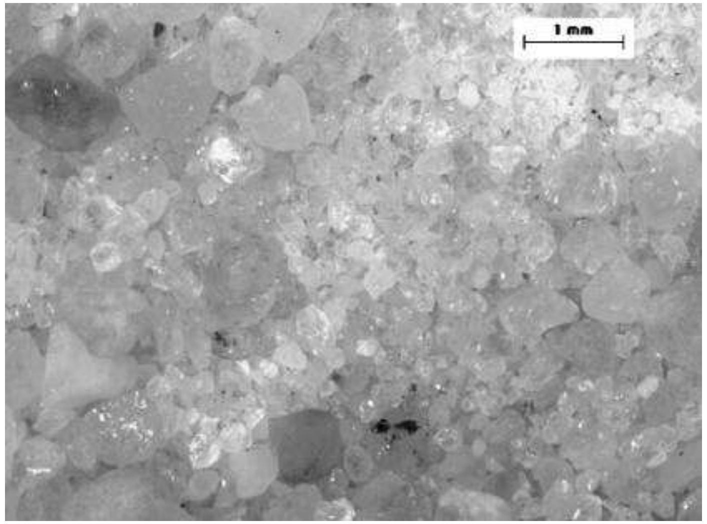
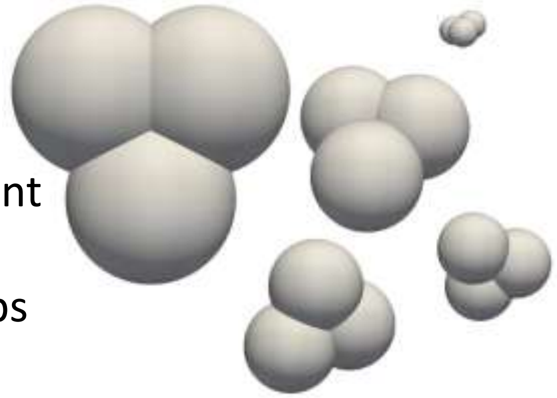


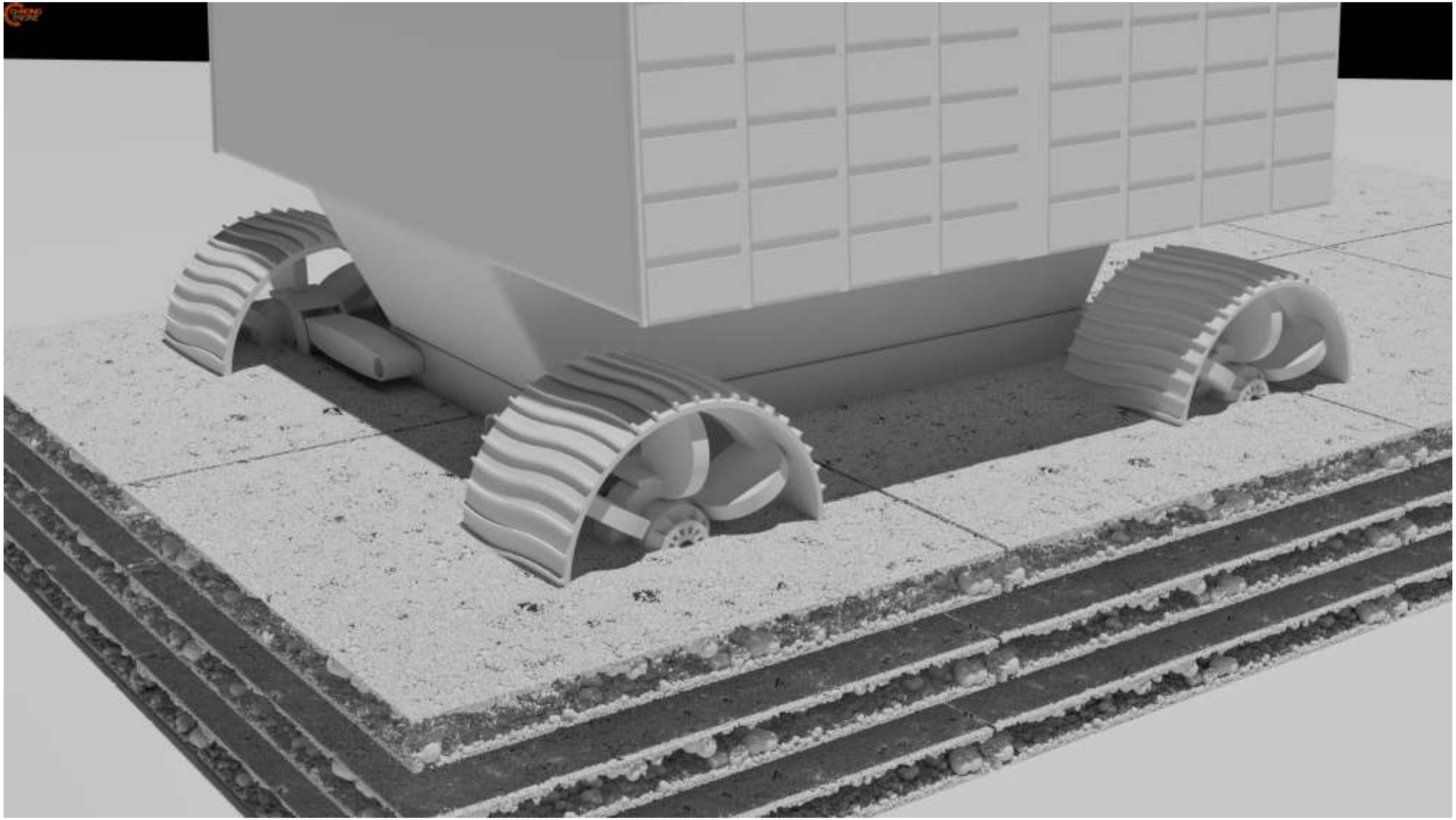
Photo depicting GRC-1 grain geometry and particle size [1]

[1] H.A. Oravec, X. Zeng, V.M. Asnani, 2010. “Design and characterization of GRC-1: A soil for lunar terramechanics testing in Earth-ambient conditions”. Journal of Terramechanics, 47(6), pp. 361–377.



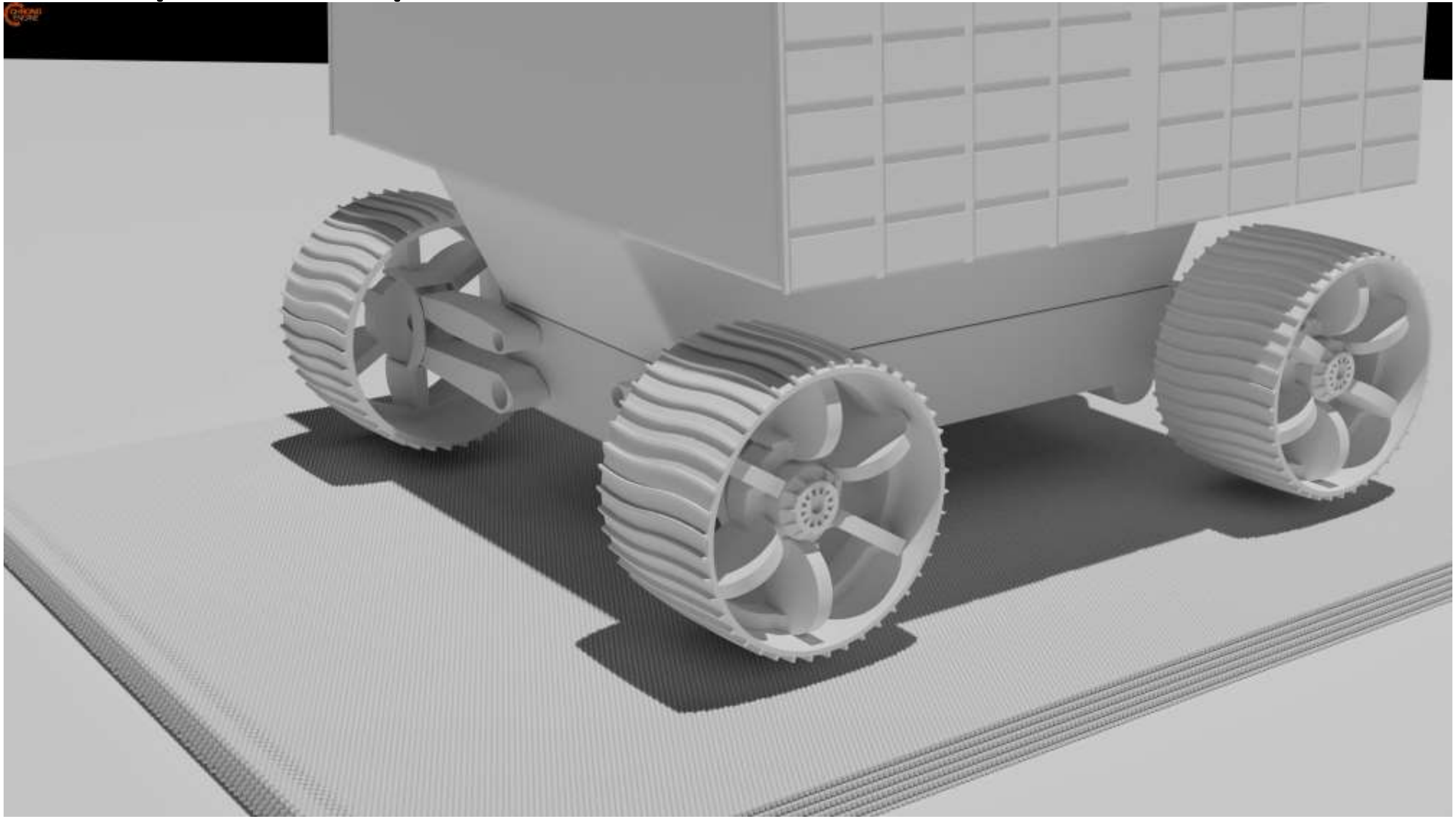
The particle shapes present in the DEM simulations, which are 3-sphere clumps of different sizes

VIPER on GRC-1 terrain (10X enlarged)



Earth gravity. Particles have a GRC-1-like distributions of sizes.

VIPER on sphere-represented terrain



Same rover. The sinkage, mobility different when using monodisperse spheres

Continuum Representation Model (CRM) for Terramechanics

- Fully resolved terramechanics using DEM is often times too computationally demanding
- We started using embraced a continuum representation of the terrain
- The PDEs associated with the continuum representation of the terrain solved via SPH

Continuum Representation Model for Terramechanics

- Equations: mass conservation & momentum balance & stress rate of change

- Spatially discretized with Smoothed Particle Hydrodynamics (SPH)

- Using constitutive model proposed in 2015 (Dunatunga & Kamrin)

Continuum modeling of granular material: governing equations

- Mass and Momentum balance equations

$$
\begin{cases}
\dfrac{d\rho}{dt} = -\rho\nabla\cdot\mathbf{u} \\
\dfrac{d\mathbf{u}}{dt} = \dfrac{\nabla\boldsymbol{\sigma}}{\rho} + \mathbf{f}_b
\end{cases}
$$

Cauchy stress tensor

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau}$$

deviatoric component of stress tensor

isotropic pressure

$$p = -\frac{1}{3}\mathrm{tr}(\boldsymbol{\sigma}) = -\frac{1}{3}(\sigma_{xx} + \sigma_{yy} + \sigma_{zz})$$

The constitutive model; draws on Zaremba-Jaumann equation

$$\frac{d\boldsymbol{\sigma}}{dt} = \dot{\boldsymbol{\phi}} \cdot \boldsymbol{\sigma} - \boldsymbol{\sigma} \cdot \dot{\boldsymbol{\phi}} + \overset{\triangle}{\boldsymbol{\sigma}} ,$$

$$\dot{\boldsymbol{\phi}} = \frac{1}{2}(\nabla \mathbf{u} - \nabla \mathbf{u}^{\intercal})$$

rotation rate tensor

$$\overset{\triangle}{\boldsymbol{\sigma}} = 2G(\dot{\boldsymbol{\varepsilon}} - \frac{1}{3}\mathrm{tr}(\dot{\boldsymbol{\varepsilon}})\mathbf{I}) + \frac{1}{3}K\,\mathrm{tr}(\dot{\boldsymbol{\varepsilon}})\mathbf{I} ,$$

Zaremba-Jaumann rate of the Cauchy stress

bulk modulus

$$K = \frac{2(1+\nu)}{3(1-2\nu)}G$$

$$\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^{\intercal})$$

(strain rate, elastic regime)

plastic strain rate

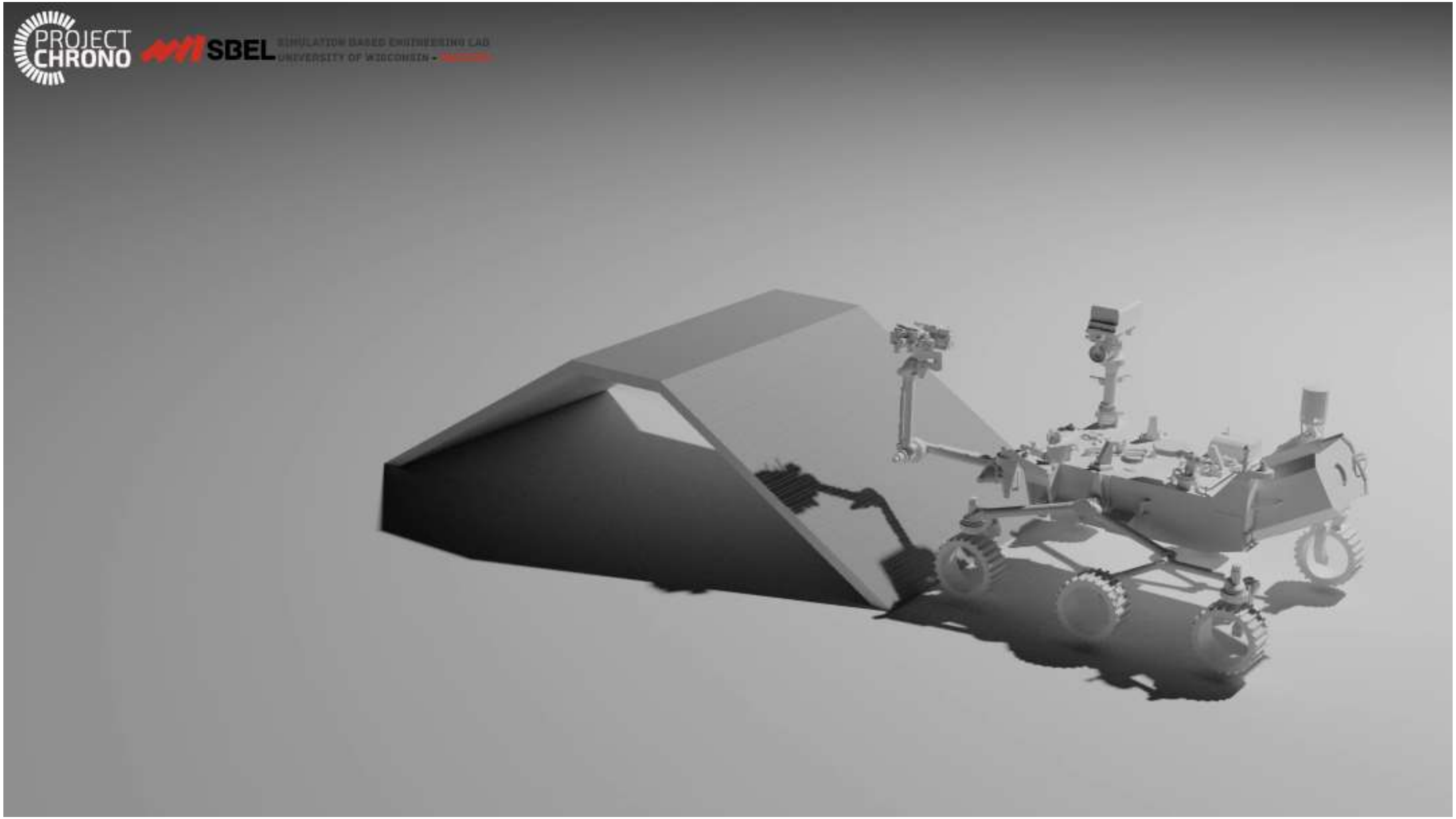
$$\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^{\intercal}) - \frac{1}{\sqrt{2}}\dot{\lambda}\frac{\boldsymbol{\tau}}{\bar{\tau}}$$

(strain rate, plastic regime)

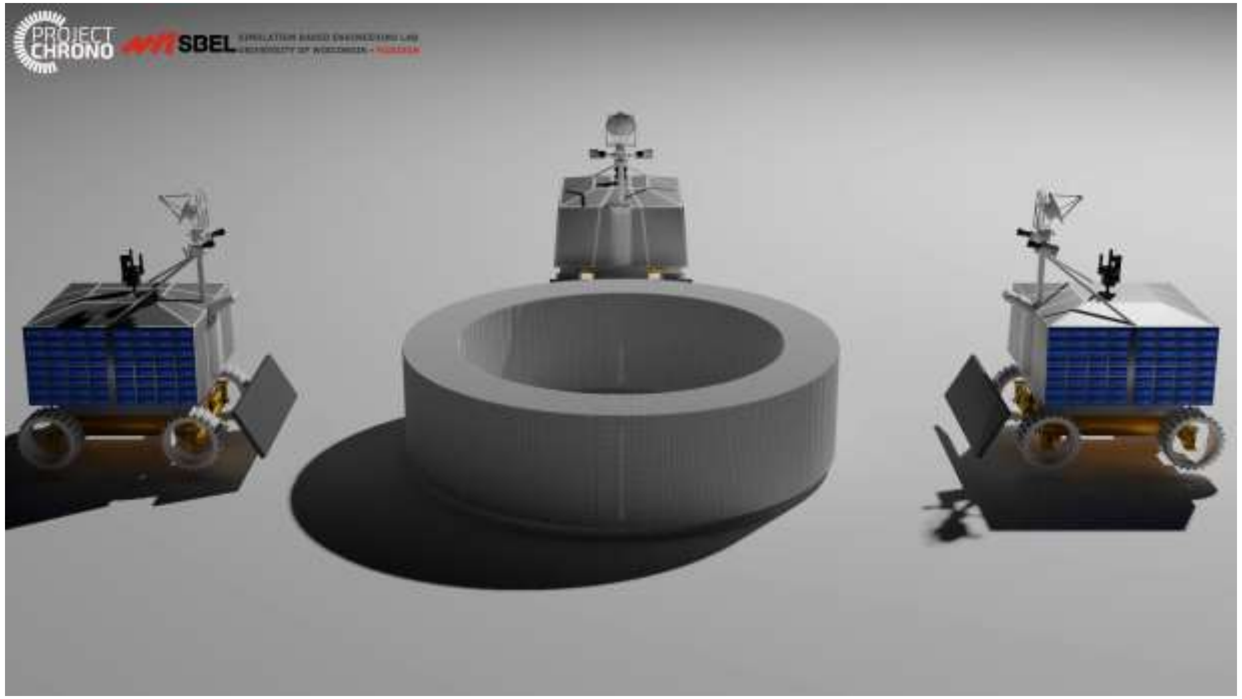
equivalent shear stress

Sachith Dunatunga and Ken Kamrin. "Continuum modelling and simulation of granular flows through their many phases." Journal of Fluid Mechanics 779 (2015): 483-513.

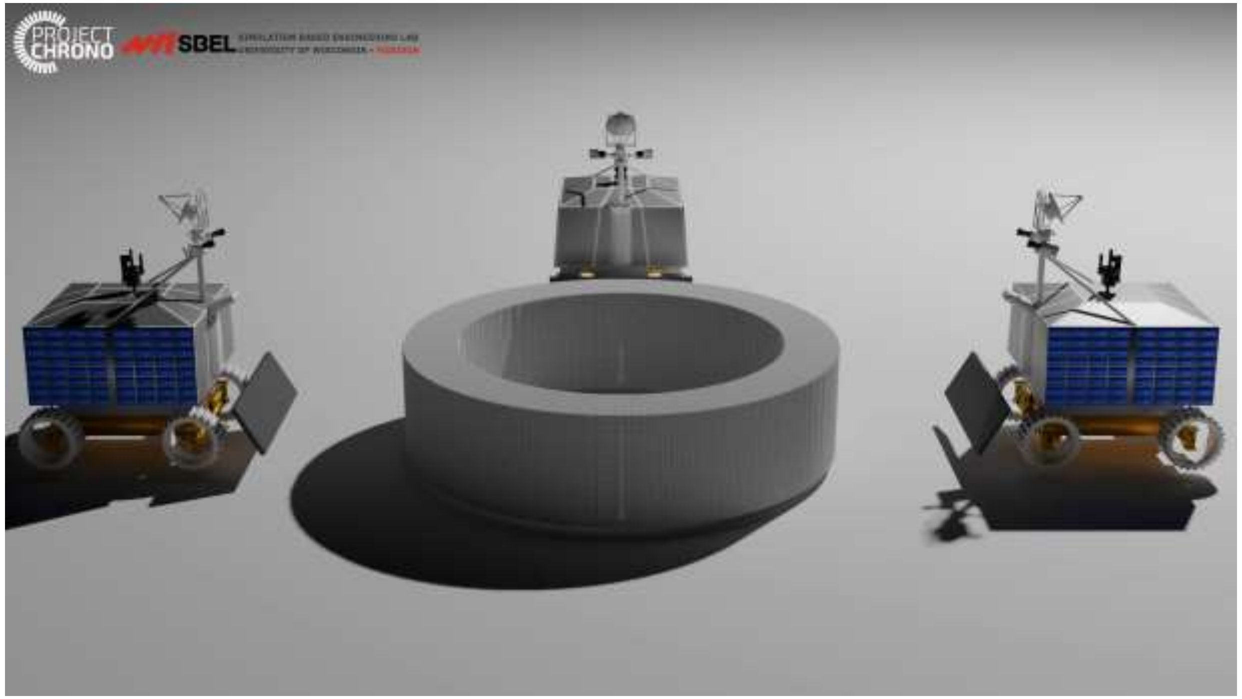
Terramechanics w/ CRM [courtesy of SPH]



Bulldozing, under Earth gravity & Moon Gravity



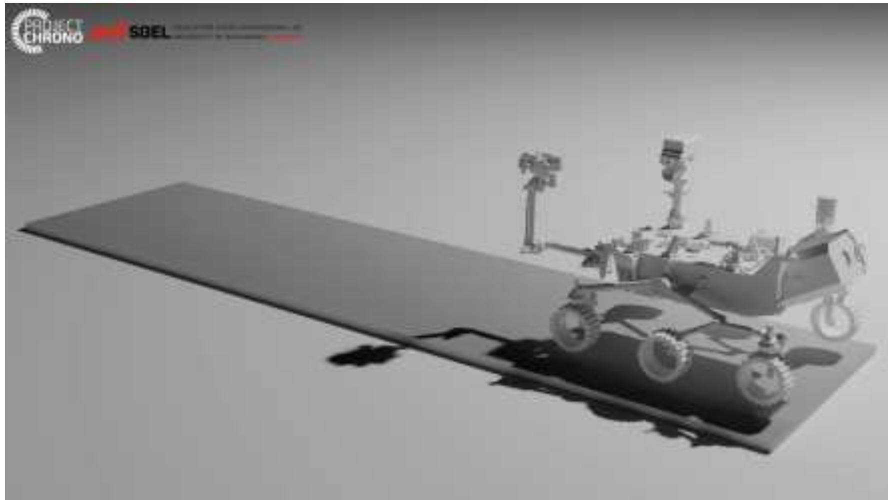
Earth gravity



Moon gravity

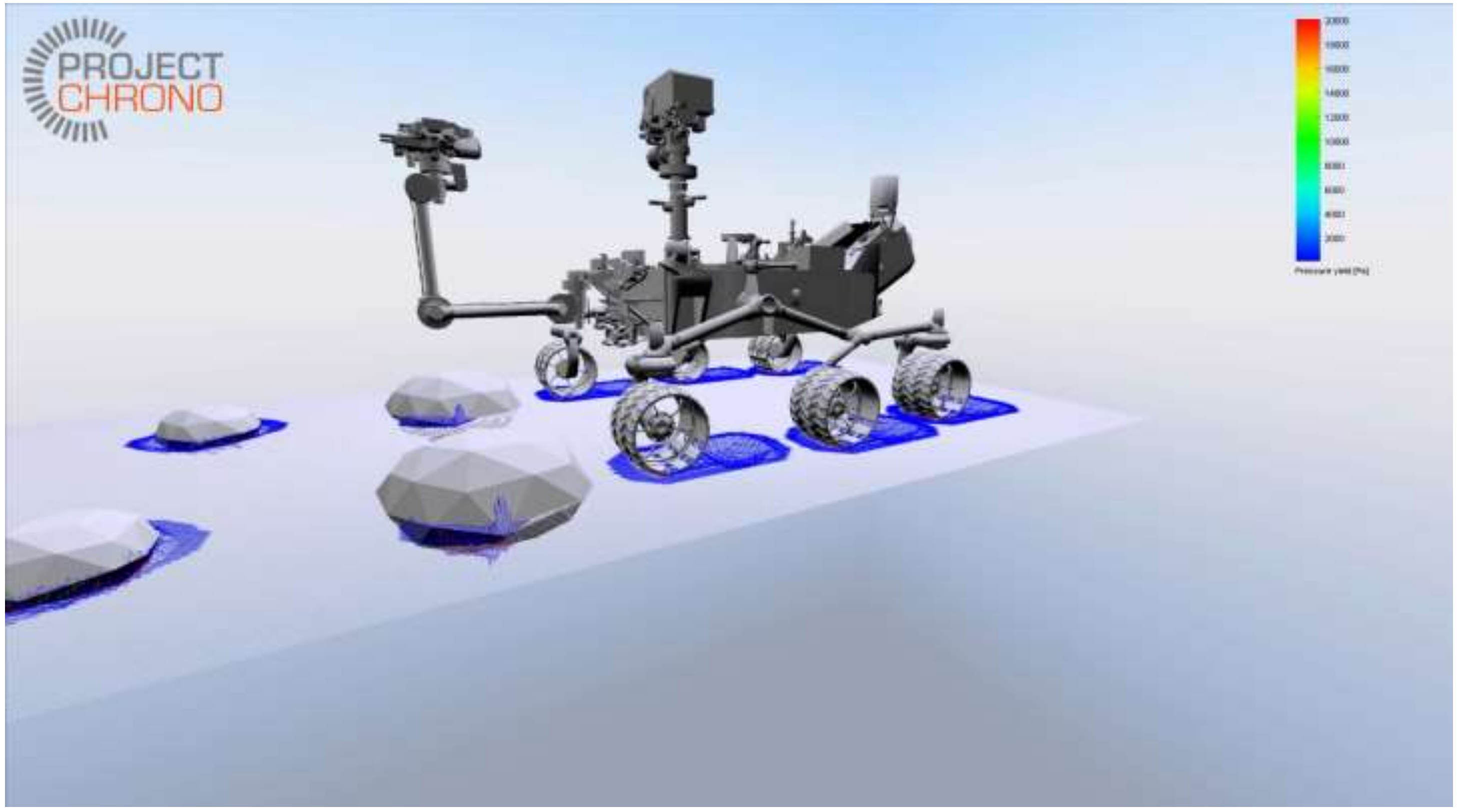
CRM: RTF of 30-300

RTF = 45



- Number of SPH markers: 2.5 M
- Step size: 2.5e-4 s
- Element size: 0.01 m
- Simulation time: 20 s
- Runtime: 15 mins
- Device: A100 GPU

Soil Contact Model (SCM) is faster…



Soil Contact Model (SCM)
(RTF ≈ 1)

The train of thought, from CRM to SCM

- SCM is fast
- SCM is often accurate enough
- When SCM is both fast and accurate, no point in using CRM (let alone DEM)

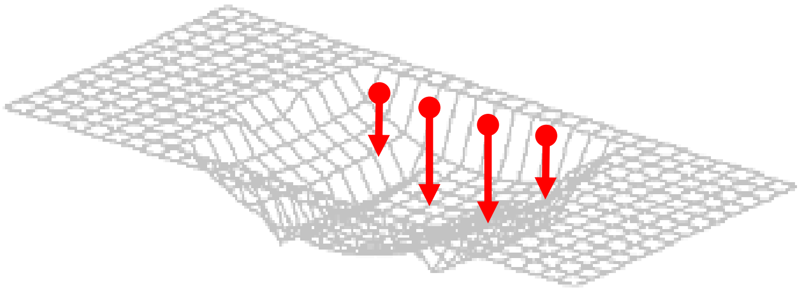
- Can I get SCM to be a good proxy for SCM?
- How do I go about this?

SCM formulation of deformable terrain

- SCM draws on the semi-empirical Bekker-Wong theory
- Pressure p related to sinkage z:

$$p = \left(\frac{K_c}{b} + K_\varphi\right) z^n$$

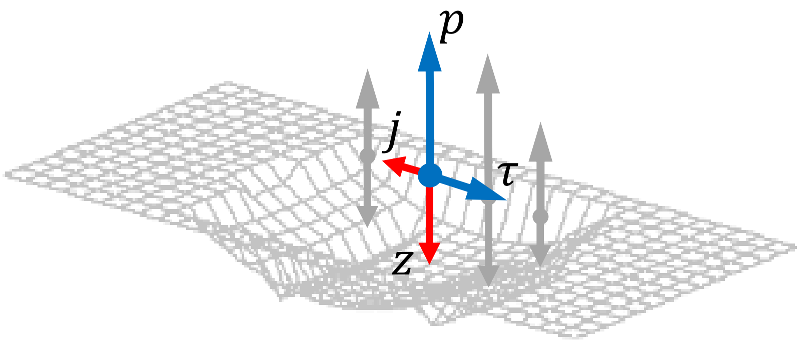
  - Parameters: K_φ, K_c, n, as in Bekker-Wong



- Tangential stress τ given by Janosi-Hanamoto:

$$\tau = \tau_{max}\left(1 - e^{-j/k}\right)$$
$$\tau_{max} = c + p \tan\varphi$$

  - Parameters: c cohesion, φ internal friction angle



Bevameter to the rescue…

[credit: Keweenaw Research Center, Michigan Tech University]

The idea anchoring the proposed approach to calibrate SCM

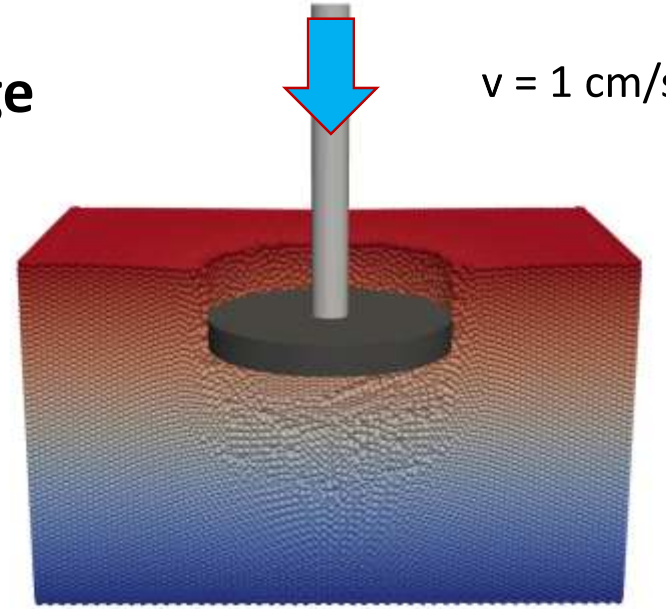
Rely on a virtual Bevameter test that is run using CRM

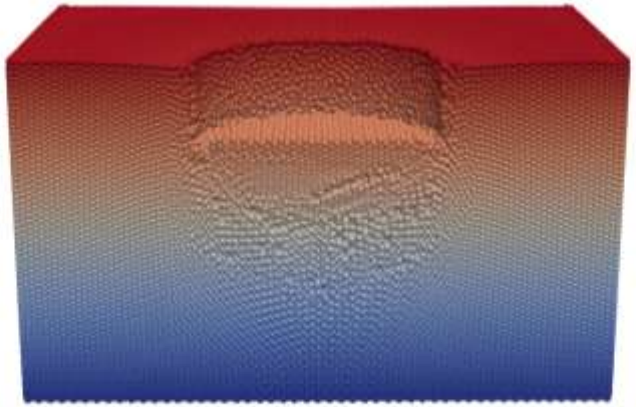
Virtual Bevameter in Chrono via CRM in Two Stages

Simulation-Based Engineering Lab SBEL

- **Stage 1: Plate sinkage**

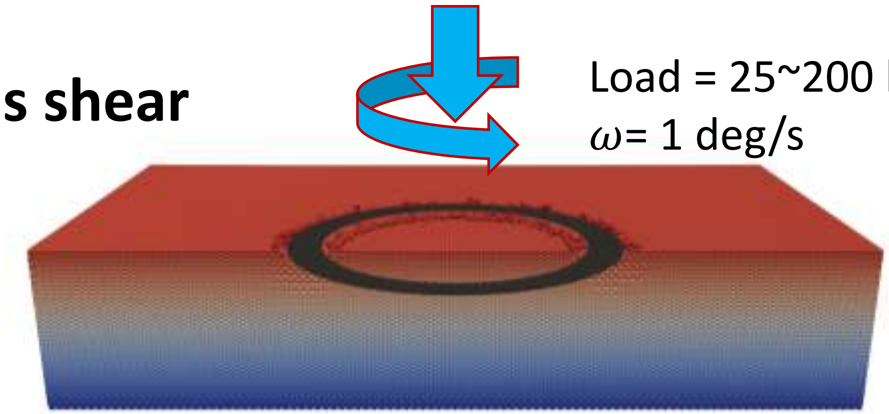
v = 1 cm/s

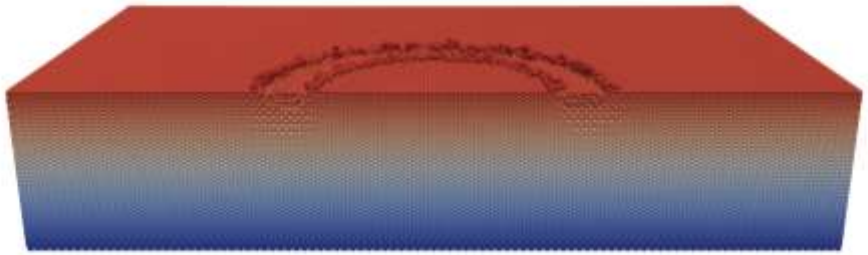




- **Stage 2: Annulus shear**

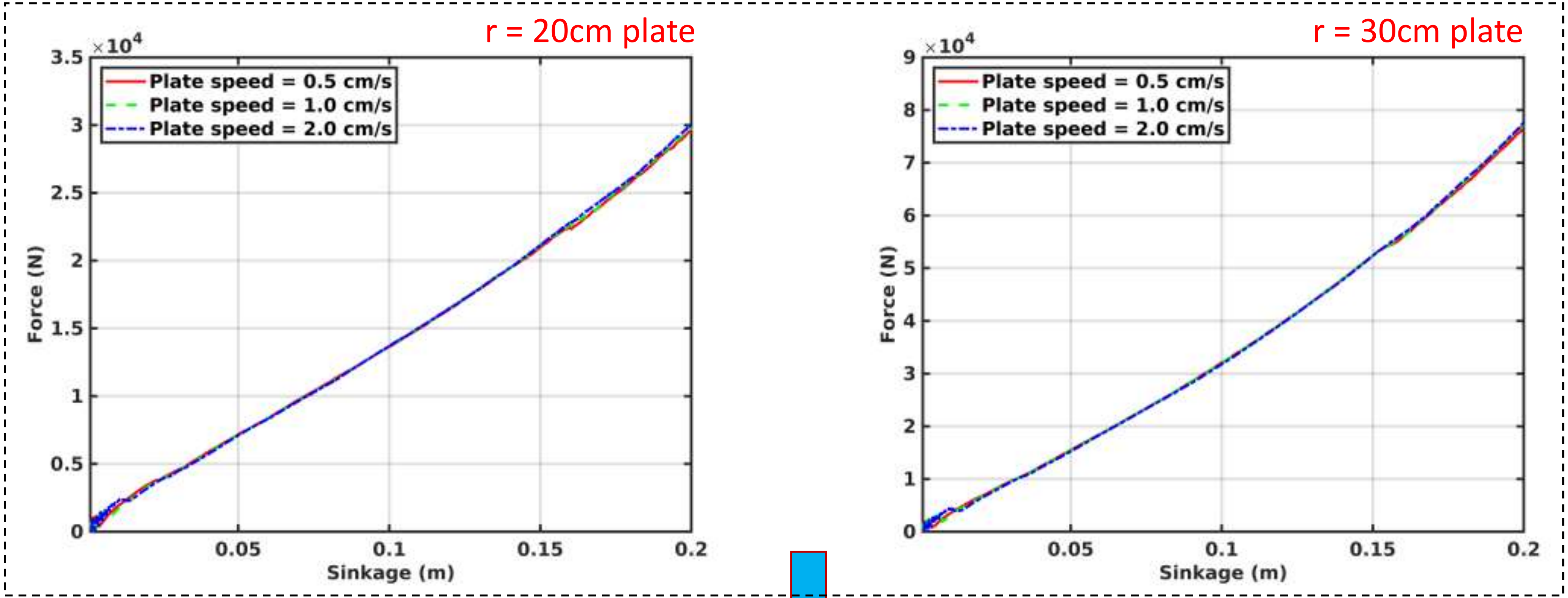
Load = 25~200 kg
ω= 1 deg/s





Virtual Bevameter in Chrono via CRM

- **Plate sinkage test**



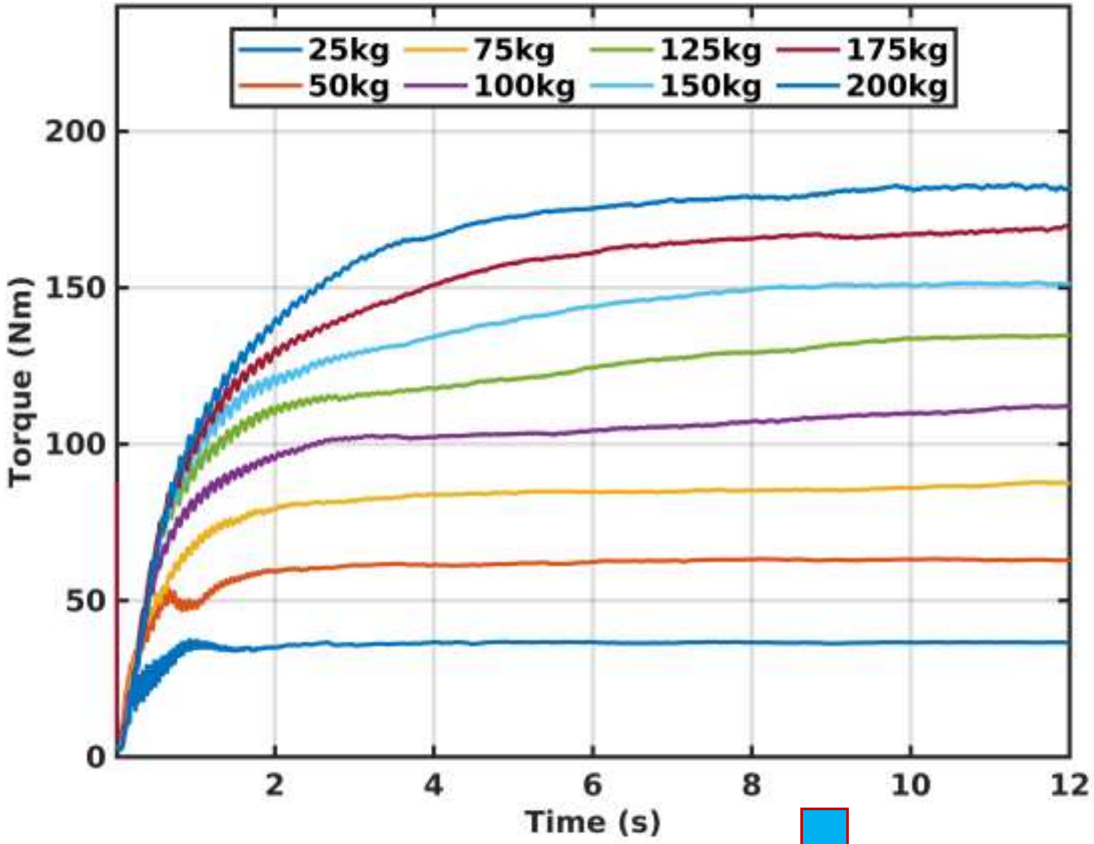
Collected Data

Sinkage (m)	0.025	0.05	0.075	0.1	0.125	0.15	0.175	0.2
Force (N) - 20 cm plate	3905	7102	10278	13693	17203	21148	25221	30012
Force (N) - 30 cm plate	7799	15232	23329	31788	41513	52356	63854	77640

Virtual Bevameter in Chrono via CRM

- **Annulus shear test**

Inner radius = 15 cm
Outer radius = 20 cm



Collected Data

Load (kg)	25	50	75	100	125	150	175	200
Torque (N-m)	36	62	84	105	124	141	157	170

Virtual Bevameter in Chrono via CRM

”Experimental” data

Sinkage (m)	0.025	0.05	0.075	0.1	0.125	0.15	0.175	0.2
Force (N) - 20 cm plate	3905	7102	10278	13693	17203	21148	25221	30012
Force (N) - 30 cm plate	7799	15232	23329	31788	41513	52356	63854	77640

Load (kg)	25	50	75	100	125	150	175	200
Torque (N-m)	36	62	84	105	124	141	157	170

Bayesian Framework – used to calibrate low fidelity model using "ground truth" generated by the high-fidelity model

- At each time step t_i, $1 \leq i \leq N$, assume the difference between the data produced by the model $F(i;q)$ (working with a set of model parameters q), and the actual observation $F^{obs}(i)$ is some zero-mean, normally distributed random variable:

$$F^{obs}(i) = F(i;q) + \epsilon_i \text{ , with } \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

- This implies a Gaussian likelihood function of the form

$$\mathcal{L}(F^{obs}|q) = \exp(\frac{-SS_q}{2\sigma^2})$$

$$SS_q \equiv \sum_{i=1}^{N} \left[F^{obs}(i) - F(i;q)\right]^2$$

- Use Bayes' formula to obtain the posterior distribution

$$\pi(q|F^{obs}) = \frac{\mathcal{L}(F^{obs}|q) \cdot \pi(q)}{\int \mathcal{L}(F^{obs}|q) \cdot \pi(q) \, dq}$$

- I'm interested in that value of q that explains the best the observed data F_{obs}

Calibration of SCM model using CRM “ground truth”

Most likely SCM parameters

SCM parameter	K_c (N/m^{n+1})	K_ϕ (N/m^{n+2})	n	c (Pa)	φ (deg)
Calibrated value	-1.1e5	2.22e6	1.2	2496	24

Validation the SCM model parameters obtained w/ CRM

Single wheel validation

Rig information

- Slip ratio: 0 - 0.8
- Fixed angular velocity: 1 rad/s
- Translational velocity: $v = \omega r_w(1 - slip)$

Wheel information

Offroad rigid wheel

- Mass: 108 kg
- Radius: 0.47 m
- Width: 0.25 m

VIPER wheel

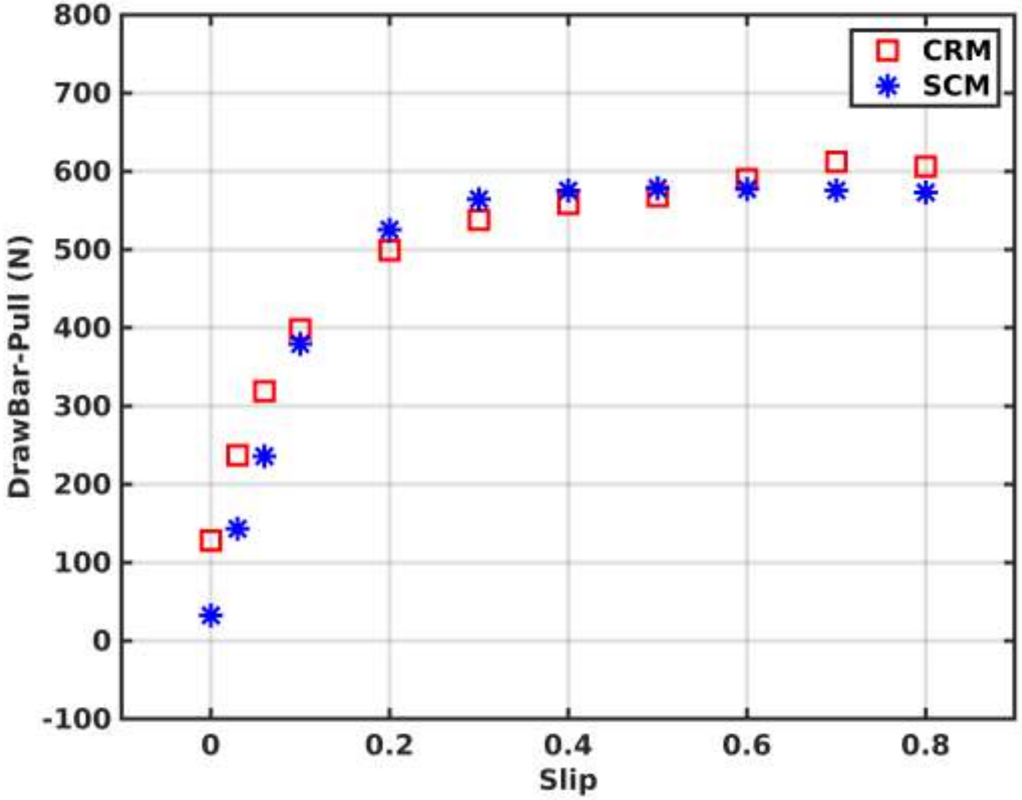
- Mass: 108 kg
- Radius: 0.25m
- Width: 0.29m

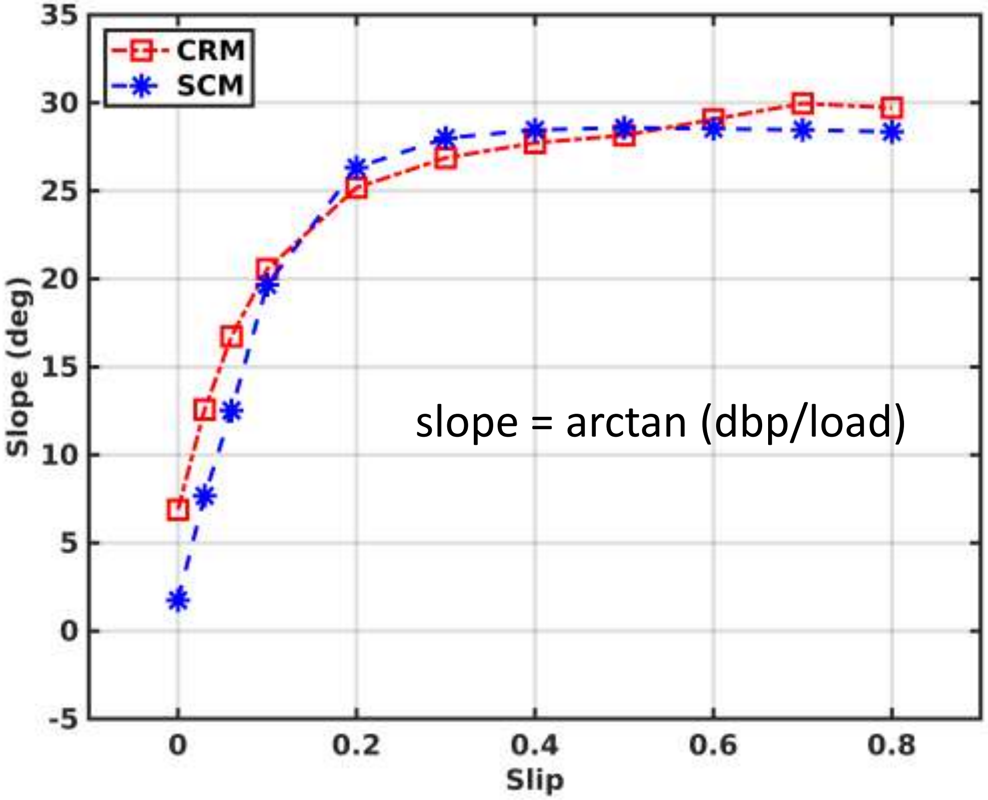
Validation of SCM model parameters obtained w/ CRM

Single wheel validation - wheel without grouser

Offroad rigid wheel

$$v = \omega r_w (1 - slip)$$





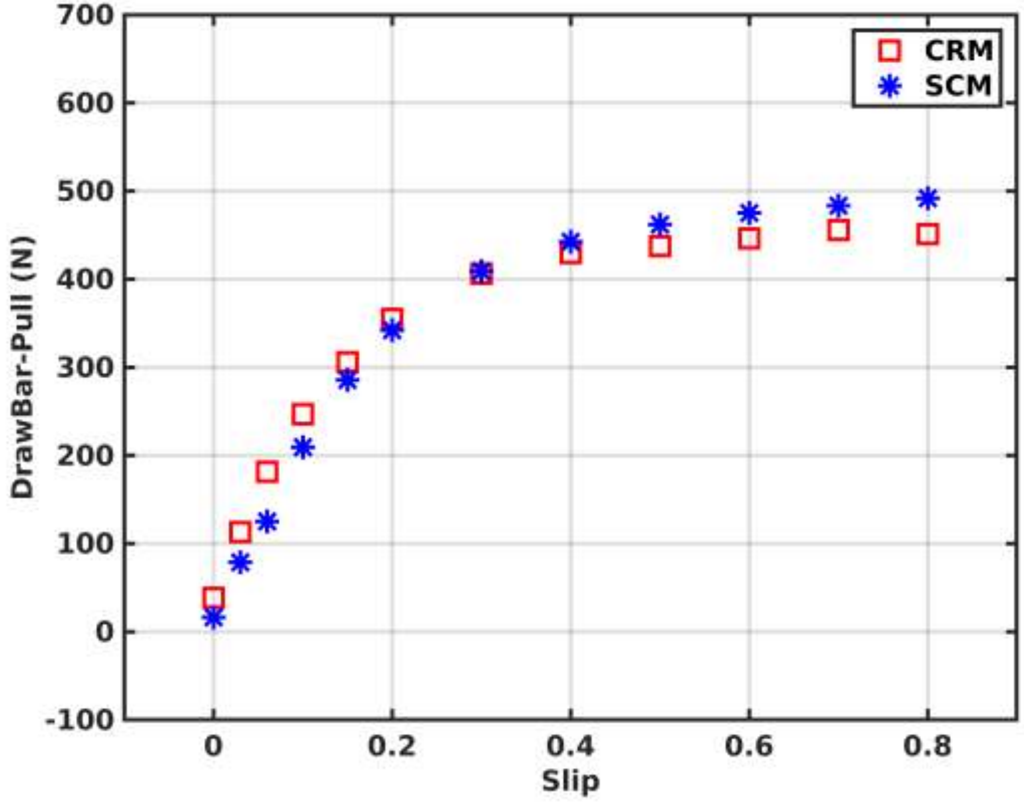
CRM vs. SCM simulation

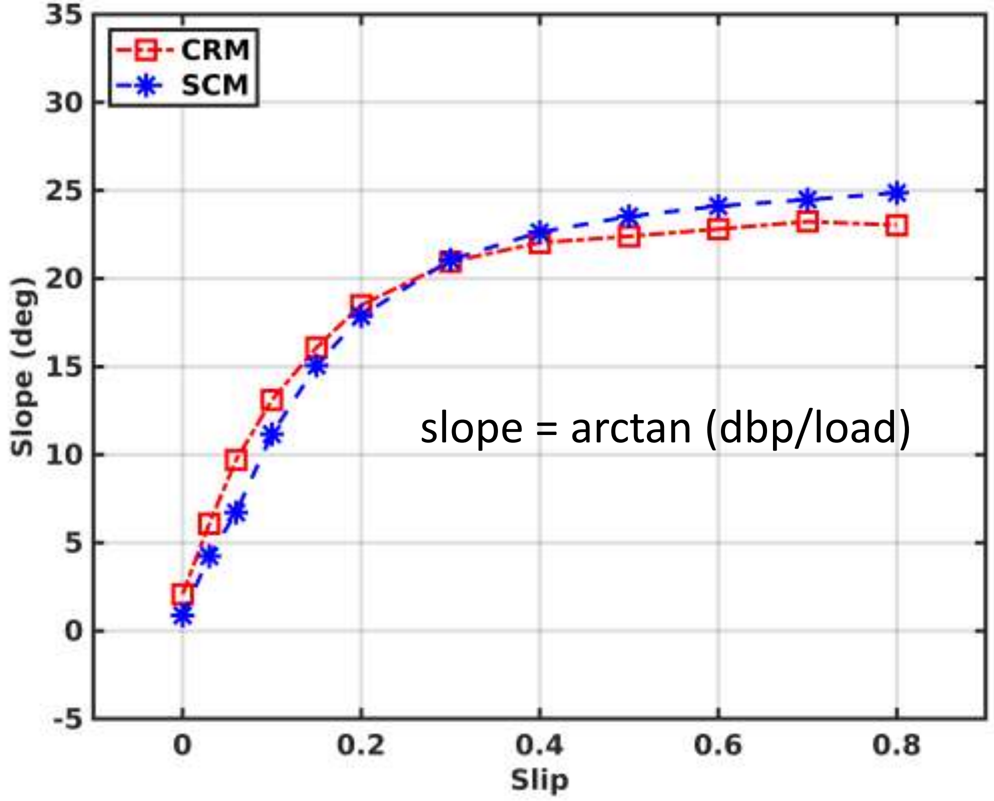
Validation of SCM model parameters obtained w/ CRM

Single wheel validation - wheel with grouser

VIPER wheel

$$v = \omega r_w (1 - slip)$$





CRM vs. SCM simulation

Validation of the parameters of the SCM model

Full rover validation - Moon VIPER rover

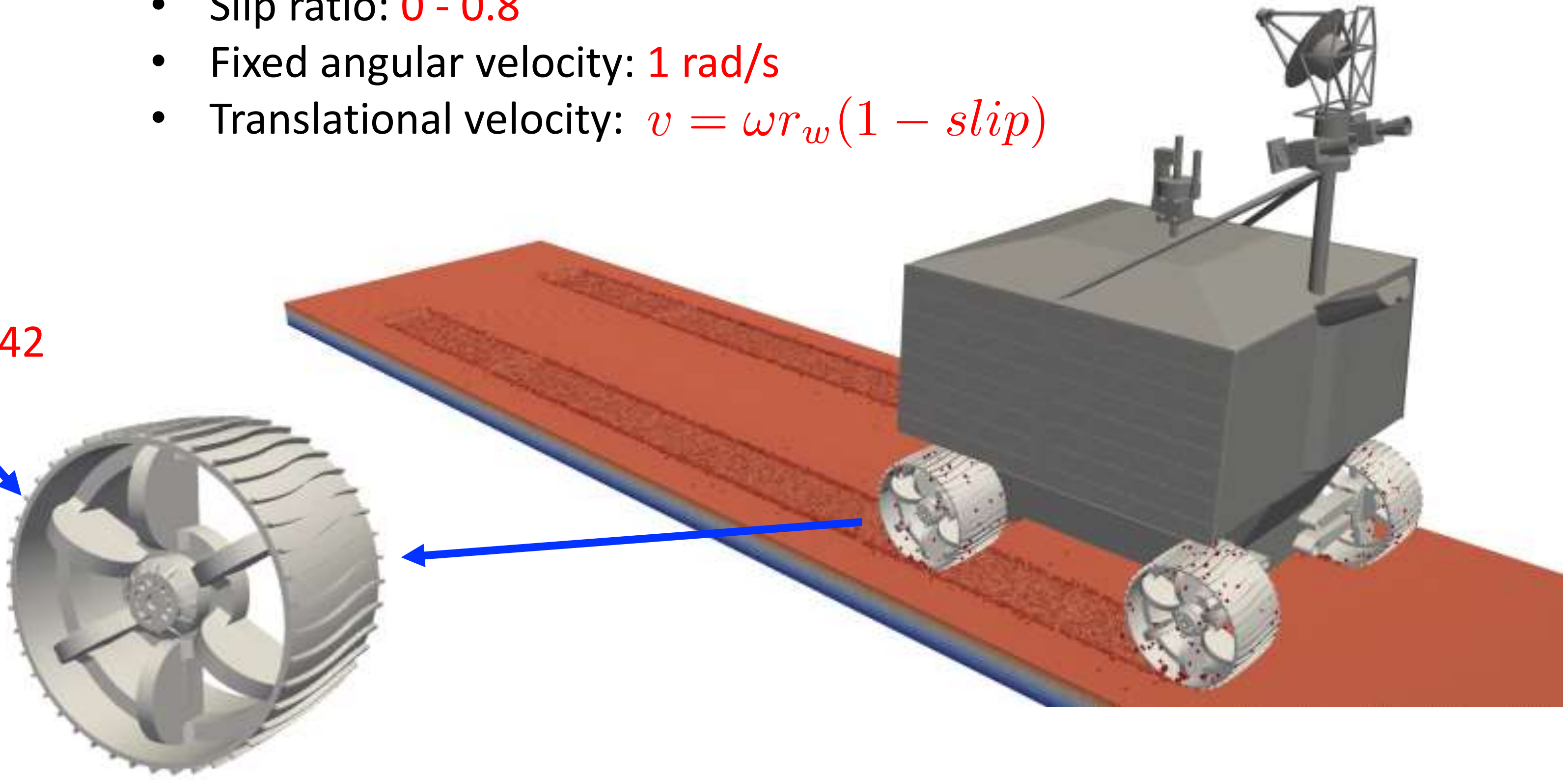
Rover information

- Mass: 430 kg
- Radius: 0.24 m
- Width: 0.29 m

Rig information

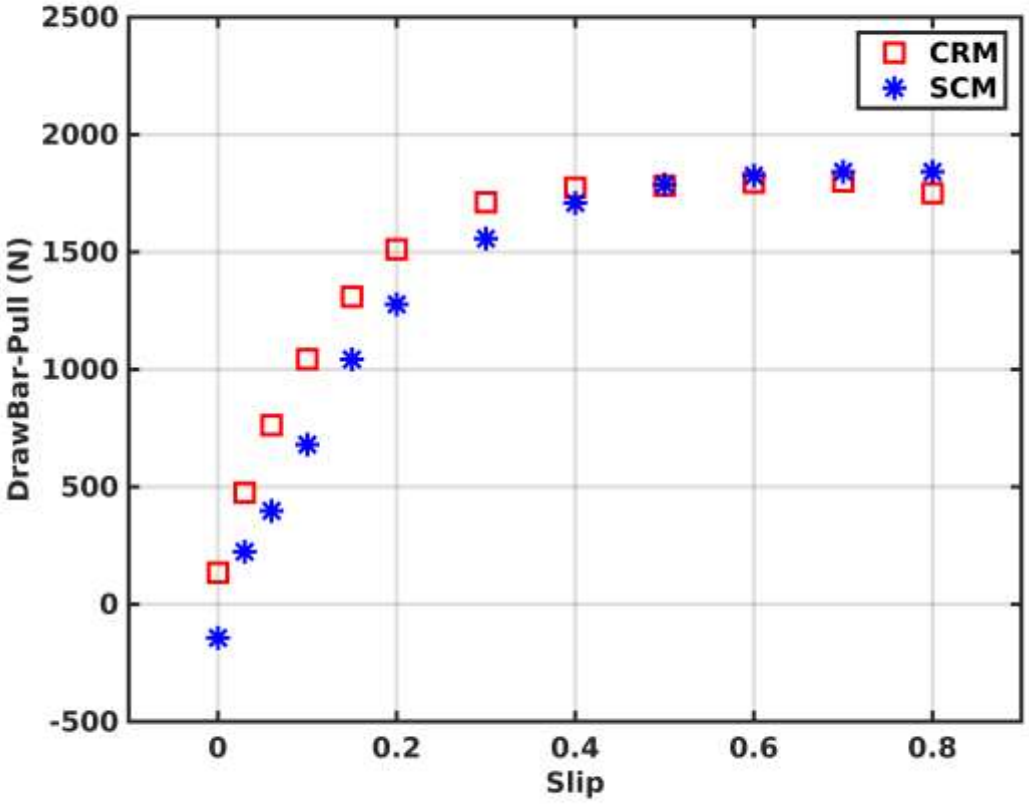
- Slip ratio: 0 - 0.8
- Fixed angular velocity: 1 rad/s
- Translational velocity: $v = \omega r_w (1 - slip)$

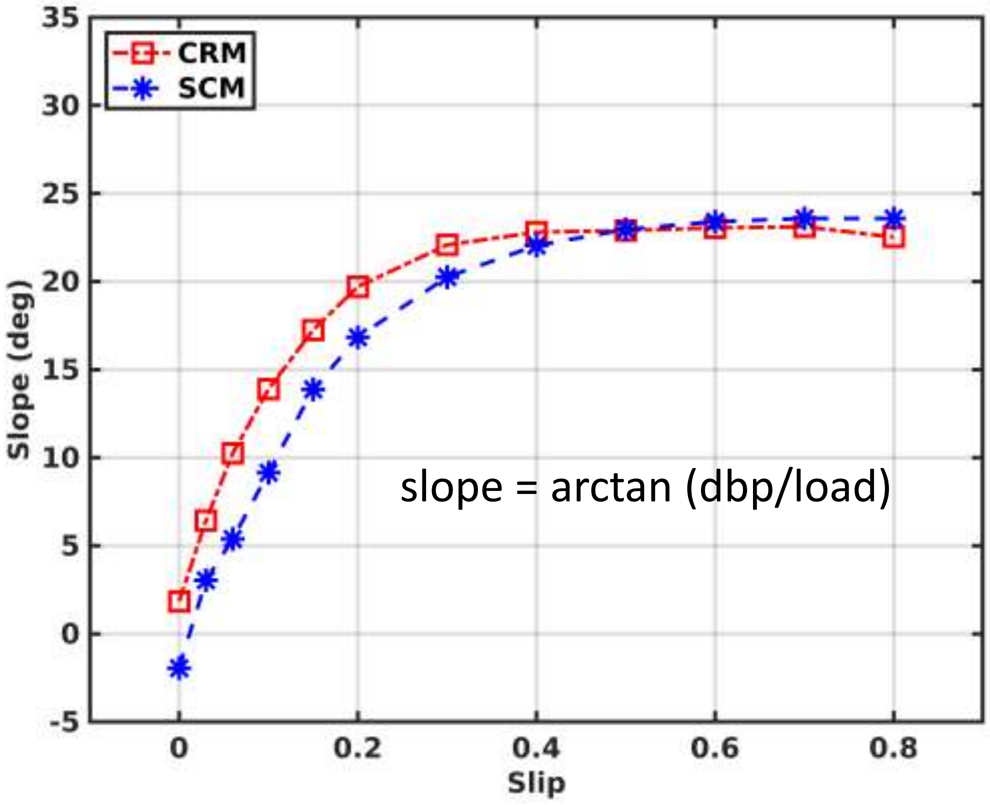
Number of grousers: 42
Grouser height: 1 cm



Validation of the parameters of the SCM model

Full rover validation - Moon VIPER rover





CRM vs. SCM simulation

Simulation in robotics, what it calls for

- Simulate the process of sensing
- Simulate the robot/rover/autonomous vehicle
- Simulate the world in which the robot/rover/autonomous vehicle operates

Closing thoughts

- Simulation can play a role in:
  - Designing better automation
  - Testing chips, in the loop
  - Testing human-robot interaction

- This “simulation-in-robotics” field is nascent
  - Lots and lots of open problems

ART/ATK & Chrono

ART repo	https://github.com/uwsbel/autonomy-research-testbed	
ATK repo	https://github.com/uwsbel/autonomy-toolkit	

Chrono Websites	projectchrono.org	
	projectchrono.org/pychrono	
Software	github.com/projectchrono/chrono	
	anaconda.org/projectchrono/pychrono	
Latest developments	github.com/projectchrono/chrono/blob/develop/CHANGELOG.md	
Documentation	api.projectchrono.org	(develop version)
	api.projectchrono.org/7.0.0	(release 7.0.2)
User forum	groups.google.com/forum/#!forum/projectchrono	

PROJECT CHRONO

Thank you.

negrut@wisc.edu
Simulation Based Engineering Lab
University of Wisconsin – Madison
www.linkedin.com/in/dan-negrut-89320719

Lab website: http://sbel.wisc.edu
Chrono website: http://www.projectchrono.org
Source code: https://github.com/projectchrono
Movies: https://www.youtube.com/channel/UCpIhnh9HvfNzBtBcNRecUKw/featured