Enhancing LAMMPS Capabilities

Granular Models, Coding Concepts, CAD Interoperability and Coupling to Continuum Methods

2nd LAMMPS Workshop, Albuquerque, Aug 2011

Christoph Kloss*, Christoph Goniva**, Stefan Amberger, Alice Hager, Andreas Aigner, Michael Friedl, Stefan Pirker***

*christoph.kloss@jku.at, ** christoph.goniva@jku.at, ***stefan.pirker@jku.at

all CD Laboratory on Particulate Flow Modelling

Johannes Kepler University Linz, Austria

www.liggghts.com | www.cfdem.com | www.particulate-flow.at

Outline

Open Source DEM and CFD-DEM
Outline

1. **Introduction**
   Our Group, Mission Statement

2. **Models**
   Granular Contact Models, Heat Transfer, SPH, Multisphere Model

3. **CAD Interoperability**
   Mesh Import, Particle Insertion, Mesh regions, Wear Prediction

4. **Features and Coding Concepts**
   Load Balancing, Per-Particle Properties, Transport Equations

5. **Coupling to Continuum Methods**
   OpenFOAM®, Resolved and Unresolved CFD-DEM, Multiphysics

I. Introduction

About Us / Mission Statement

About Us
Mission Statement

CD Lab Particulate Flow Modelling (head: Stefan Pirker)

6 Post-Docs, 7 PhD students, 5 master students

Goal: Particulate Flow Modelling...

Research focus:

- Fluid flow → **CFD**, **LB, SPH**
- Granular flow → **CFD, DEM/MD**
- Heat transfer
- Experimental Validation

...of industrially relevant processes

Application examples include:

Blast furnaces, cyclones, pneumatic conveying, fluidized beds, dryers, wet scrubbers, conveyor and chute systems, tablet pressing, powder sintering, hopper flow, soil sampling, energy storage, river bed erosion, blood flows...

The LIGGGHTS/CFDEM Codes
Overview

LIGGGHTS = An Open Source, C++, MPI parallel DEM code

LAMMPS IMPROVED FOR GENERAL GRANULAR AND GRANULAR HEAT TRANSFER SIMULATIONS

CFDEM = Coupling of LIGGGHTS to CFD code OpenFOAM®

CFD-DEM

WWW.LIGGGHTS.COM | WWW.CFDEM.COM

The web platform now has **about 600 registered users**

Published in [1, 2, 3, 4, 10, 11, 12, 13, 14, 16]

The LIGGGHTS+CFDEM Codes
Cooperations

CD Lab

SIEMENS VAI

RHI

Christian Doppler Forschungsgesellschaft

voestalpine

PLANSEE

BOREALIS
SHAPING *the* FUTURE *with* PLASTICS

ThyssenKrupp

JKU
JOHANNES KEPLER UNIVERSITY LINZ

Further (formal and informal) cooperations for LIGGGHTS and CFDEM

BASF The Chemical Company

Sandia National Laboratories

EPFL ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

ZÜBLIN

IWF

MECALYSIS analyse ∘ optimise ∘ realise

TU/e Technische Universiteit Eindhoven University of Technology
Where innovation starts

INTRAME

Aalto University School of Science and Technology

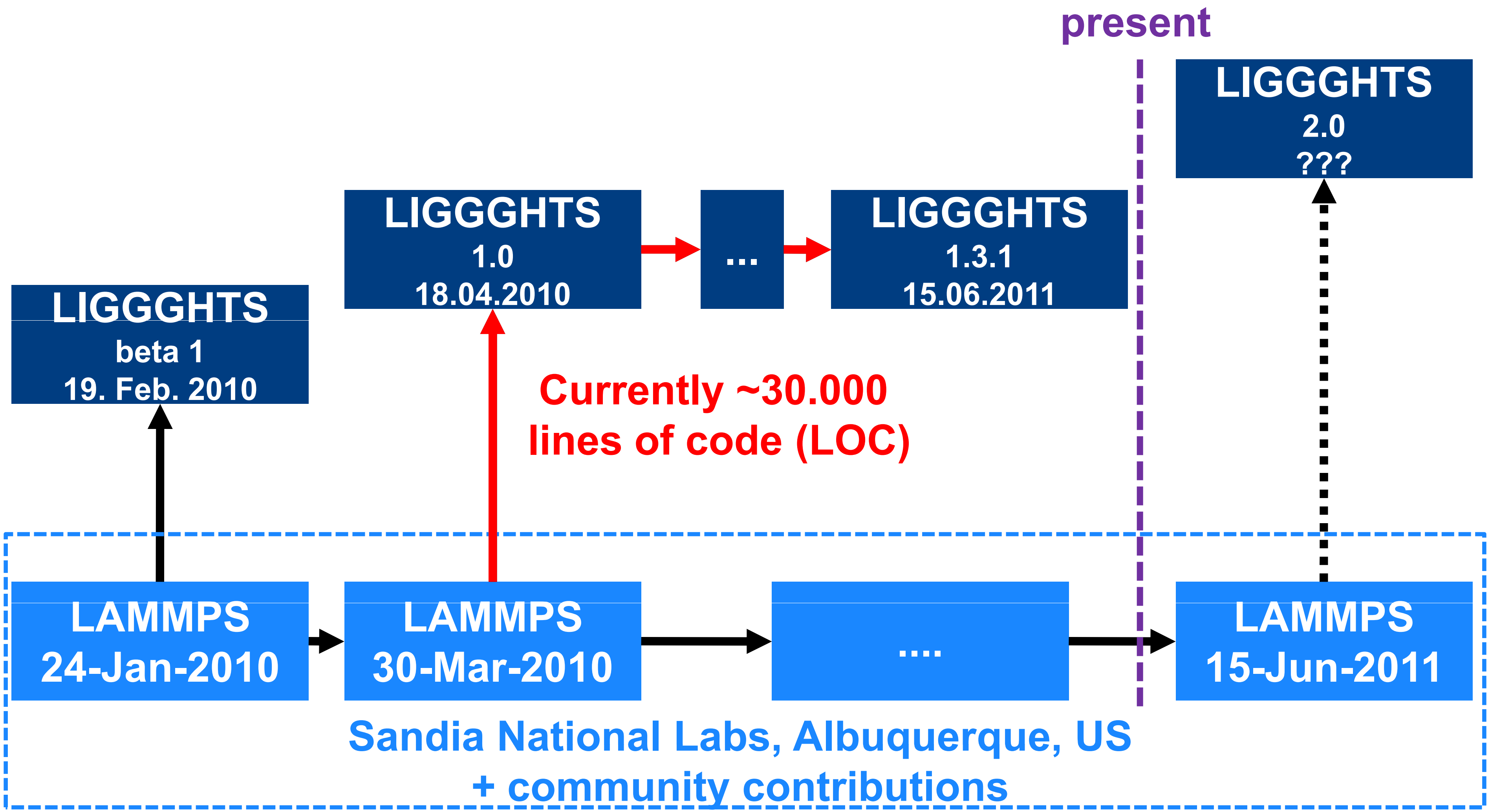
CeParTec GmbH Centre of Particle Technology

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

ArcelorMittal

DLR

The LIGGGHTS Code
Relation to LAMMPS

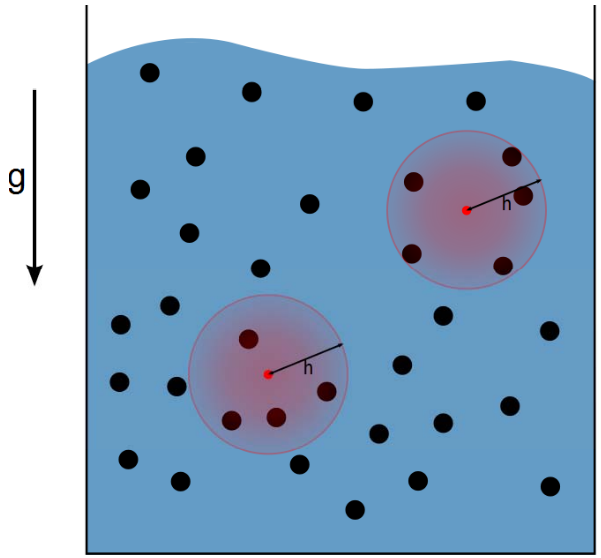


II. Models

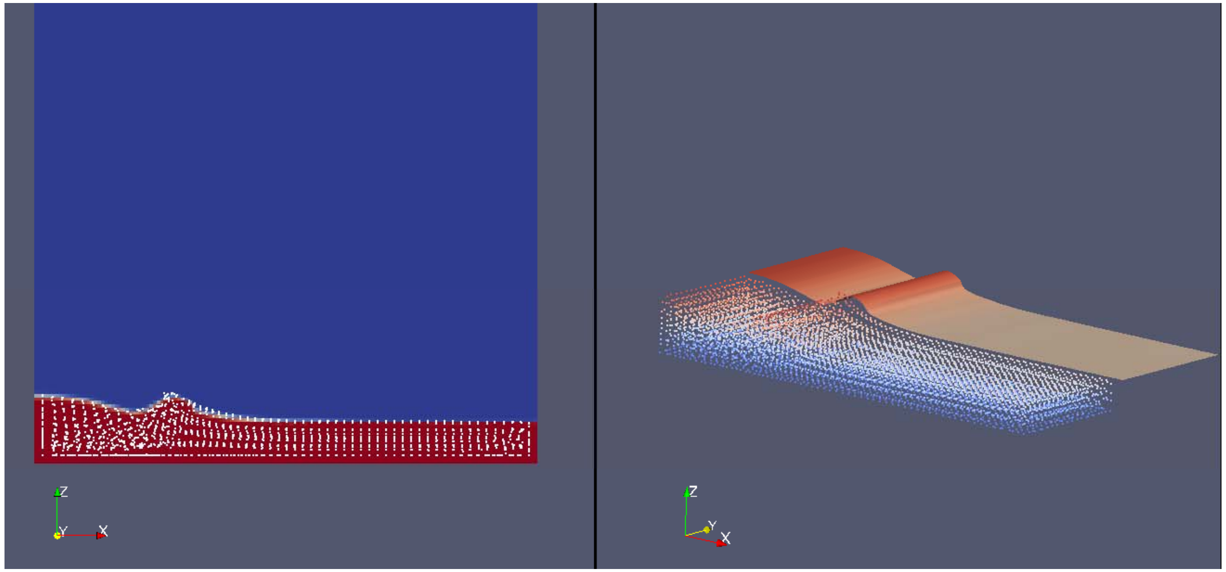
LIGGGHTS Models - Released
SPH – Smoothed Particle Hydrodynamics

SPH is well-suited for geometrically complex systems, and cases where numerical diffusion of Finite Volume approach is not acceptable

Released in LIGGGHTS 1.4 – commands pair sph, fix sph/density/continuity, fix sph/density/corr, fix sph/pressure/summation, fix wall/sph, fix wall/region/sph



Principle of SPH



Sloshing Tank with SPH in LIGGGHTS vs. Volume of Fluid (VOF) Method in OpenFOAM®

LIGGGHTS Models - Released

Hertz-Mindlin (HM) Granular Contact Model

$$F = \underbrace{(k_n \underbrace{\delta n_{ij}}_{\substack{\text{normal}\\ \text{overlap}}} - \gamma_n \underbrace{vn_{ij}}_{\substack{\text{normal}\\ \text{relative vel.}}})}_{\text{normal force}} + \underbrace{(k_t \underbrace{\delta t_{ij}}_{\substack{\text{tangential}\\ \text{overlap}}} - \gamma_t \underbrace{vt_{ij}}_{\substack{\text{tangential}\\ \text{relative vel.}}})}_{\text{tangential force}}$$

$$k_n = \frac{4}{3} Y^* \sqrt{R^* \delta_n},$$

$$\gamma_n = -2\sqrt{\frac{5}{6}}\beta\sqrt{S_n m^*} \geq 0,$$

$$k_t = 8G^* \sqrt{R^* \delta_n},$$

$$\gamma_t = -2\sqrt{\frac{5}{6}}\beta\sqrt{S_t m^*} \geq 0.$$

$$S_n = 2Y^* \sqrt{R^* \delta_n}, \qquad S_t = 8G^* \sqrt{R^* \delta_n}$$

$$\beta = \frac{\ln(e)}{\sqrt{\ln^2(e) + \pi^2}}, \quad \frac{1}{Y^*} = \frac{(1-\nu_1^2)}{Y_1} + \frac{(1-\nu_2^2)}{Y_2},$$

$$\frac{1}{G^*} = \frac{2(2+\nu_1)(1-\nu_1)}{Y_1} + \frac{2(2+\nu_2)(1-\nu_2)}{Y_2}$$

$$\frac{1}{R^*} = \frac{1}{R_1} + \frac{1}{R_2}, \quad \frac{1}{m^*} = \frac{1}{m_1} + \frac{1}{m_2}$$

Y...Young's modulus G...Shear modulus

ν...Poisson ratio e...coeff.of restitution

LIGGGHTS Models - Released

Hertz-Mindlin (HM) Contact Model

$$F = \underbrace{(k_n \underbrace{\delta n_{ij}}_{\text{normal overlap}} - \gamma_n \underbrace{v n_{ij}}_{\text{normal relative vel.}})}_{\text{normal force}} + \underbrace{(k_t \underbrace{\delta t_{ij}}_{\text{tangential overlap}} - \gamma_t \underbrace{v t_{ij}}_{\text{tangential relative vel.}})}_{\text{tangential force}}$$

$$k_n = \frac{4}{3} Y^* \sqrt{R^* \delta_n},$$

$$\gamma_n = -2\sqrt{\frac{5}{6}}\, \beta \sqrt{S_n m^*} \geq 0,$$

$$k_t = 8 G^* \sqrt{R^* \delta_n},$$

$$\gamma_t = -2\sqrt{\frac{5}{6}}\, \beta \sqrt{S_t m^*} \geq 0.$$

$$S_n = 2Y^* \sqrt{R^* \delta_n}, \qquad S_t = 8G^* \sqrt{R^* \delta_n}$$

$$\beta = \frac{ln(e)}{\sqrt{ln^2(e) + \pi^2}}, \quad \frac{1}{Y^*} = \frac{(1-\nu_1^2)}{Y_1} + \frac{(1-\nu_2^2)}{Y_2},$$

$$\frac{1}{G^*} = \frac{2(2+\nu_1)(1-\nu_1)}{Y_1} + \frac{2(2+\nu_2)(1-\nu_2)}{Y_2}$$

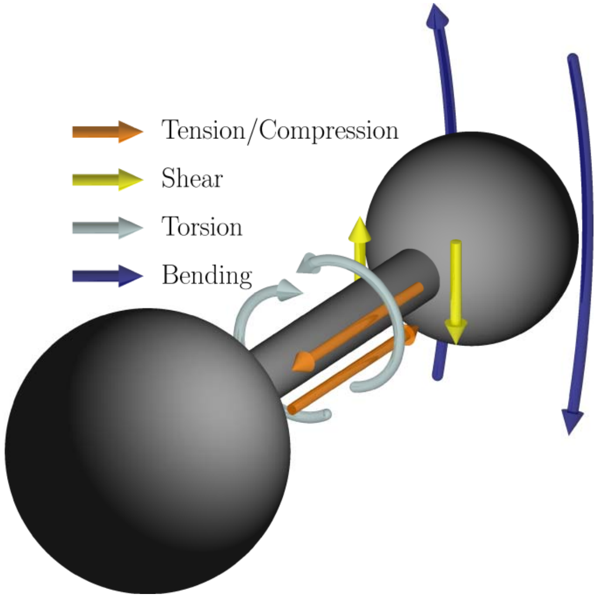
$$\frac{1}{R^*} = \frac{1}{R_1} + \frac{1}{R_2}, \quad \frac{1}{m^*} = \frac{1}{m_1} + \frac{1}{m_2}$$

Y...Young's modulus G...Shear modulus

ν...Poisson ratio e...coeff.of restitution

LIGGGHTS Models – Unreleased
Particle Bonds

Particle bonds are inter-particle connections able to resist



- **tension/compression,**
- **shear,**
- **torsion and**
- **bending**

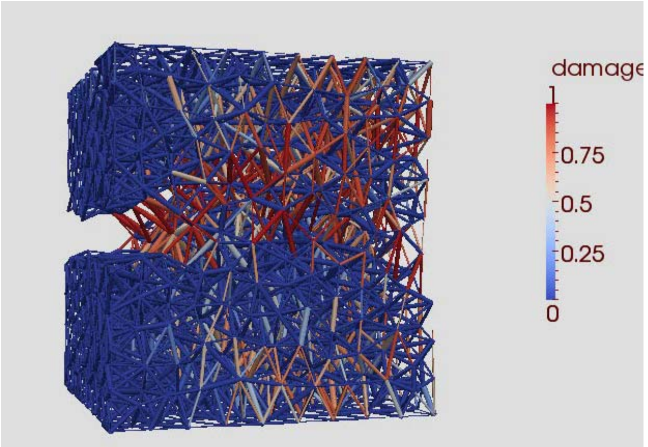
up to a certain breakage limit. Once broken, the bond connection is lost permanently. Particle bonds and related models can be used to **model soils, „glued particles", particle breakage, crack formation in beams** („lattice beam model") etc...

Looking for co-workers to make model ready for release!

Asaf et. al, Soil & Tillage Research 92 (2007) 227–242 "Determination of discrete element model parameters required for soil tillage"
Shmulevich: Soil & Tillage Research 111 (2010) 41–53 "State of the art modeling of soil–tillage interaction using discrete element method"
Zhang, Li: Journal of Terramechanics 43 (2006) 303–316 "Simulation on mechanical behavior of cohesive soil by Distinct Element Method",

Pics from: Latham, S, Weatherly, D.,: Scripting Parallel Discrete Element Simulations with ESyS_Particle, https://twiki.esscc.uq.edu.au/bin/view/ESSCC/ESySParticleDownload

Lattice Beam Model
Dynamic fracture roughness in a concrete microstructure

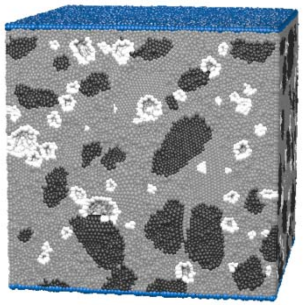


LIGGGHTS will incorporate a beam lattice model (e.g. *,) used to study fracture and fragmentation phenomena.**

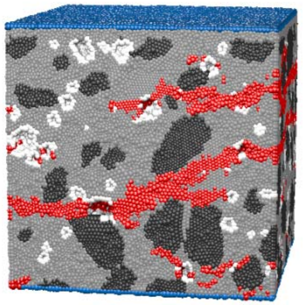
*** G. Lilliu, J.G.M. van Mier, 3D lattice type fracture model for concrete, Engrg. Fract. Mech. 70 (2003) 927–941.**
****J.G.M. van Mier, E. Schlangen, A. Vervuurt, Lattice type fracture models for concrete, in: H.B. Mühlhaus (Ed.), Continuum Models for Materials with Microstructure, John Wiley & Sons, 1995, pp. 341–377.**

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

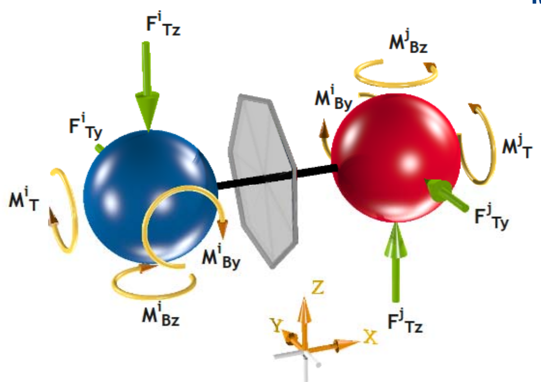
Jean-Francois Jerier
EPF Lausanne
Computational Solid Mechanics



Numerical concrete sample based on a X-ray micro-tomography image (25mm^3)



The crack paths (red) crossing the sample

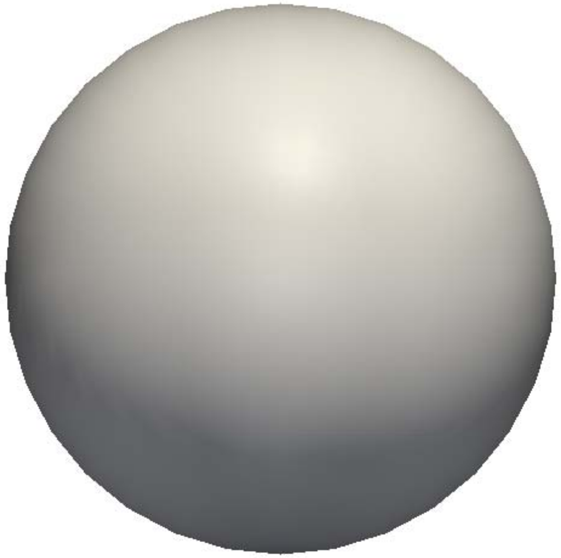


Timoshenko beam theory leads to the interaction between the spheres

500.000 polydisperse spheres and 4 million beams represent this sample
Each simulation calculated on 60 processors in 3 days

LIGGGHTS Models
Capturing non-sphericity

Granular particles outside laboratories are rarely spherical

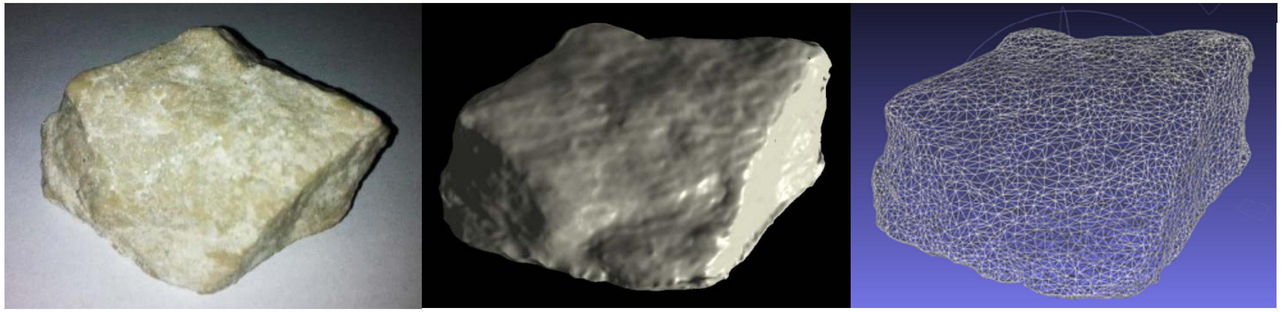


$\neq$

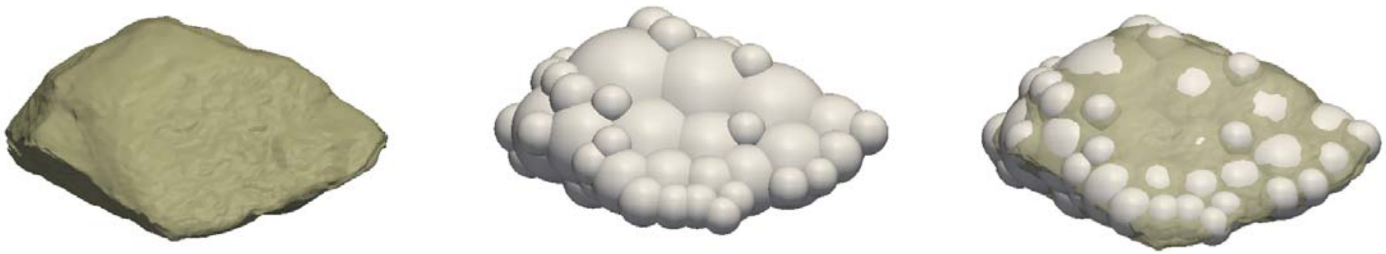
LIGGGHTS Models
Capturing non-sphericity

Multi-Sphere Approach

- **Step 1 – Particle Image by Laser Scanner**



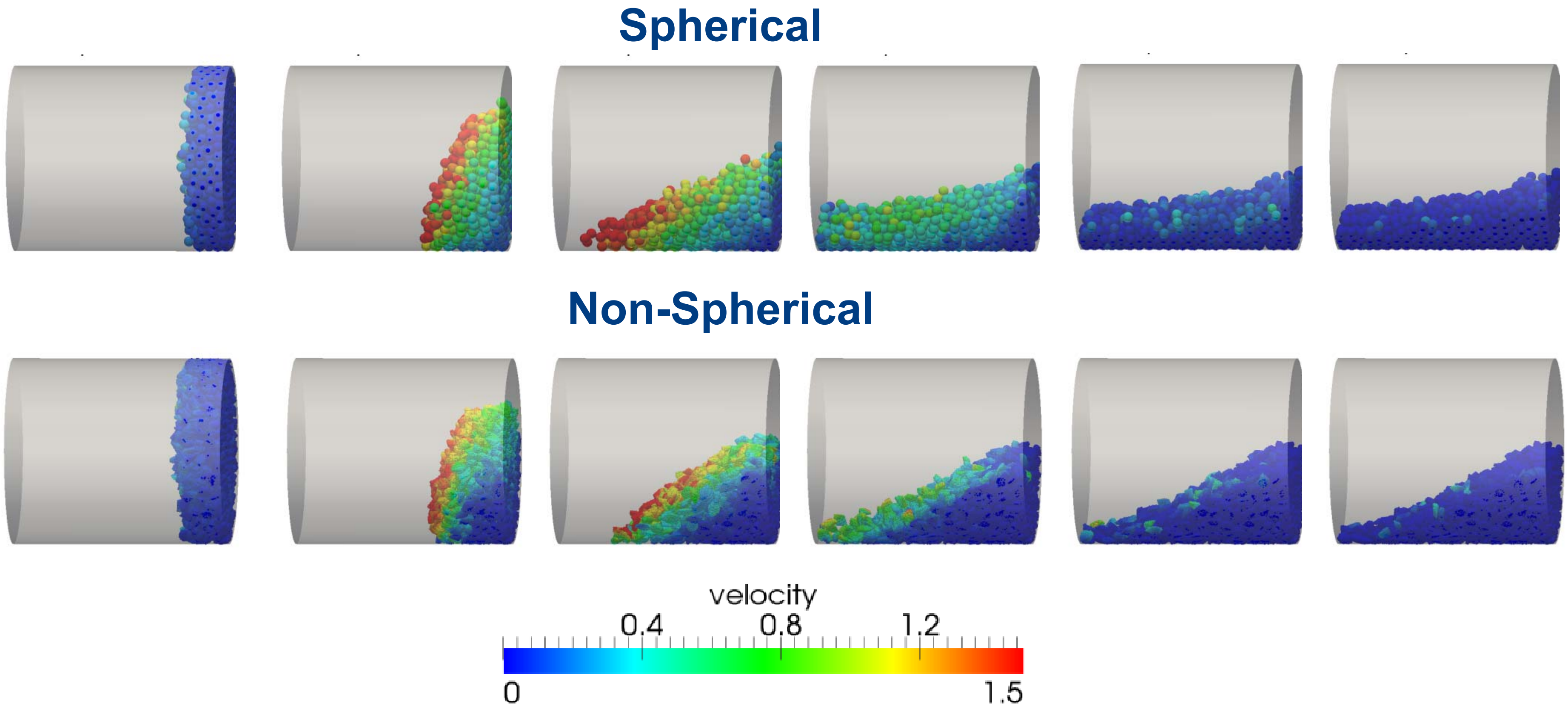
- **Step 2 – Generate Multi-Sphere Simulation Model**



LIGGGHTS Models
Capturing non-sphericity

- **Step 3 – LIGGGHTS Simulation**

Show case cylinder: Gravity tilted so that angle of repose forms



LIGGGHTS Models
Capturing non-uniformity

In reality, no particle is like any other...

- **Fix particledistribution/discrete** takes particletemplates as input and serves as input for insertion commands
- **Fix particletemplate/sphere** and **fix particletamplate/multisphere**

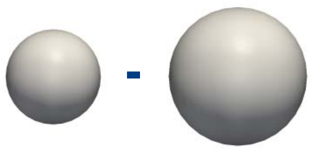
fix particledistribution/discrete

FixParticletemplate **list;

list[0] → **fix particletemplate/sphere**

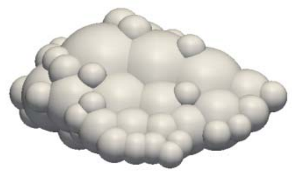
50 mass%
ρ = 2000 kg/m³

list[1] → **fix particletemplate/sphere**



uniform size distribution
15 mass%
ρ = 1500 kg/m³

list[2] → **fix particletemplate/multisphere**

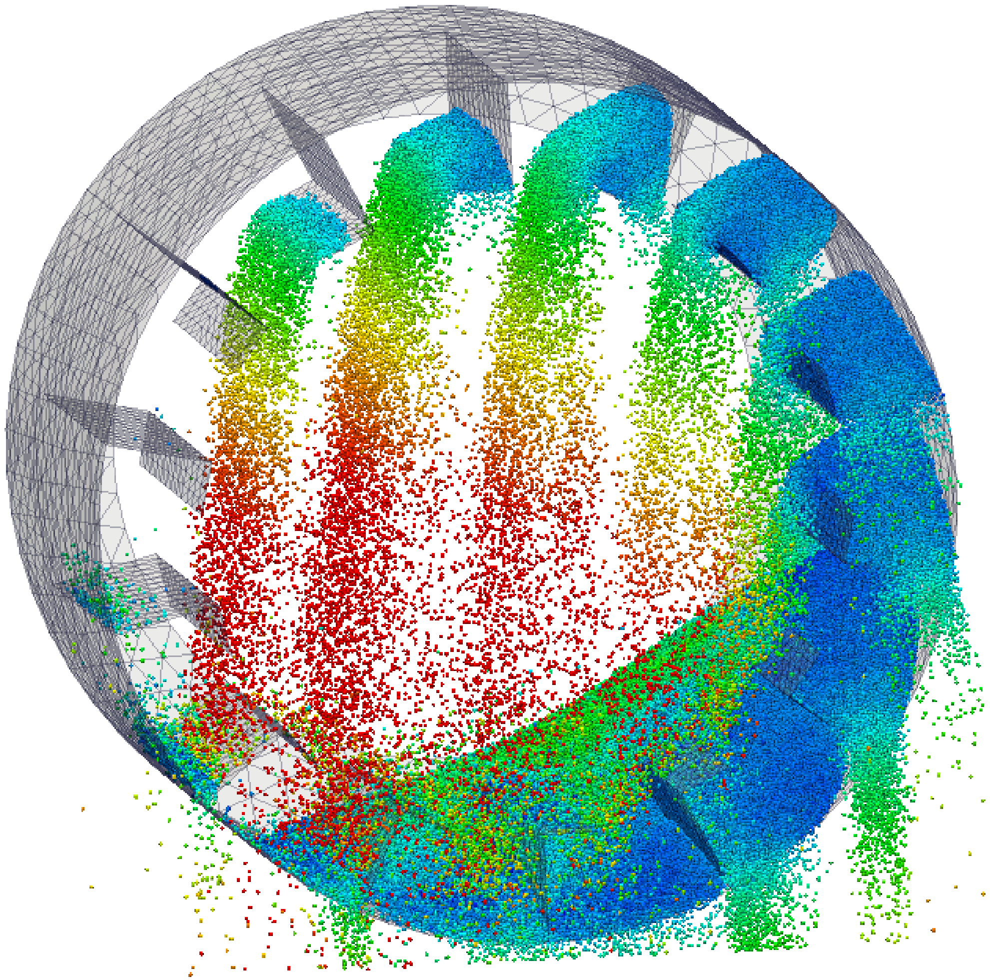


35 mass%
ρ = 3500 kg/m³

III. CAD Interoperability

CAD Interoperability
Importing a Mesh

Industrial scale granular problems need CAD/Mesh interface!



Rotary dryer, ~1M particles

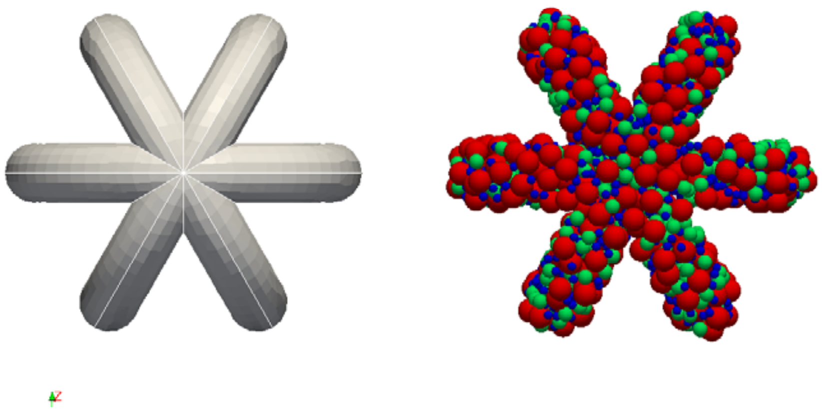
Work with Diego Peinado (Intrame Madrid)

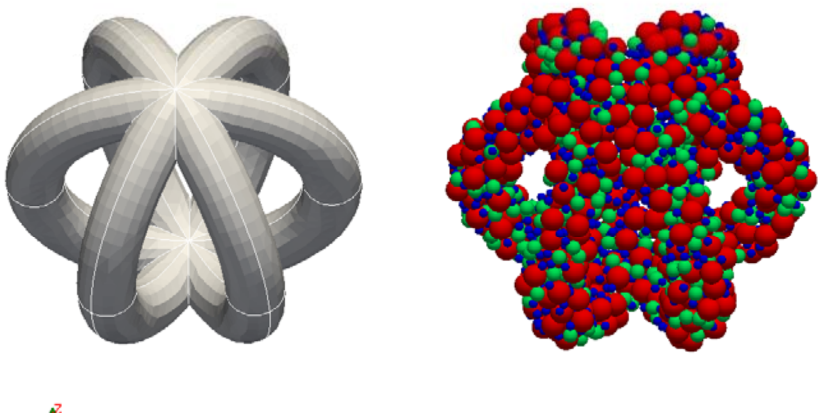
Command fix mesh/gran reads a triangular mesh from STL or VTK

Command fix move/mesh/gran translates/rotates etc the mesh (like fix move)

INTRAME

CAD Interoperability
Generating a Non-trivial Particle Packing



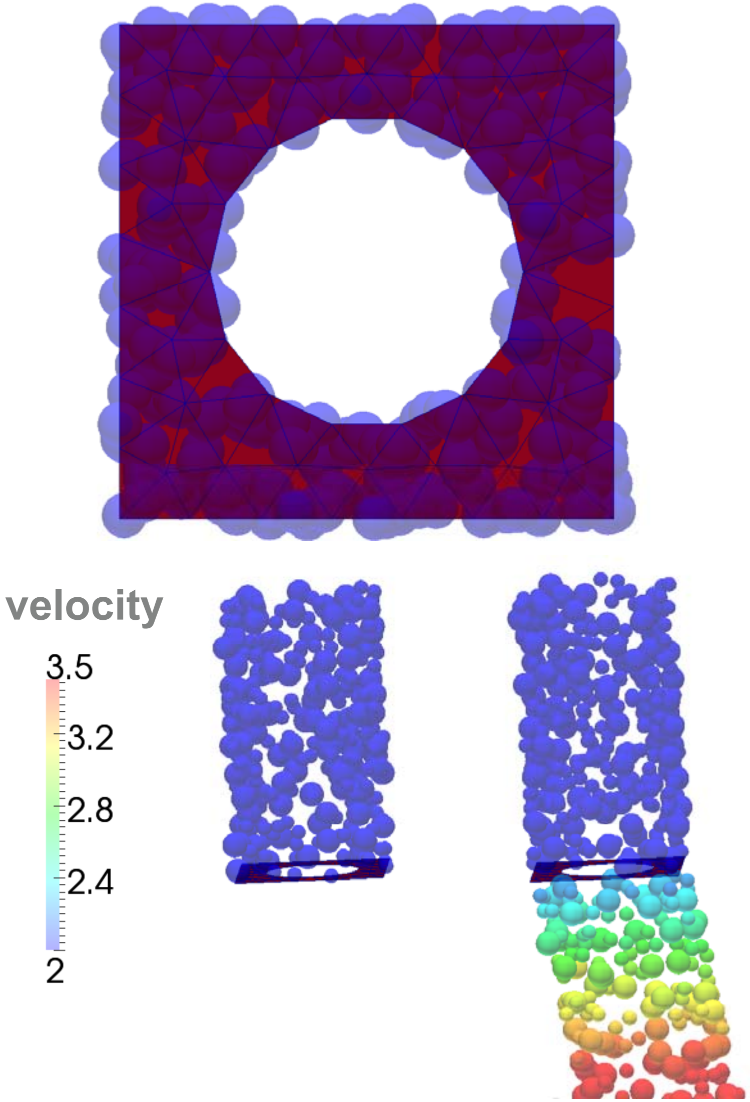


Packing in tetrahedral mesh

Command region tetmesh
Reads tet mesh from VTK to be used as a region

Command fix insert/pack
Works similar to fix pour
Can use arbitrary regions (also region tetmesh)

CAD Interoperability
Generating a Particle Stream



Particle insertion at surface

Command fix insert/stream

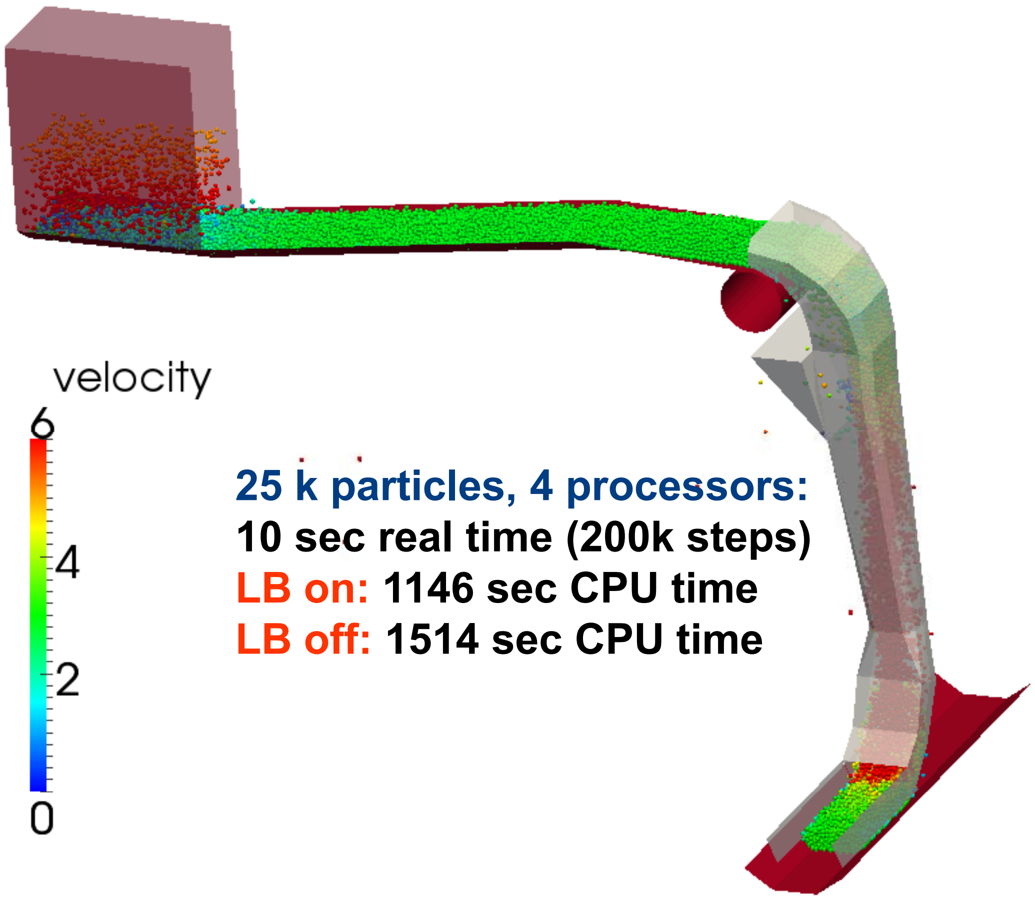
Extrudes triangular surface mesh

Particles are inserted in packages like with fix pour and are integrated by the fix with constant vel until they reach the surface mesh. BCs are enforced at the surface mesh.

Industrial Application
Wear Prediction at Transfer Chute

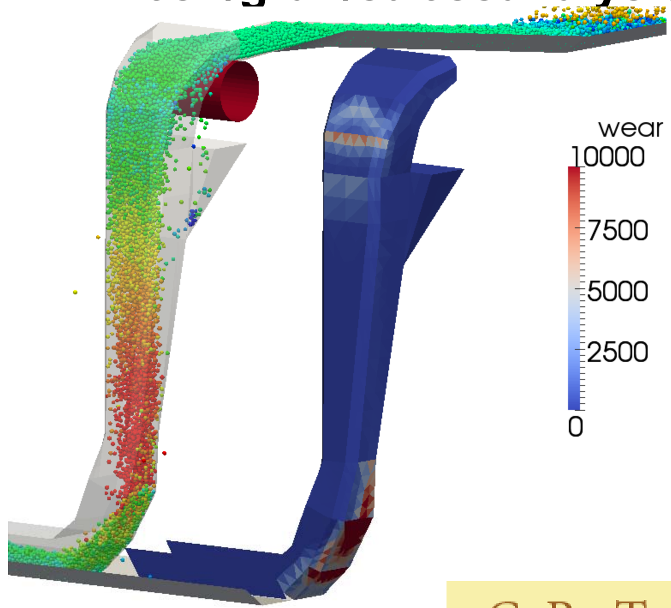
Work with Andre Katterfeld (Univ. Magdeburg, Cepartec)

Goniva, C, Katterfeld, A, Kloss, C: „Simulation of dust emission and transport and chute wear“
Proc. of 16. Fachtagung Schüttgutfördertechnik Magdeburg, Sept 2011



25 k particles, 4 processors:
10 sec real time (200k steps)
LB on: 1146 sec CPU time
LB off: 1514 sec CPU time

Wear prediction (Finnie model)
Fix mesh/gran/stressanalysis



OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

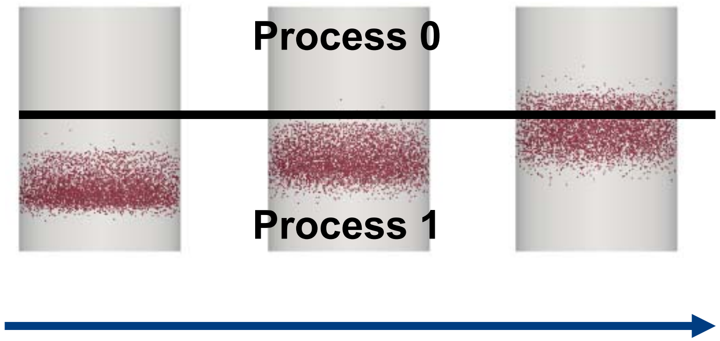
CeParTec GmbH Centre of Particle Technology

IV. Features and Coding Concepts

Dynamic Load-Balancing
What is Load-Balancing

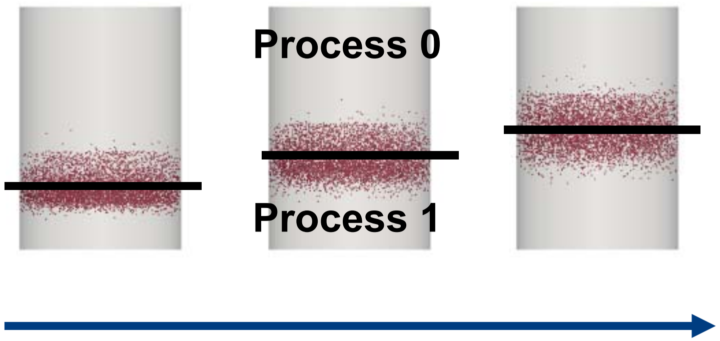
How to distribute load between processors?

Without dynamic load balancing:



advancing simulation time

With dynamic load balancing:

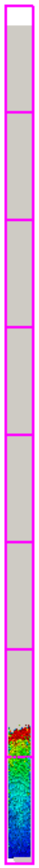


advancing simulation time

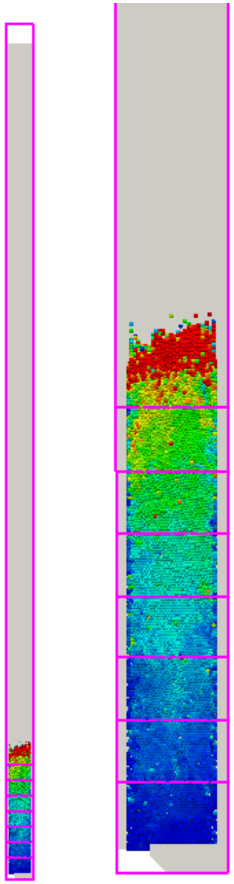
Dynamic Load-Balancing
Silo Test Case

Strategy: “Simple” LB based on number of owned atoms

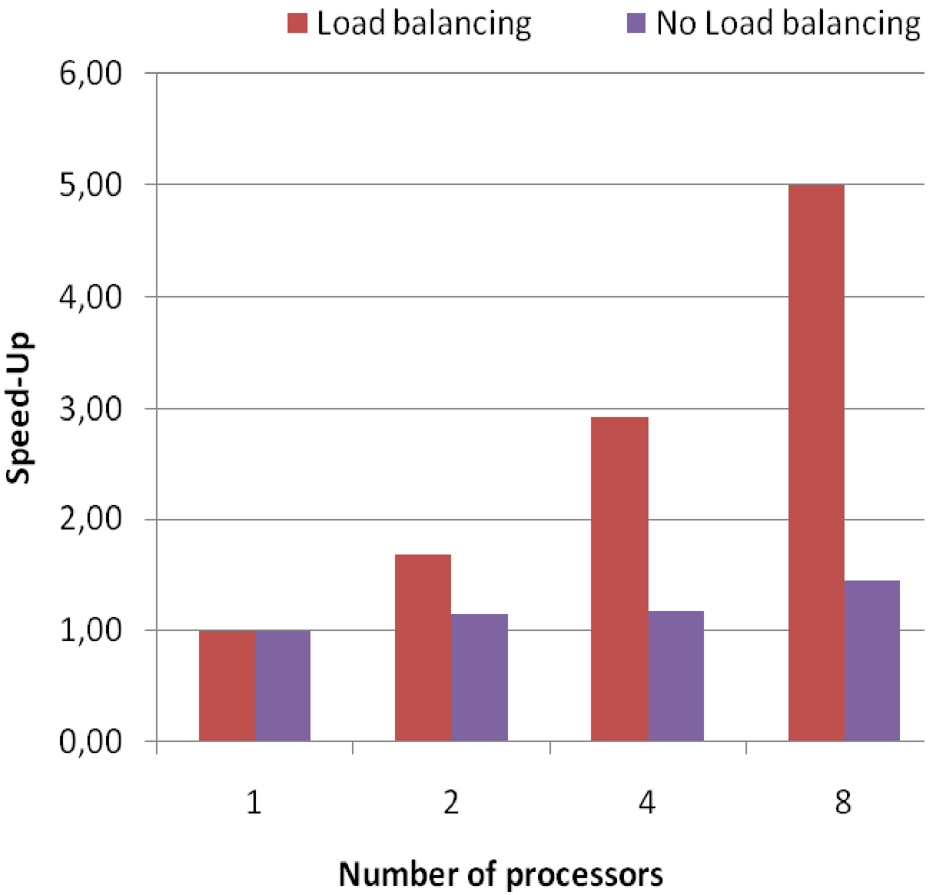
LB off



LB on



Speed-up



Coding Concepts
Strengths and Weak Points for LAMMPS

Strengths of coding in LAMMPS

- Code easy to read and understand
- Code easy to modify
- Good encapsulation for many modelling tasks
  (fixes, computes, pair styles)

But:

- Have to write lots of lo-level code (error prone), some of which is redundant

Abstraction Layer I: Per-Particle Properties
Motivation

We need a lot of different per-particle properties for our models

- Temperature, heat flux, heat source (e.g. from chemical reactions)
- Moisture content (water in pores), liquid surface film height
- Particle Reynolds number
- Drag force, Magnus force, Saffman force,... exerted by surrounding fluid
- Surface dust content, dust emitted to fluid
- Energy stored in elastic deformation, energy dissipated by contacts...
- ....

→ **Need dozens of new per-particle properties for our models**

Abstraction Layer I: Per-Particle Properties
Motivation

Why not store new properties in Atom class?

- **Error prone:** Have to implement low-level code via a new AtomVec class (pack/unpack, interproc exchange, restart...)
- Have to **change many classes** to make new property fully functional (at least Atom, AtomVec, Dump, Set, ComputePropertyAtom, FixAdapt)
- This is a **recurring task**, so we can add an abstraction layer

Solution: Store it in a new fix property/peratom class

- Implement **low-level code only once**
- **Look-up mechanism** via Modify class so each other class can use it
- Every **model (fix) can now request** to store a new **per-particle property**
  Lazy storage: only store a property if any of the models really need it
- **Need to touch only one class for new property**
  (dump, set... implemented only once for fix property/peratom)

Abstraction Layer I: Per-Particle Properties
Usage

```
//register dragforce
  if(!dragforce)
  {
        char* fixarg[11];
        fixarg[0]="dragforce";
        fixarg[1]="all";
        fixarg[2]="property/peratom";
        fixarg[3]="dragforce";
        fixarg[4]="vector";
        fixarg[5]="no";
        fixarg[6]="yes";
        fixarg[7]="no";
        fixarg[8]="0.";
        fixarg[9]="0.";
        fixarg[10]="0.";
        dragforce = modify->add_fix_property_peratom(11,fixarg);
  }
```

Type of fix

Abstraction Layer I: Per-Particle Properties
Usage

```
//register dragforce
  if(!dragforce)
  {
        char* fixarg[11];
        fixarg[0]="dragforce";
        fixarg[1]="all";
        fixarg[2]="property/peratom";
        fixarg[3]="dragforce";
        fixarg[4]="vector";
        fixarg[5]="no";
        fixarg[6]="yes";
        fixarg[7]="no";
        fixarg[8]="0.";
        fixarg[9]="0.";
        fixarg[10]="0.";
        dragforce = modify->add_fix_property_peratom(11,fixarg);
  }
```

Name of property

Abstraction Layer I: Per-Particle Properties
Usage

```
//register dragforce
  if(!dragforce)
  {
        char* fixarg[11];
        fixarg[0]="dragforce";
        fixarg[1]="all";
        fixarg[2]="property/peratom";
        fixarg[3]="dragforce";
        fixarg[4]="vector";
        fixarg[5]="no";
        fixarg[6]="yes";
        fixarg[7]="no";
        fixarg[8]="0.";
        fixarg[9]="0.";
        fixarg[10]="0.";
        dragforce = modify->add_fix_property_peratom(11,fixarg);
  }
```

Type and size of data

Abstraction Layer I: Per-Particle Properties
Usage

```
//register dragforce
  if(!dragforce)
  {
        char* fixarg[11];
        fixarg[0]="dragforce";
        fixarg[1]="all";
        fixarg[2]="property/peratom";
        fixarg[3]="dragforce";
        fixarg[4]="vector";
        fixarg[5]="no";
        fixarg[6]="yes";
        fixarg[7]="no";
        fixarg[8]="0.";
        fixarg[9]="0.";
        fixarg[10]="0.";
        dragforce = modify->add_fix_property_peratom(11,fixarg);
  }
```

Settings for forward comm, reverse comm, restart

Abstraction Layer II: “Transport Equations”
Motivation

There is another recurring task: Solving an ODE for each particle for a quantity, based on a flux and a source

E.g. solve for **heat transfer, moisture, evolution of surface dust...**

$$m_p c_p \frac{dT_{p,i}}{dt} = \underbrace{\sum_{contacts\ i-j} \dot{Q}_{pi-pj}}_{\substack{heat\ conduction \\ by\ contacts}} + \underbrace{\dot{Q}_{pi,source}}_{\substack{heat\ generation \\ due\ to\ sources, \\ e.g.\ reactions}}$$

$$\dot{Q}_{pi-pj} = \frac{4 k_{pi} k_{pj}}{k_{pi} + k_{pj}} \left(A_{contact,i-j}\right)^{1/2} \Delta T_{pi-pj}$$

Solution: A new **fix transportequation/scalar**

- Every **model can request** a scalar transport equation to be solved
- The transportequation makes use of multiple fixes of type property/peratom

Abstraction Layer II: “Transport Equations”
Motivation

There is another recurring task: Solving an ODE for each particle for a quantity, based on a flux and a source

E.g. solve for **heat transfer, moisture, evolution of surface dust...**

$$m_p c_p \frac{dT_{p,i}}{dt} = \underbrace{\sum_{contacts\ i-j} \dot{Q}_{pi-pj}}_{\text{heat conduction by contacts}} + \underbrace{\dot{Q}_{pi,source}}_{\text{heat generation due to sources, e.g. reactions}}$$

$$\dot{Q}_{pi-pj} = \frac{4 k_{pi} k_{pj}}{k_{pi} + k_{pj}} \left(A_{contact,i-j}\right)^{1/2} \Delta T_{pi-pj}$$

Solution: A new **fix transportequation/scalar**

- Every **model can request** a scalar transport equation to be solved
- The transportequation makes use of multiple fixes of type property/peratom

V. Coupling to Continuum Methods

Coupling to Continuum Methods
Motivation

Many problems are better modeled using a continuum approach

Our main interest lies in **fluid mechanics**

Weapon of choice: OpenFOAM®, www.openfoam.com

Open Field Operation and Manipulation

Leader in OpenSource CFD, market share Germany ~30%, US ??

OpenFOAM

- Initiated by **Henry Weller** and **Hrvoje Jasak** at **Imperial** in the 90s
- **Constant development by universities and companies**,
  ~ 2.000.000 LOC according to Jasak (end of 2009)
- Capable of doing **Finite Volume, Finite Element, Particle Methods (Lagrangian tracking, DSMC, MD), Electromagnetics, Financial**

Coupling to Continuum Methods
How does OpenFOAM (R) work? – PISO Algorithm

Pressure-Implicit Split-Operator (PISO) Algorithm

```
for (runTime++; !runTime.end(); runTime++) // use the runTime object to control time stepping
{
  //linear momentum equation. The flux of U, phi, is treated explicity
  fvVectorMatrix UEqn( fvm::ddt(U) + fvm::div(phi, U) - fvm::laplacian(nu, U) );
  solve(UEqn == -fvc::grad(p)); // solve using the last known value of p, → U approx. satisfies momentum

  for (int corr=0; corr<nCorr; corr++) // --- PISO loop---- take nCorr corrector steps
  {
    volScalarField rUA = 1.0/UEqn.A();
    U = rUA*UEqn.H();
    phi = (fvc::interpolate(U) & mesh.Sf())

    for (int nonOrth=0; nonOrth<=nNonOrthCorr; nonOrth++)
    {
      fvScalarMatrix pEqn ( fvm::laplacian(rUA, p) == fvc::div(phi)  ); // set up the pressure equation
      pEqn.solve();

      if (nonOrth == nNonOrthCorr) phi -= pEqn.flux(); // on last non-orth. corr., correct flux using new p
    }  // end of non-orthogonality looping
    U -= rUA*fvc::grad(p);
    U.correctBoundaryConditions();
  } // end of the PISO loop
} // end of the time step loop
```

momenum predictor

$$\boxed{\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) \;=\; -\nabla p + \nabla \cdot (\nu \nabla \mathbf{u}) + q_{\mathbf{u}}.}$$

Coupling to Continuum Methods
How does OpenFOAM (R) work?

```
fvVectorMatrix UEqn
(
        fvm::ddt(U) + fvm::div(phi, U) - fvm::laplacian(nu, U)
);
solve
(
        UEqn == -fvc::grad(p)
);
```

=

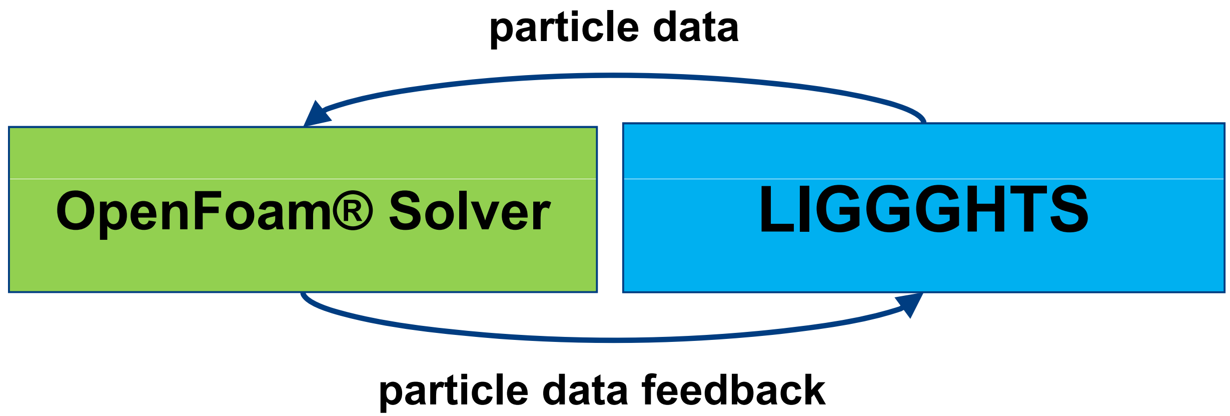
$$\boxed{\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) \quad = \quad -\nabla p + \nabla \cdot (\nu \nabla \mathbf{u})}$$

OpenFOAM® is using the paradigm of object-oriented programming to make code to solve PDEs easily readable

Coupling to Continuum Methods
Motivation

Motivation for using OpenFOAM® is manifold:

- for **postprocessing** of DEM/MD simulations
- for simulation of **coupled fluid-granular flow (FV method)**
- can use the full model portfolio of OF (**FE, electrostatics,**...)

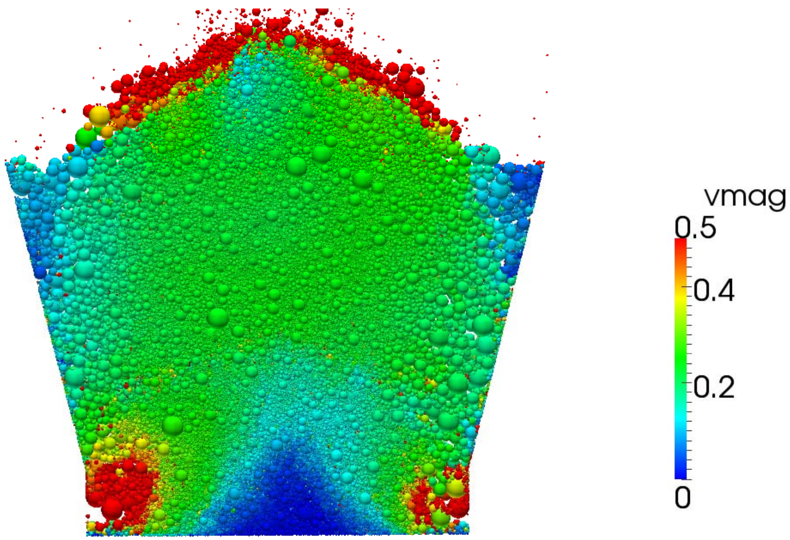


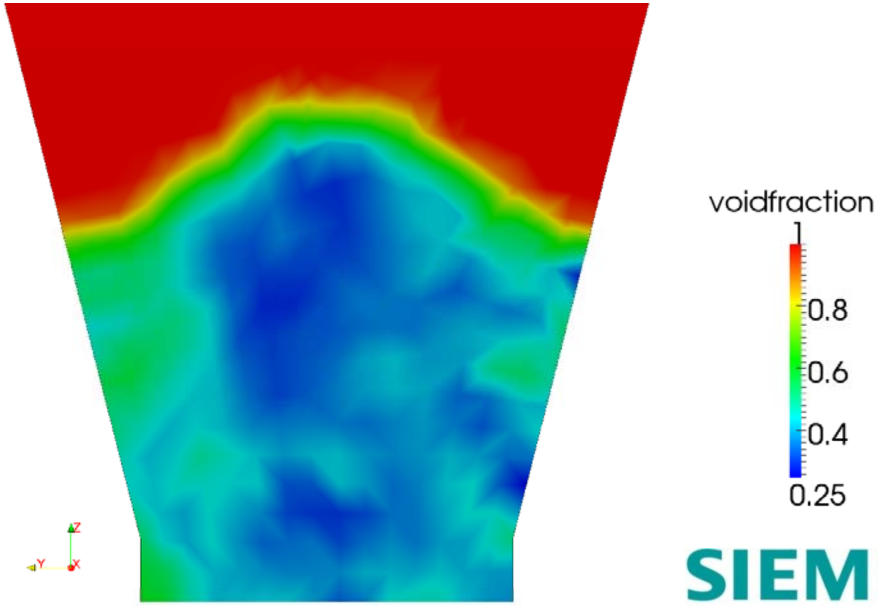
More information and downloads: www.cfdem.com

Coupling to Continuum Methods
Motivation

Post-processing of DEM/MD data

- Able to **transfer arbitrary particle properties to OpenFOAM fields**
- Can use arbitrary **unstructured meshes**
- Example: Evaluating voidage, which is an important process parameter in multiphase flow reactors

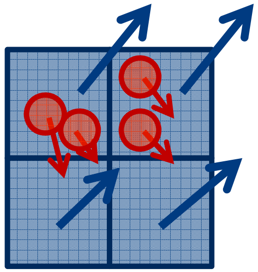




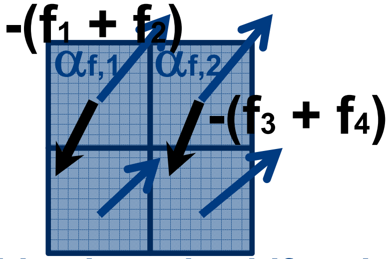
Coupling to Continuum Methods
Motivation

Simulation of Coupled Fluid-Granular Flow: CFD-DEM Method

- Able to **transfer arbitrary particle properties to OpenFOAM fields**
- Can use arbitrary **unstructured meshes**
- Able to perform **resolved CFD-DEM** and **unresolved CFD-DEM**

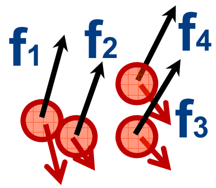


=

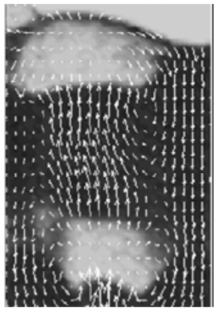


CFD with adapted voidfraction and momentum sources

+



DEM with force coming from CFD



=

+

Coupling to Continuum Methods
Unresolved CFD-DEM

Theoretical background:

Navier-Stokes equations for the fluid in presence of a granular phase

$$\frac{\partial \alpha_f \rho_f}{\partial t} + \nabla \cdot \left(\alpha_f \rho_f \mathbf{u_f} \right) = 0$$

$$\frac{\partial \left(\alpha_f \rho_f \mathbf{u_f} \right)}{\partial t} + \nabla \cdot \left(\alpha_f \rho_f \mathbf{u_f} \mathbf{u_f} \right) = -\alpha_f \nabla p - \mathrm{K_{fs}} \left(\mathbf{u_f} - \mathbf{u_s} \right) + \nabla \cdot \left(\alpha_f \boldsymbol{\tau} \right) + \alpha_f \rho_f \mathbf{g}$$

α_f	fluid volume fraction
u_f	fluid velocity
τ, p	stress tensor, pressure
ρ_f	fluid density
K_{fs}	fluid solid momentum exchange term
	comprises drag force, Magnus and Saffman force, virtual mass force,...

Coupling to Continuum Methods
Unresolved CFD-DEM

Theoretical background:

Example for fluid solid momentum exchange K_{fs} :

Di Felice (1994): “The voidage function for fluid-particle interaction systems.” Int. J. of Multiphase Flow, Vol. 20, p153-159

$$\left.\begin{aligned}
\mathbf{F_d} &= \frac{1}{2}\rho_f\left(\mathbf{u_f}-\mathbf{u_p}\right)\left|\mathbf{u_f}-\mathbf{u_p}\right| C_d \frac{d_p^{\,2}\pi}{4}\alpha_f^{\,1-\chi}, \\
C_d &= \left(0.63+\frac{4.8}{\mathrm{Re}_p}\right)^2, \\
\chi &= 3.7-0.65\exp\left[-\frac{\left(1.5-\log_{10}\mathrm{Re}_p\right)^2}{2}\right]
\end{aligned}\right\}\quad \mathrm{K_{fs}} = \frac{\alpha_f\cdot\left|\sum_i \mathbf{F_d}\right|}{V_{cell}\cdot\left|\mathbf{u_f}-\mathbf{u_p}\right|}$$

Better models available are based on Lattice Boltzmann simulation (e.g. Koch and Hill)

Application Example
Soil Sampling with “Mole” / Work with DLR Bremen

physics

goal:
-penetration of heated body into frozen soil

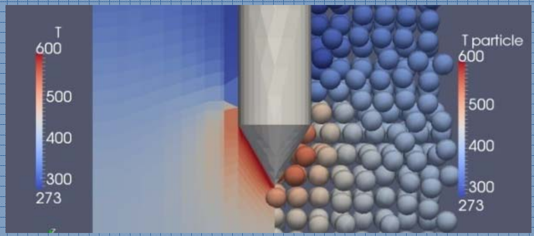
phenomena:
-heat transfer from body to soil

-soil mechanics

-phase change due to heating

-body motion

models



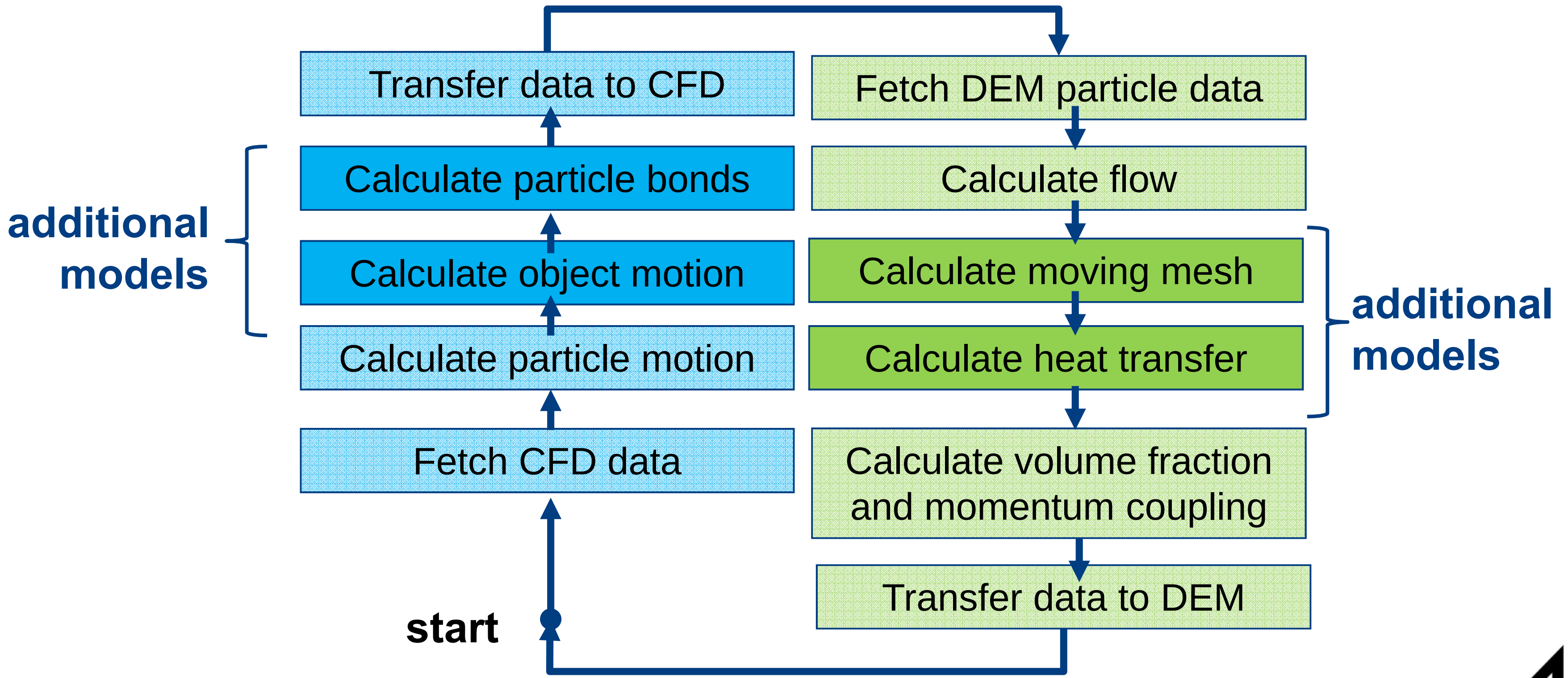
modeling:
-heat transfer by finite volume (continuum) approach

-DEM approach for soil modelling

- temperature triggered particle bonds

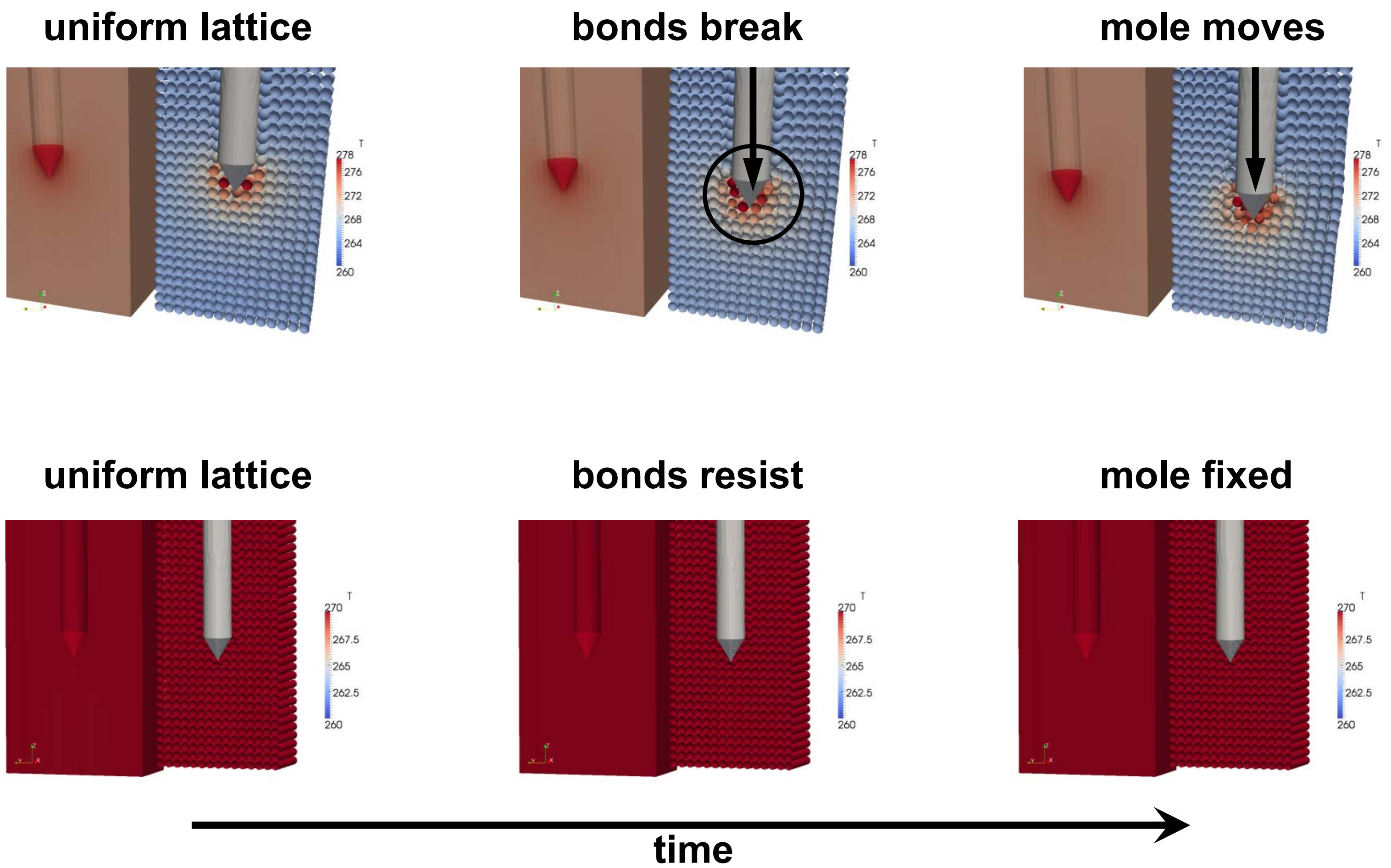
- coupled 6 DOF and moving mesh

Application Example
Soil Sampling with “Mole”



Application Example
Soil Sampling with "Mole"

moving mesh application: coupling of Temperature and object motion



heated mole

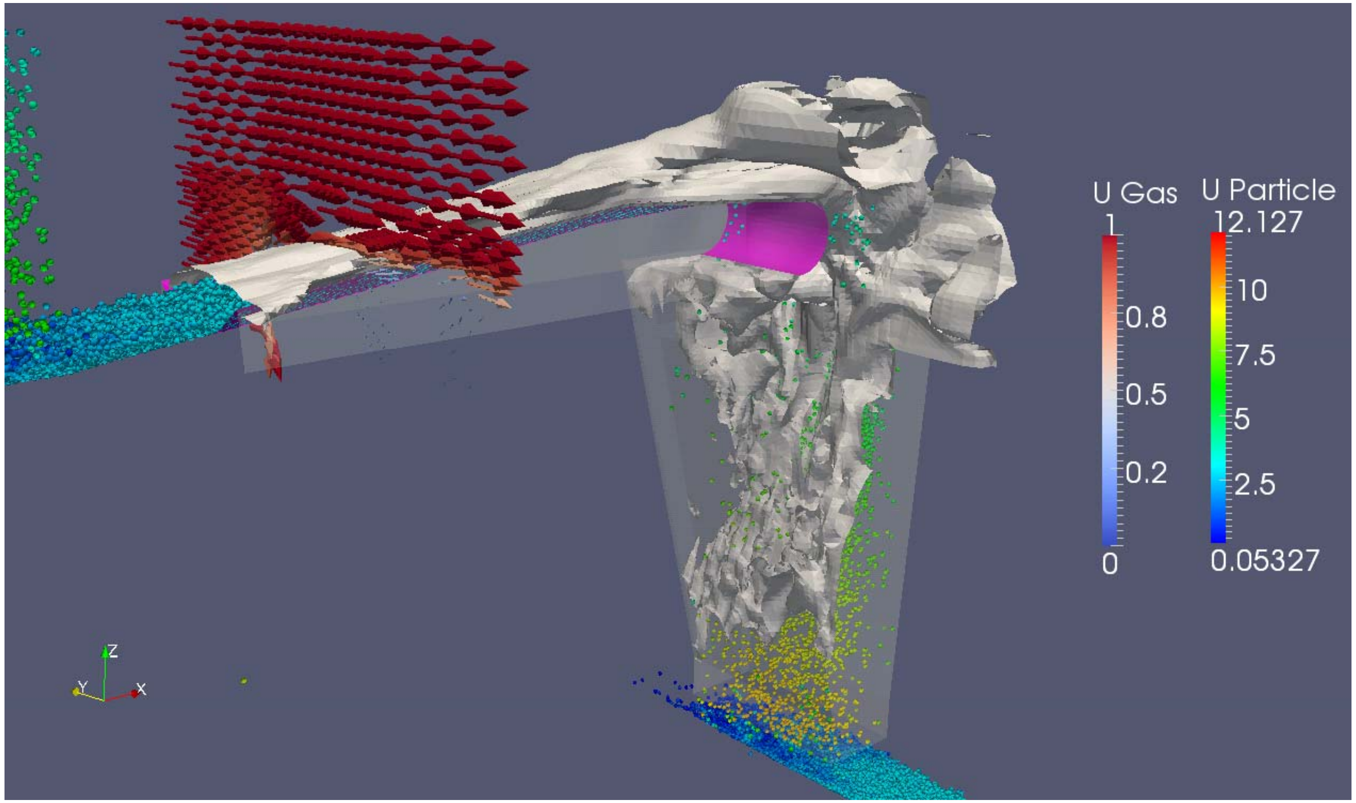
cold mole

Industrial Application
Reduction of Dust Emission in a Transfer Chute

Work with Andre Katterfeld (Univ. Magdeburg, Cepartec)

Goniva, C, Katterfeld, A, Kloss, C: „Simulation of dust emission and transport and chute wear"
Proc. of 16. Fachtagung Schüttgutfördertechnik Magdeburg, Sept 2011

Original Geometry

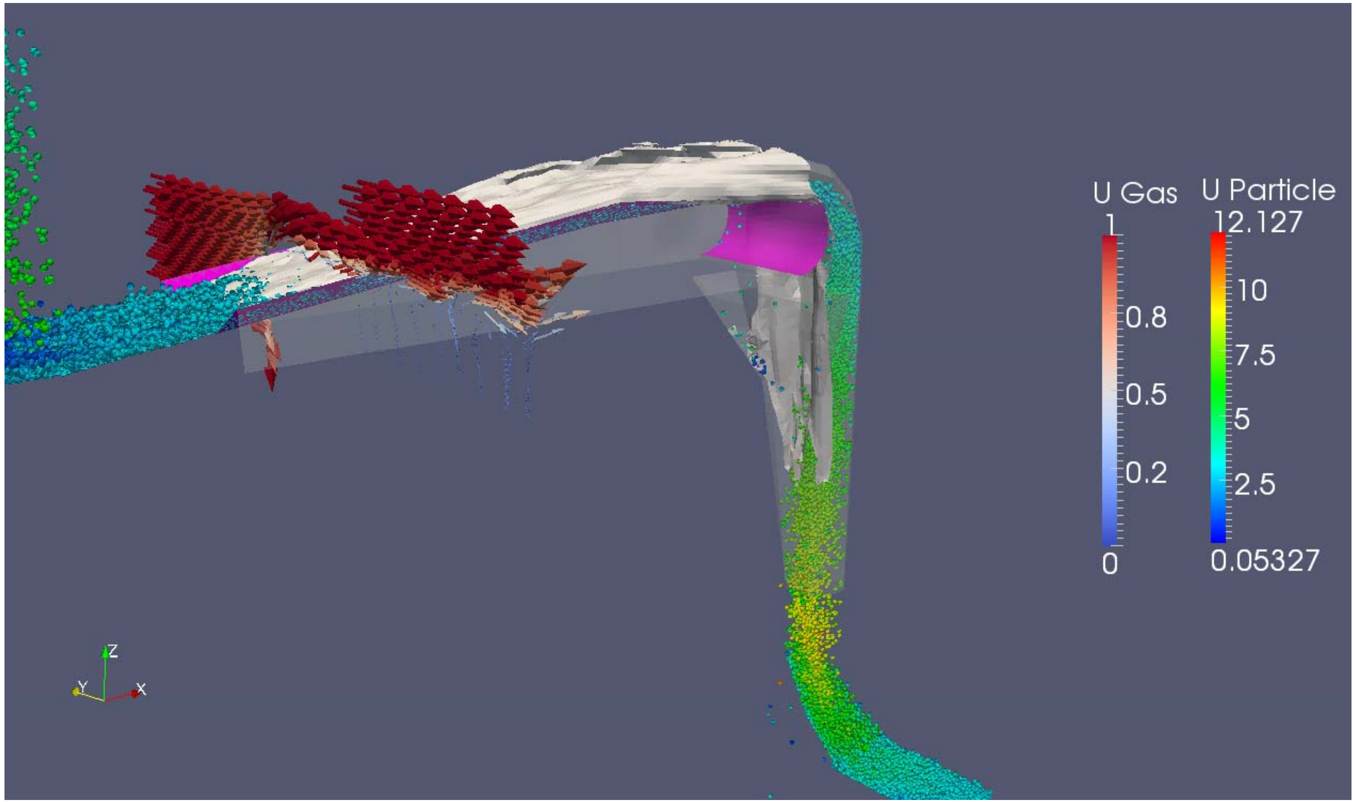


Industrial Application
Reduction of Dust Emission in a Transfer Chute

Work with Andre Katterfeld (Univ. Magdeburg, Cepartec)

Goniva, C, Katterfeld, A, Kloss, C: „Simulation of dust emission and transport and chute wear“
Proc. of 16. Fachtagung Schüttgutfördertechnik Magdeburg, Sept 2011

Optimized Geometry



Thank you for your attention!
Questions?

www.cfdem.com | www.particulate-flow.at