BOOKLET OF PRESENTATIONS

1st Yade Workshop

July 7th-9th, 2014, Grenoble (Campus SMH)

Organizers: Bruno Chareyre, Caroline Chalak, 3SR lab.

Forewords

Particle-scale modeling remains an area of active developments, decades after the pioneering work of P. Cundall. A large part of these developments is mirrored by contributions to the open source platform Yade-DEM. The objective of this workshop was to gather people interested in DEM and DEM-related developments, with special focus on new models and couplings, algorithmic issues, performance and parallelization.

It has been our great pleasure to host this event, which enabled not only presentations of recent advances in DEM modeling but also the sharing of experiences in DEM and fruitful opportunities of socialization.

We hope that this YADE workshop is the first of a long series.

Bruno Chareyre, Caroline Chalak

The page number of talks in the present booklet is indicated below next to the title

If you need to cite some of the talks, here is an exemple:
Gladky A., 2014, Applications of DEM at IMFD Freiberg, in booklet of the 1st YADE Workshop, ed. B. Chareyre & C. Chalak, Grenoble, pp. 14-29.

July 7th

Morning
11:00 – 13:30 Icebreaker + lunch

Afternoon
13:30 – 15:30 **DEM applications I**

Introduction (p.7)
Bruno Chareyre (Université Grenoble Alpes / 3SR)

Applications of DEM at IMFD Freiberg (p.14)
Anton Gladky (TU Freiberg)

Multi-scale FEMxDEM modelling of cohesive-frictional granular materials (p.30)
Jacques Desrues (CNRS, Université Grenoble Alpes / 3SR)

DEM-FEM Couplings (p.48)
Jan Stránský (TU Prague)

Rockfall protective systems (p.90)
Klaus Thoeni (Univ. Newcastle AU)

15:30 – 16:00 Break

16:00 – 18:00 **DEM applications II**

DEM modeling of mass finishing at the IWF (p.123)
Alexander Eulitz (T.U. Berlin / IWF)

Biphasic particles to simulate fresh pervious concrete compaction (p.147)
Ricardo Pieralisi (Univ. Catalunya)

DEM techniques for rock mass and fractures (p.167)
Jérôme Duriez / Frédéric Donzé (Université Grenoble Alpes / 3SR)

Rockfall impacts on trees and other wooden structures
Franck Bourrier (IRSTEA Grenoble)

19:30 - **Workshop dinner in Grenoble**

July 8th

9:00 – 10:30 **Numerical methods and modelling techniques**

Discrete Element modelling of fractured rock (p.183)
Jérôme Duriez (Univ. Grenoble Alpes / 3SR)

Higher order integration schemes
Burak ER (Bursa Technical University, Turkey)

Particles-node separation (in Woo-DEM) (p.196)
Václav Šmilauer (Prague)

Extended use of periodic boundary conditions (p.207)
Jan Stránský (TU Prague)

Inconsistencies of contact laws in DEM (p.266)
Bruno Chareyre (Univ. Grenoble Alpes / 3SR)

10:30 – 10:45 Break

10:45 – 11:10 **Open-Source development**
Versioning, testing, packaging and releasing Yade (p.298)
Anton Gladky (TU Freiberg)
Remi Cailletaud (CNRS, Univ. Grenoble Alpes / 3SR)

11:10 – 12:30 **Complex shapes**

Wave Propagation in Galerkin-based Discrete Element Particle Assemblies
Burak ER (Bursa Technical University, Turkey)

DEM-Membranes (p.333)
Václav Šmilauer (Prague)

Polyhedral particles (p.354)
Jan Eliáš (Brno) / Jan Stransky (TU Prague)

Cylinders, and grids (and more...) (p.381)
François Kneib (IRSTEA Grenoble)
Klaus Thoeni (Univ. Newcastle AU)

12:30 – 14:00 Lunch

14:00 – 15h45 **Multi-phase couplings (part I)**

DEM-LBM coupling (p.395)
Luc Sibille (Univ. Grenoble Alpes / 3SR)

DEM models of the pendular state (p.407)
Christian Jakob (presented by A. Gladky) (TU Freiberg)
Caroline Chalak (Univ. Grenoble Alpes / 3SR)

Coupling DEM and a two-phase flow model for bedload transport (p.414)
Raphaël Maurin (IRSTEA Grenoble)

DEM-SPH coupling
Anton Gladky (TU Freiberg)

DEM-CFD coupling (a few words – for completeness)
Bruno Chareyre

15:45 – 16:15 break

16:15 – 18:00 **Multi-phase couplings (part II)**

The pore-scale (PFV) approach of DEM-fluids couplings (p.422)
Emanuele Catalano (ITASCA France) / Bruno Chareyre (Univ. Grenoble Alpes / 3SR)

PFV + Stokesian Dynamics for flowing fluid-grain mixtures (p.439)
Donia Marzougui (Univ. Grenoble Alpes / 3SR)

Coupling the DEM and a pore-scale model of two-phase flow (p.543)
Chao Yuan (Univ. Grenoble Alpes / 3SR)

Hydrofracturing rocks
Efthymios Papachristos (Univ. Grenoble Alpes / 3SR)

General discussion

July 9th

Yade-DEM project meeting

9:00 – 12:30 Morning session

Talks

Multicore & performance benchmarks (p.462)
Klaus Thoeni (Univ. Newcastle AU)
Alexander Eulitz (TU Berlin)

What happened to the insertion sort collider? (p.502)
Bruno Chareyre (Univ. Grenoble)

DEM on GPU (p.508)
Václav Šmilauer (Prague)

Linking external libraries (practice time)
Anton Gladky (TU Freiberg)

Compilation on MS Windows (p.533)
Václav Šmilauer (Prague)

General discussions:

Performance and Parallelization
Contact laws
The P2PCD project
Survey of DEM softwares
Project management
Editorial strategy
Funding the project and consulting activities
Next Yade workshops?

12:30 – 13:30 Lunch

13:30 – 18:00 **Coding session + brainstormings**

1st YADE WORKSHOP

7-9 July

Grenoble

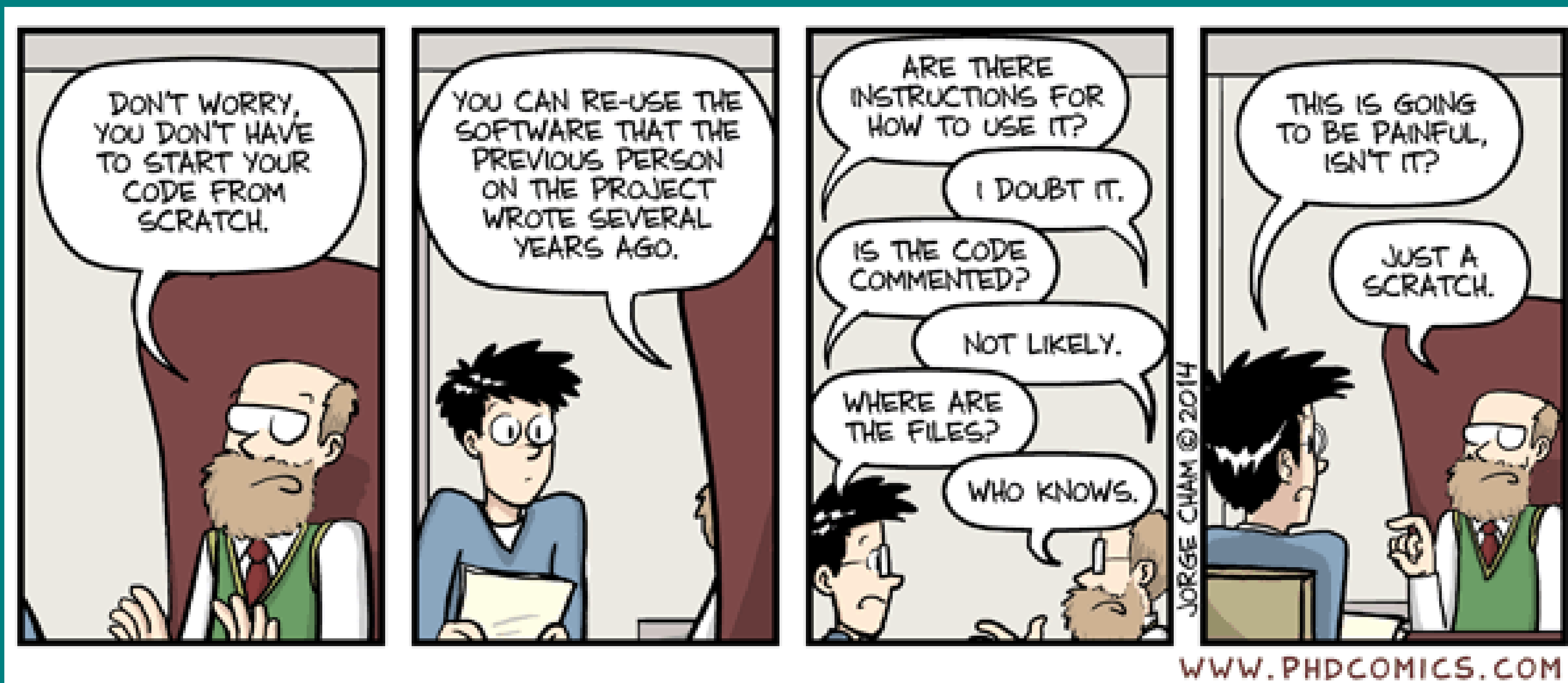
1st YADE WORKSHOP

Computer codes in science (part 1)

(reported by Wolfram Rühak, XX. CMWR, Stuttgart 2014)

1st YADE WORKSHOP

Computer codes in science (part 2)



(reported by Anton Gladky – reaction to the previous one)

1st YADE WORKSHOP

The YADE-DEM Project

Initial idea: F.V. Donzé
Core devs: J. Kozicki, V. Smilauer
Packaging: A. Gladky
Hosting: R. Cailletaud
Other contributors: many

1st YADE WORKSHOP

This workshop

~40 participants, mostly devs. and users of yade-dem

Academic support

MaiMoSiNE

RNVOR

FED

3SR Laboratoire Sols - Solides - Structures - Risques

Grenoble INP

Université Joseph Fourier GRENOBLE

cnrs

Région RhôneAlpes

Some objectives of this workshop

Share:

- Good practices in code development

- Goals

- Ideas and expertise in various fields of science

Special thanks

Caroline Chalak, Mireille de Sousa-Pfister, Chao Yuan

Have a nice workshop and a pleasant stay in Grenoble

TECHNISCHE UNIVERSITÄT
BERGAKADEMIE FREIBERG
The University of Resources. Since 1765.

imfd

YADE 2014

“Applications of DEM at TU Bergakademie Freiberg”

CONTENT

TU Bergakademie Freiberg

Processing machine simulations

Flow process simulation

The University of Resources

1. Situated in the city of Freiberg, Saxony, Germany
2. Founded approximately 250 years ago
3. 6 faculties
4. 5600 Students, 86 Professors
5. 2 collaborative research centers

Source: http://en.wikipedia.org/wiki/Freiberg_University_of_Mining_and_Technology

Processing machine simulations

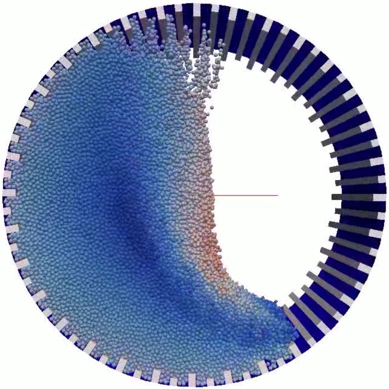


Fig. 1: Ball mill simulation

Parameter study:

1. Geometry
2. Material properties
3. Operating modes

Processing machine simulations

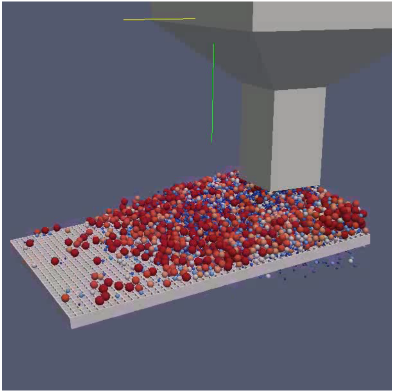


Fig. 2: Screening machine simulation

Parameter study:

1. Geometry
2. Inclination angle
3. Material properties
4. Operating modes
5. Grain size

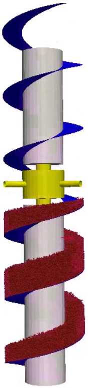


Fig. 3: Screw conveyor

Parameter study:

1. Geometry
2. Inclination angle
3. Material properties
4. Operating modes
5. Grain size

Processing machine simulations

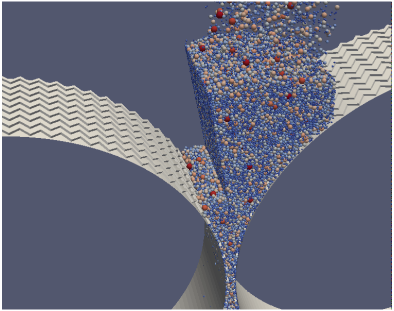


Fig. 4: High pressure grinding roller

Parameter study:

1. Geometry
2. Material properties
3. Operating modes
4. Grain size

Processing machine simulations

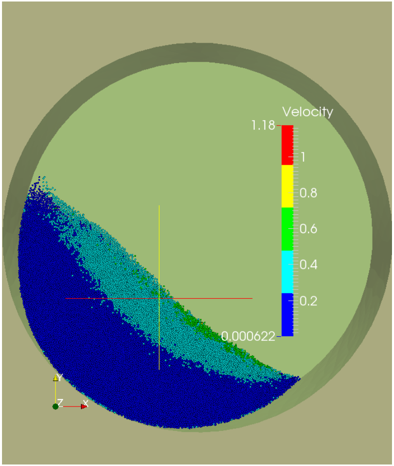


Fig. 5: Pelletizer

Parameter study:

1. Geometry
2. Material properties
3. Operating modes
4. Grain size

Processing machine simulations

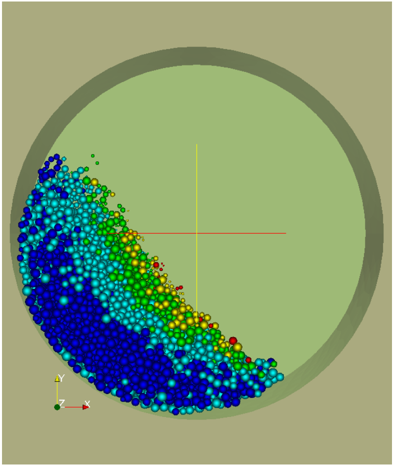


Fig. 6: Pelletizer

Parameter study:

1. Geometry
2. Material properties
3. Operating modes
4. Grain size

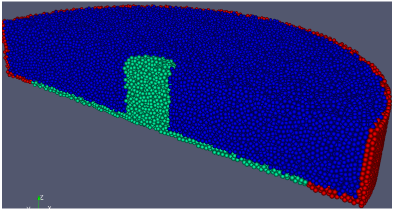


Fig. 7: Split-bottom shear cell

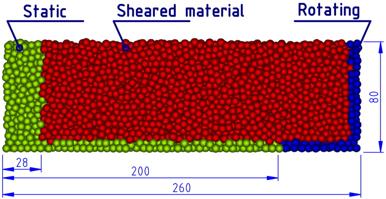


Fig. 8: Shear cell schema

Shear cell simulation

1. Specific split-bottom geometry
2. Obtained shear band, where the sheared material flows and yields
3. Getting different parameters from one simulation (p, τ, $\dot{\gamma}$ etc.)
4. Study of dry and wet (cohesive) material

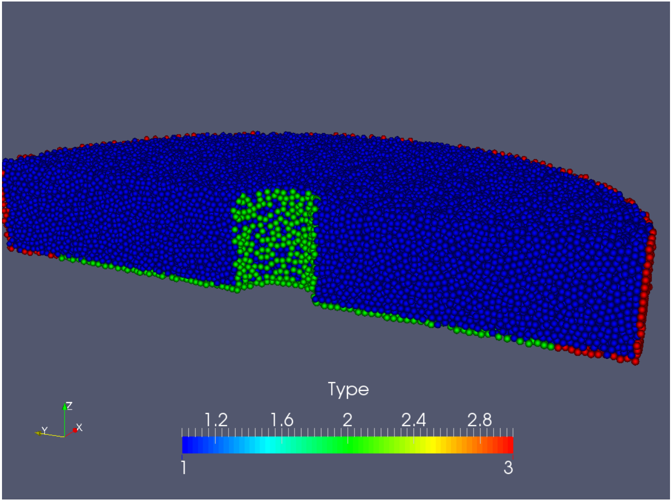


Fig. 9: Split-bottom shear cell. Common view

Flow process simulation

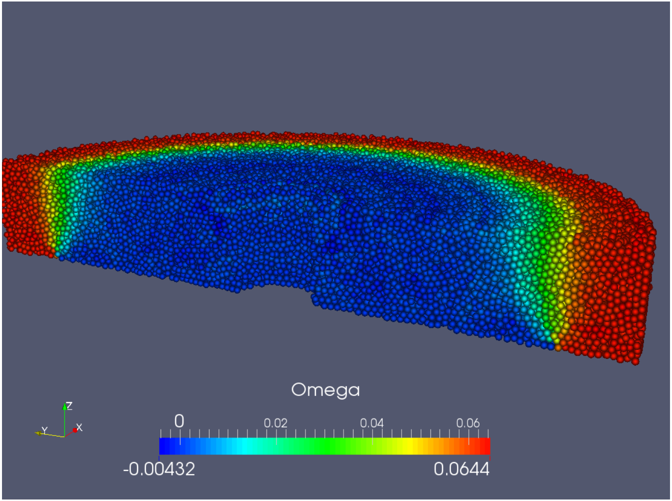


Fig. 10: Split-bottom shear cell. Rotational velocity

Flow process simulation

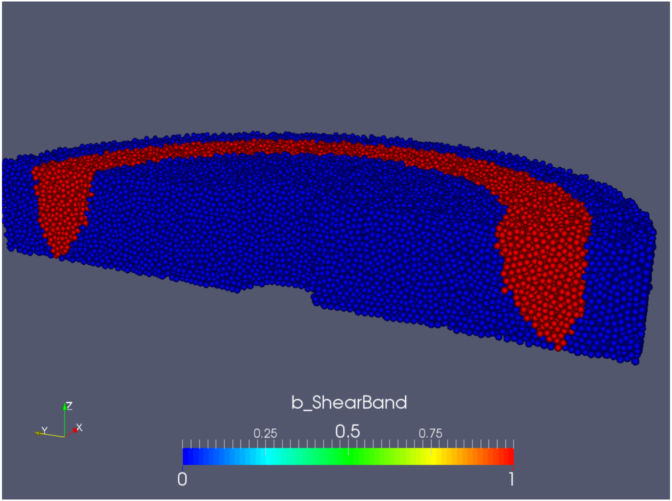


Fig. 11: Split-bottom shear cell. Shear band

Flow process simulation

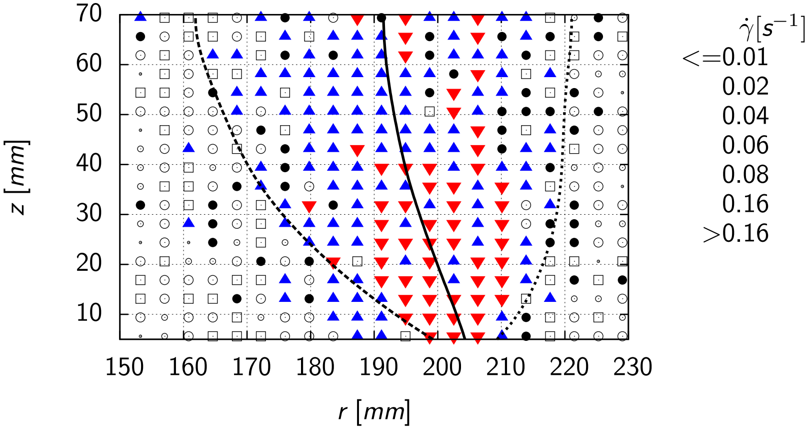


Fig. 12: Strain rate $\dot{\gamma}$ as function of radial and vertical position

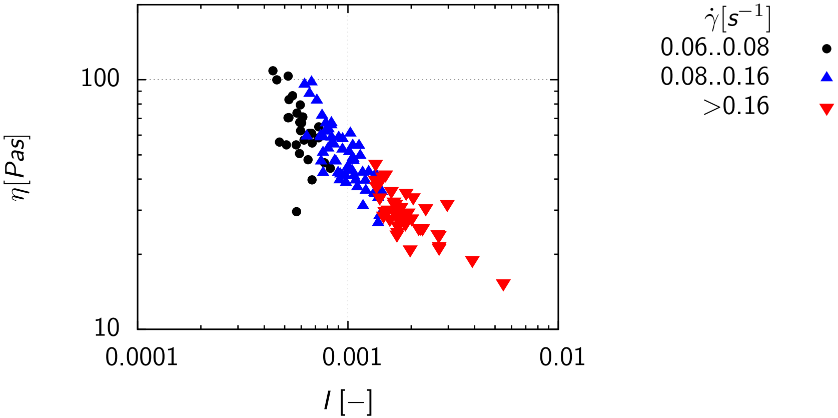


Fig. 13: Local shear viscosity η as a function as inertial number $I = \dot{\gamma} d_p / (p/\rho)$

Thank you for your attention!

Multiscale FEMxDEM modelling of cohesive-frictional granular materials

J. Desrues, T.K. Nguyen, G. Combe, D. Caillerie

Laboratoire Sols, Solides, Structures et Risques (3SR)

CNRS - Université de Grenoble

CNRS UMR 5521

GRENOBLE INP

UJF GRENOBLE I

1st YADE Workshop– Grenoble July2014

Outline

1. Introduction
2. Micro-scale (DEM) Model
3. Multi-scale Coupling Method
4. FEM-DEM simulation
5. Conclusions & Perspectives

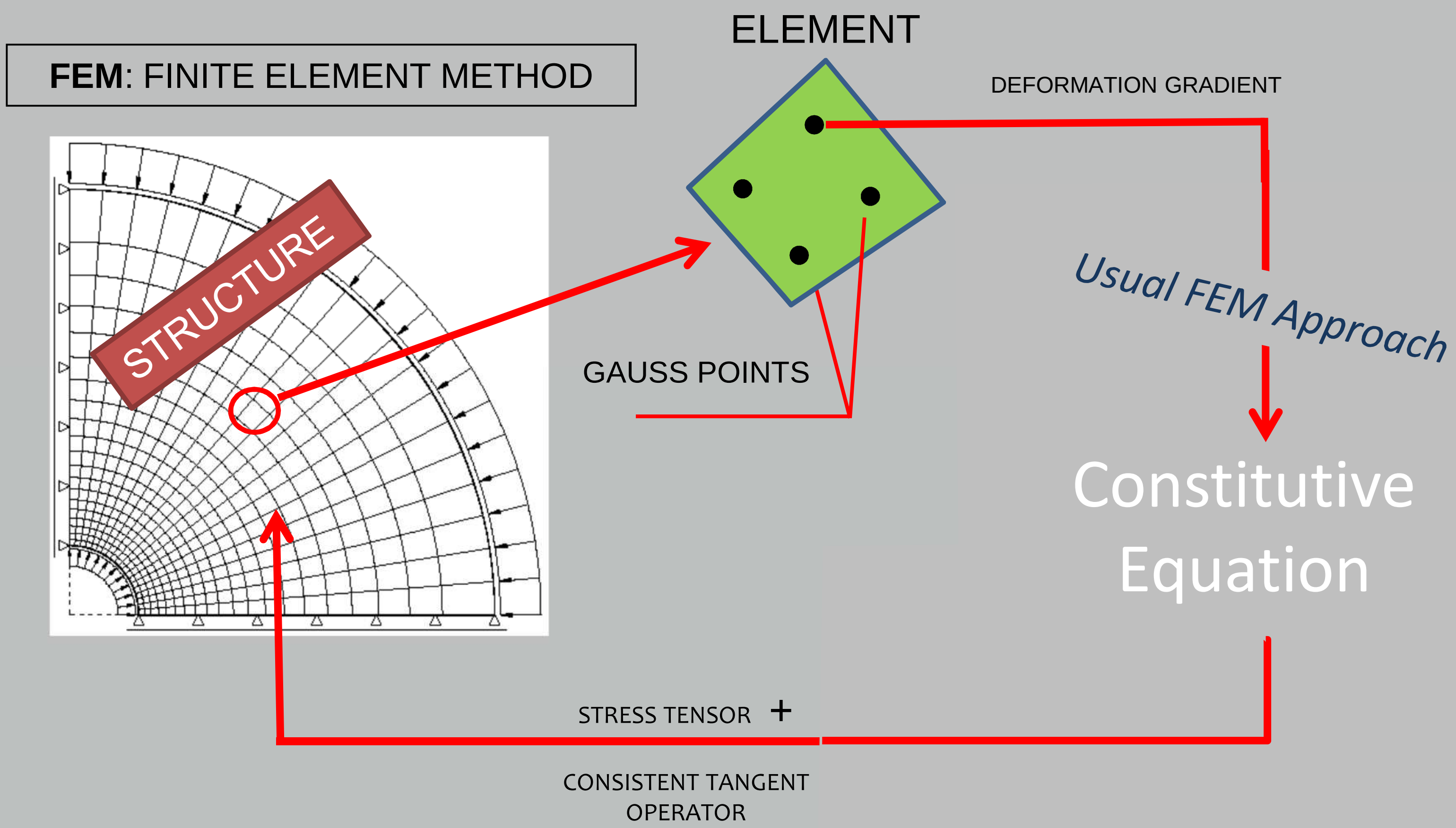
Introduction : bridging scales in Geomechanical modelling

→ **A continuum media**
or
an assembly of particles ?

Continuum : FEM	Particles : DEM
☺ well suited to Real scale problem ☹ CAN NOT realistically model their discrete nature	☺ Reproduces « naturally » the complex behaviour of grains assembly : cyclic response, anisotropy, strain path dependency ☹ Computation time depends on the number of grains -> high CPU costs > limitation to small problems
→ **Coupling FEM-DEM** ☺ ☺	

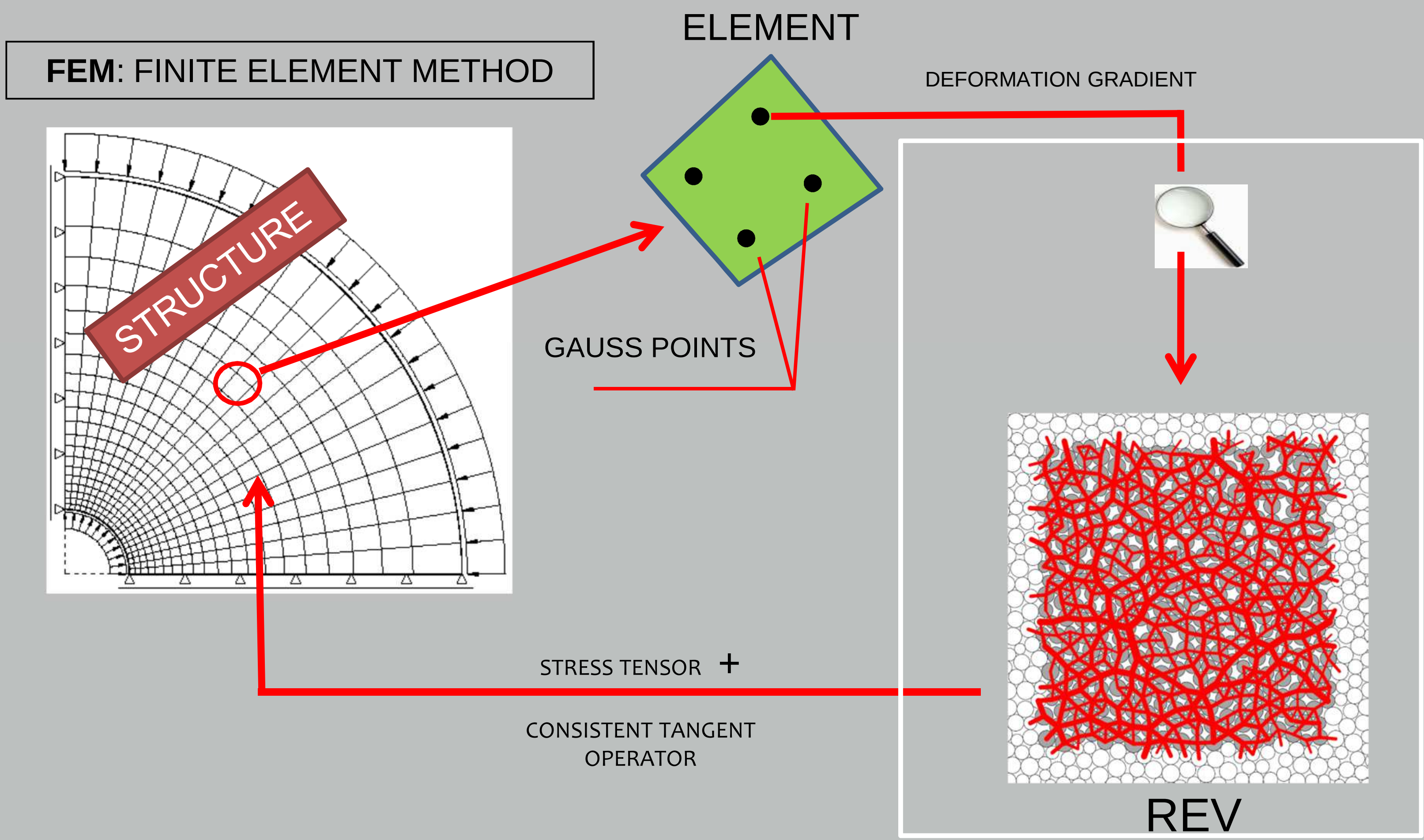
Principle

Introducing a two-scale numerical homogenization approach by FEM - DEM



Principle

A two-scale numerical homogenization approach by FEM - DEM



DEM: DISCRETE ELEMENT METHOD using PBC (periodic boundary conditions)

Micro-scale Model

- **Sorry, it's not YADE**
- The choice made has been to use a large multi-purpose FEM code (Lagamine, Liège University Ulg)

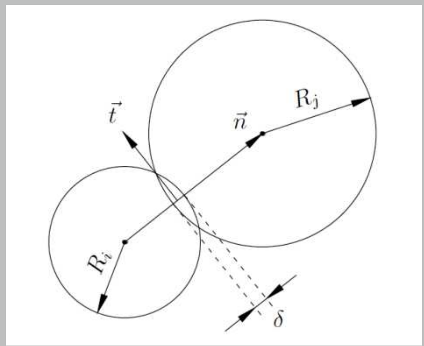
  and to incorporate in this FEM Code an as-compact-as-possible DEM kernel

  -> DEM code developped by Gaël Combe, 3SR
  strong requirement : quasi-perfect static equilibrium
  at the end of each DEM step

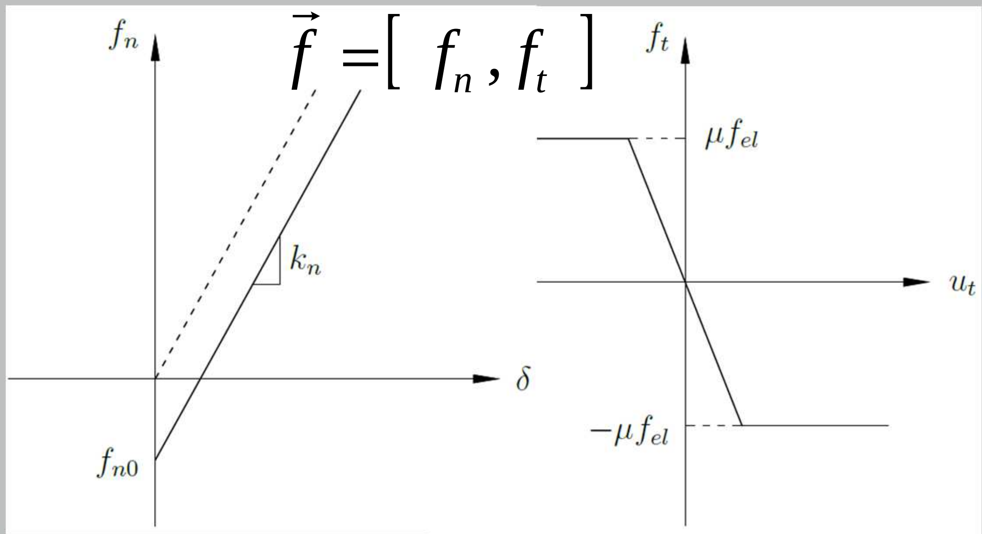
- The other team developping FEMxDEM in the world is in Hong-Kong (JiDong Shao, HK University) essentially along the same lines as our work (after IWBDG 9th Porquerolles)
  … they are using YADE ! … and they are doing excellent work, possible collaboration under consideration

Micro-scale Model

Contact laws *



Discrete **E**lement **M**ethod
(Soft contact dynamics type)
with bi-**P**eriodic **B**oundary **C**onditions



Macrosc. Stress tensor : $\sigma_{ij} = \frac{1}{S} . \sum_{k=1}^{N_C} f_i^k \cdot l_j^k$

- Normal repulsive contact force

$$f_{el} = k_n \cdot \delta$$

$$\begin{cases} \delta > 0 & \text{Contact present} \\ \delta = 0 & \text{No contact} \end{cases}$$

- Tangential contact force

$$\delta f_t = k_t \cdot \delta u_t$$

- Coulomb condition

$$\| f_t \| \leq \mu . f_{el}$$

- Cohesion

$$f_n = f_{el} + f_{n0}$$

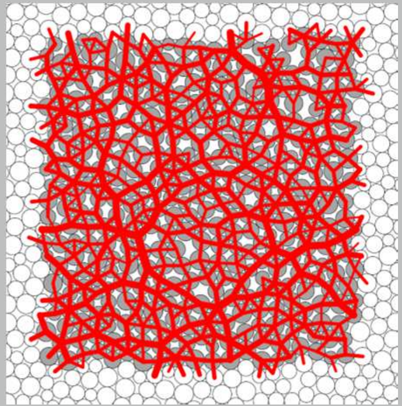
f_{n0} : cohesive force

$$f_{n0} = p^* . \sigma_0 \qquad p^* = 1,2,\ldots$$

* : (e.g. Gilabert et al., 2007)

Micro-scale Model

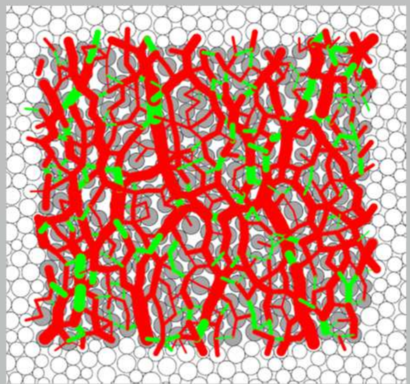
Biaxial test (DEM with PBC): REV contains 400 particles



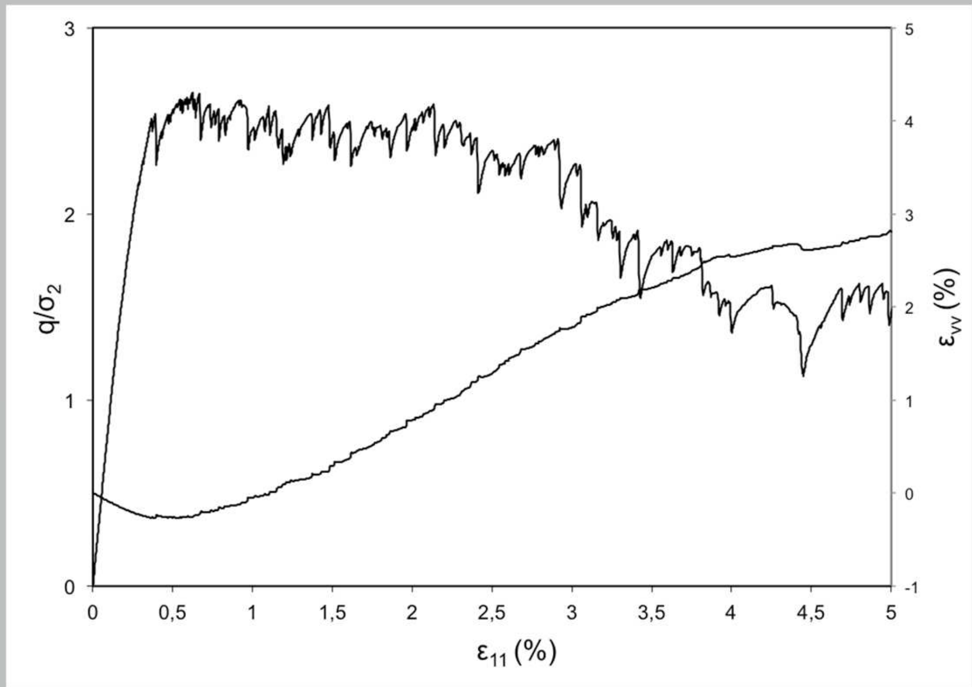
— f_c effective

— $f_c = 0$

Initial configuration

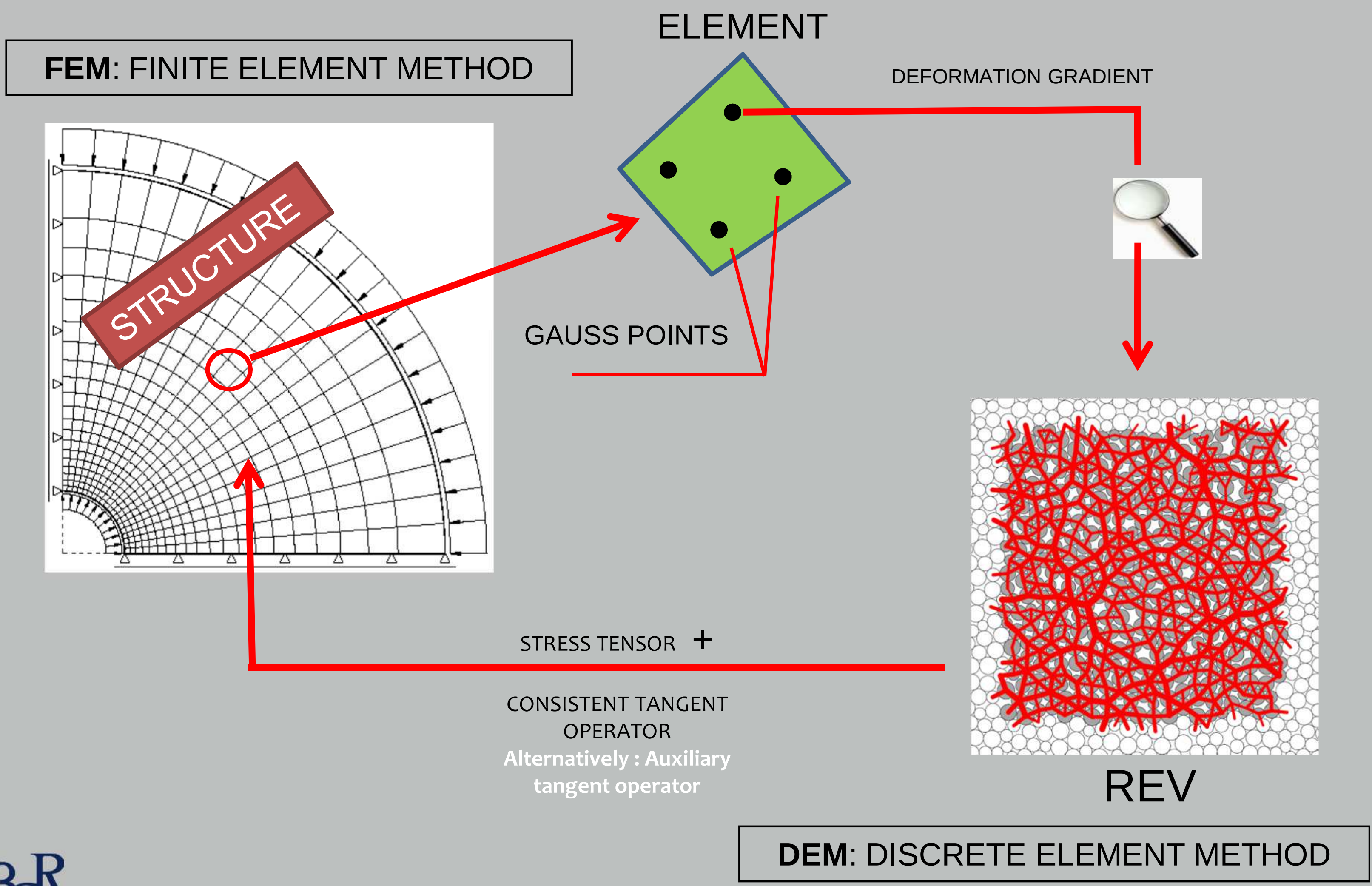


at 3% of axial strain (ε_{11})



Principle

A two-scale numerical homogenization approach by FEM - DEM



Two examples :

1. **Triaxial test :**

   ideally, should be *homogeneous*, but …

   in the lab, observation :

   localised deformation

Triaxial test on Hostun sand specimen, JL Colliat, 3SR Grenoble 1986

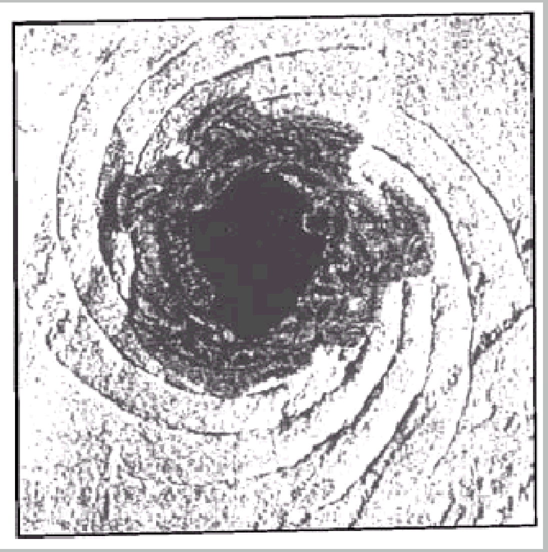
2. **Hollow cylinder under differential pressure**

   (analogous to a borehole or a gallery)

   heterogeneous by essence

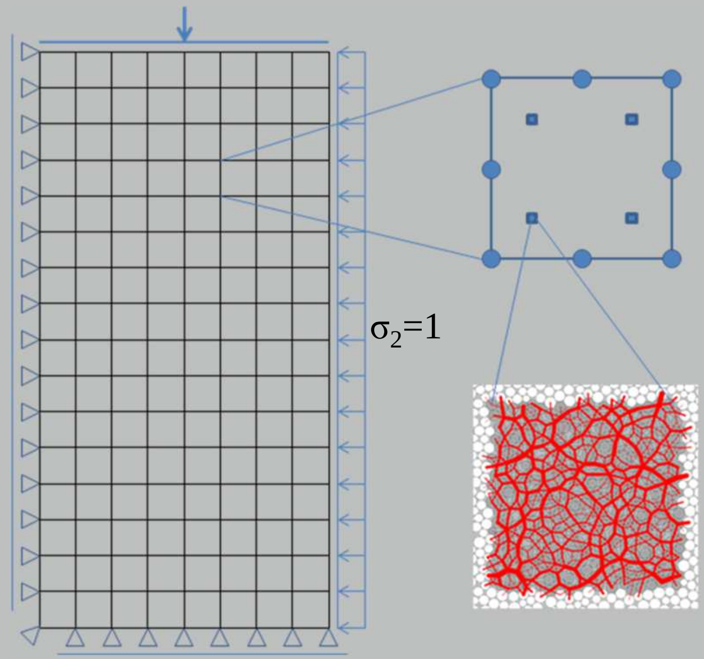
   in the field, observation :

   localised deformation



van den Hoek, P.J., Smit, D.-J., Kooijman, A.P., de Bree, P., Kenter, C.J., Khodaverdian, M., 1994. Size dependancy of hollow-cylinder stability. Eurock, vol. 94. Balkema, Rotterdam.

Multiscale Computations: Numerical results



FEM x DEM simulation of a biaxial compression test

DEM parameters

$$\kappa = k_n / \sigma_0 = 1000$$

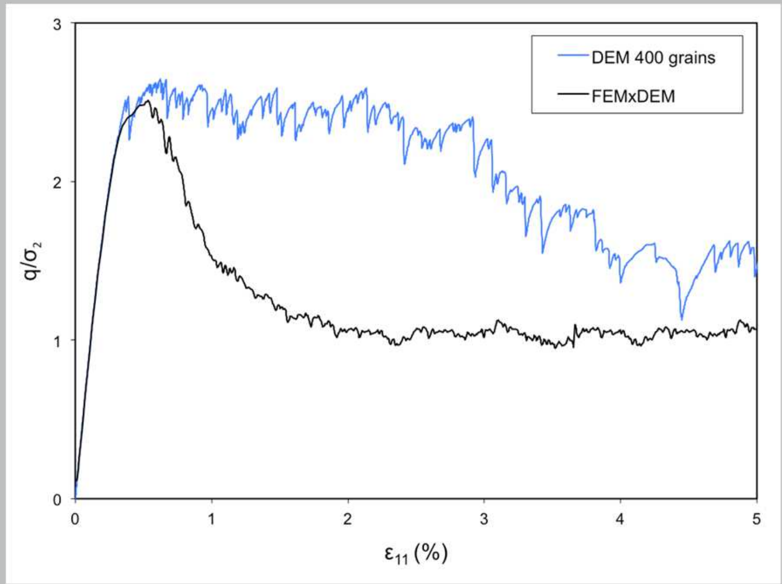
$$k_n / k_t = 1$$

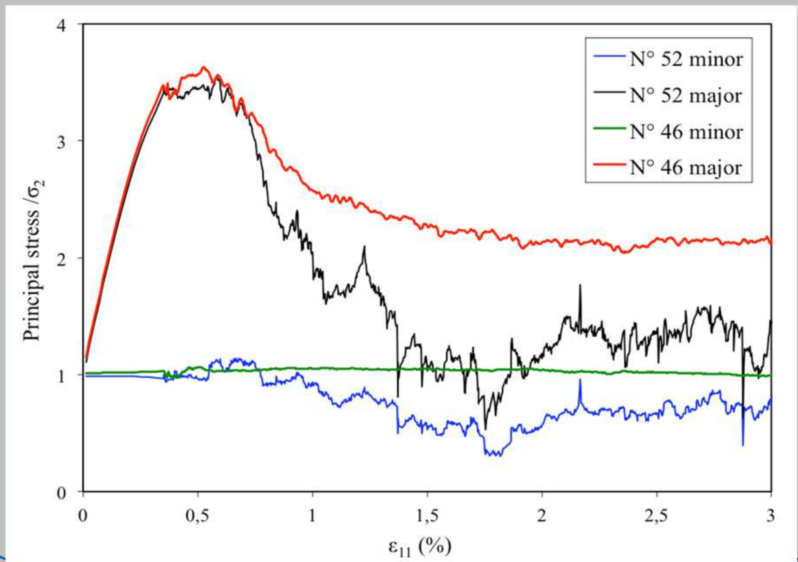
$$\mu = 0.5$$

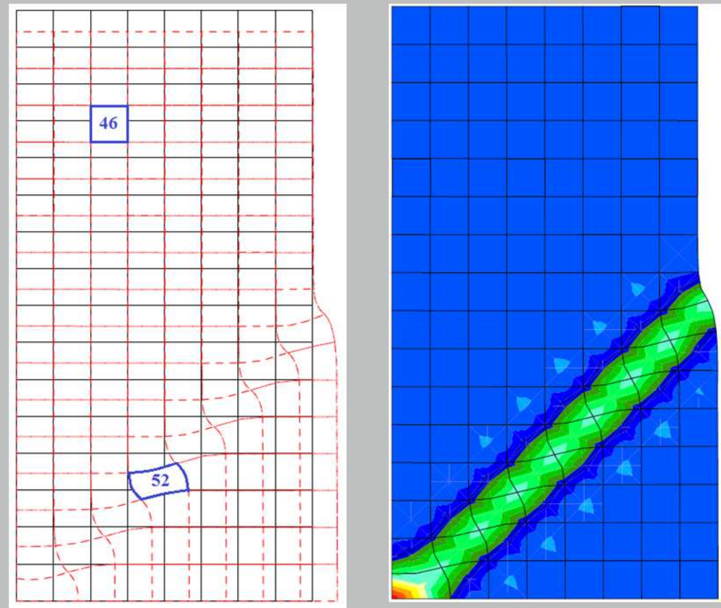
$$p^* = \frac{f_c}{a \cdot \sigma_0} = 1$$

- Macro: discretization by 128 finite elements Q8
- Micro : REV contains 400 grains

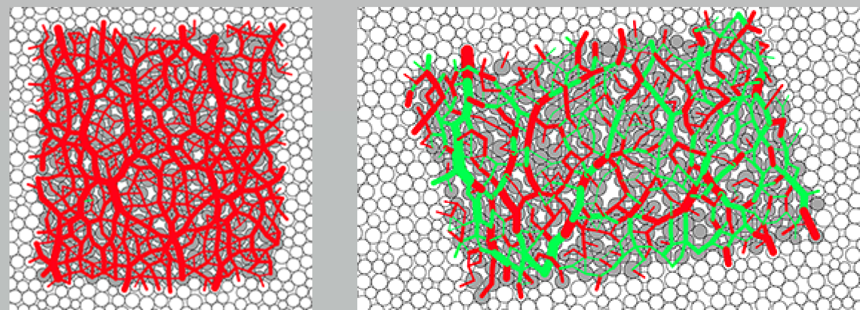
Multiscale Computations: Numerical results : Strain localization







Deformed structure and second invariant of strain tensor



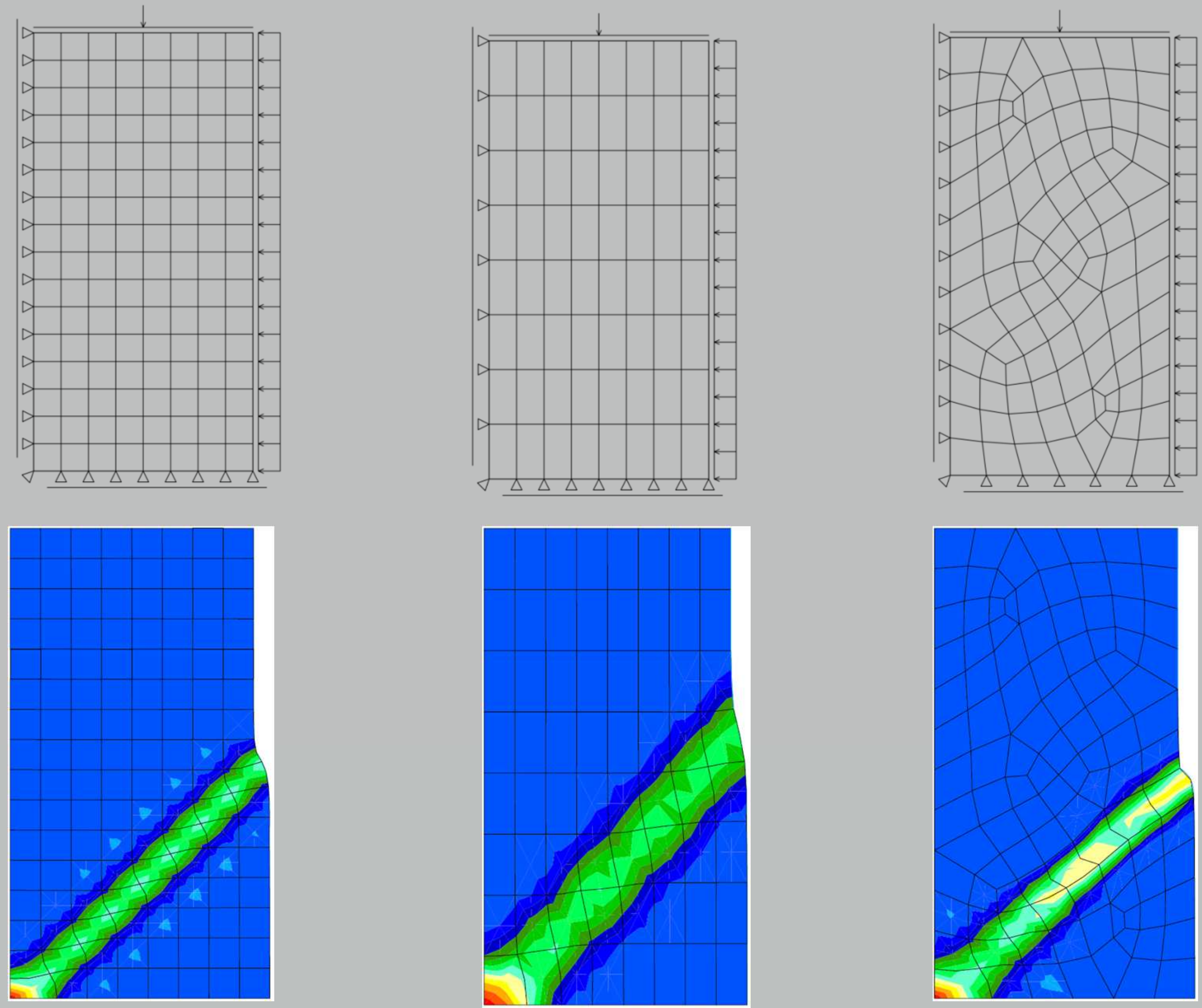
Element 46

Element 52

Deformed REV

Multiscale Computations: different meshes

Mesh dependency as usual in FEM : issues and solutions



128 elements | 64 elements | 106 elements

Two examples :

1. **Triaxial test :**

   ideally, should be *homogeneous*, but …

   in the lab, observation :

   localised deformation

Triaxial test on Hostun sand specimen, JL Colliat, 3SR Grenoble 1986

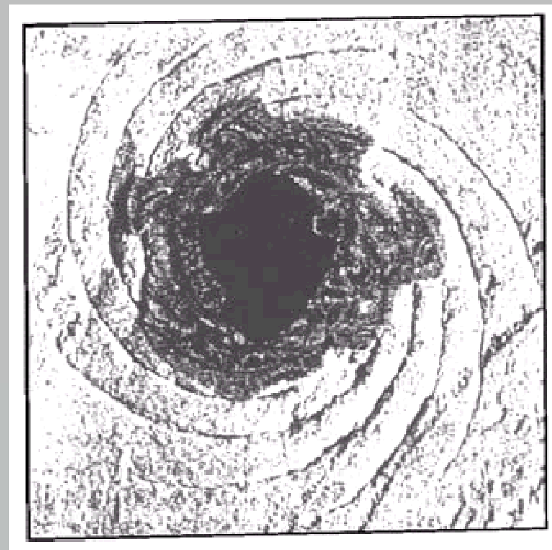
2. **Hollow cylinder under differential pressure**

   (analogous to a borehole or a gallery)

   heterogeneous by essence

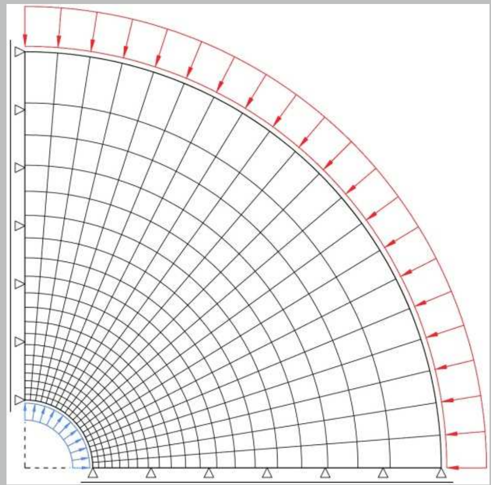
   in the field, observation :

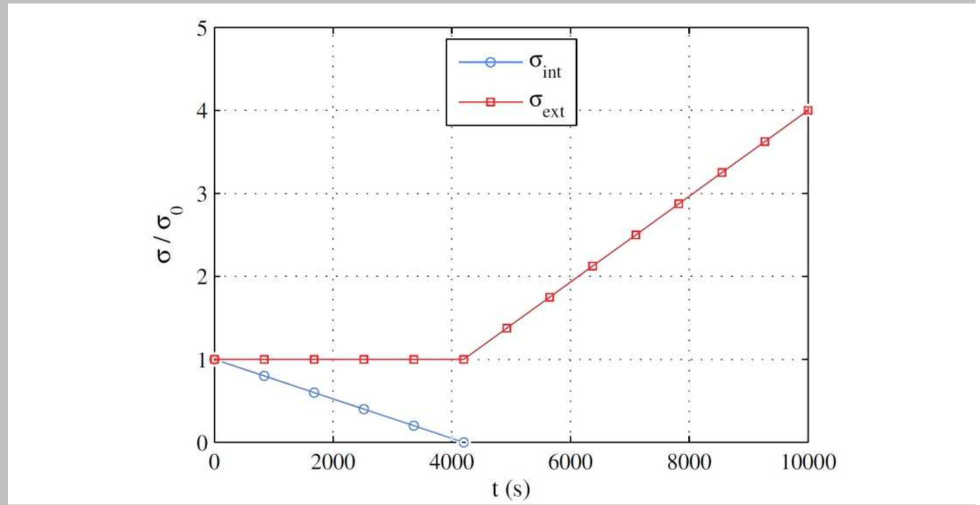
   localised deformation



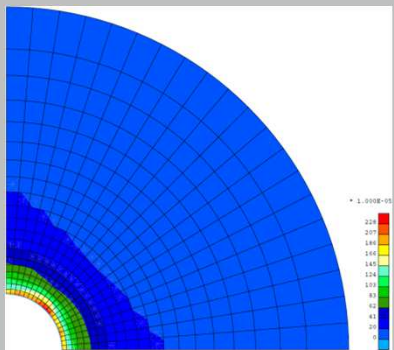
van den Hoek, P.J., Smit, D.-J., Kooijman, A.P., de Bree, P., Kenter, C.J., Khodaverdian, M., 1994. Size dependancy of hollow-cylinder stability. Eurock, vol. 94. Balkema, Rotterdam.

Multiscale Computations: Hollow cylinder (drilling), Strain localization

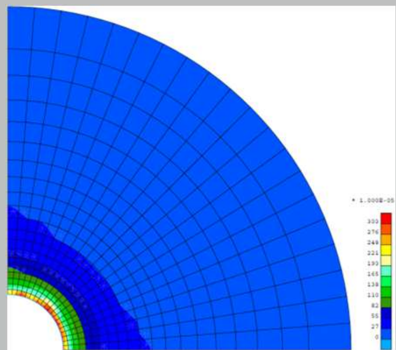




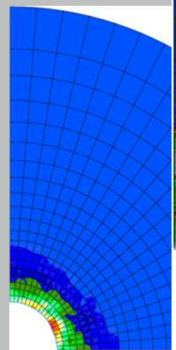
Deformed structure and second invariant of strain tensor



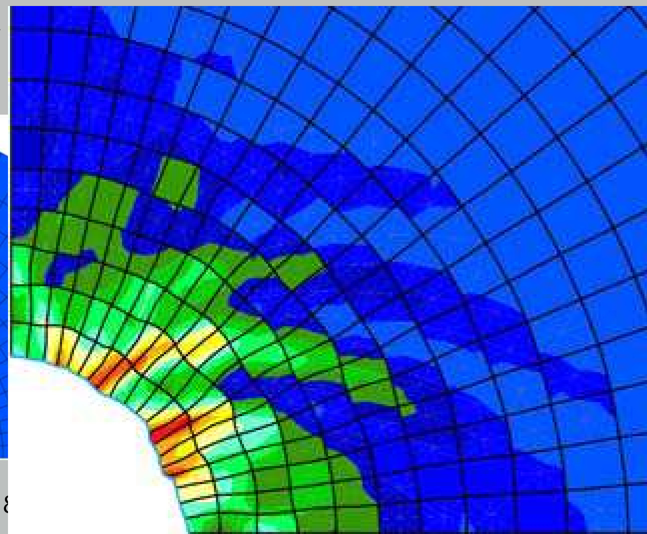
t = 3700
$\varepsilon_{equ\ max}$ = 228. 10^{-5}



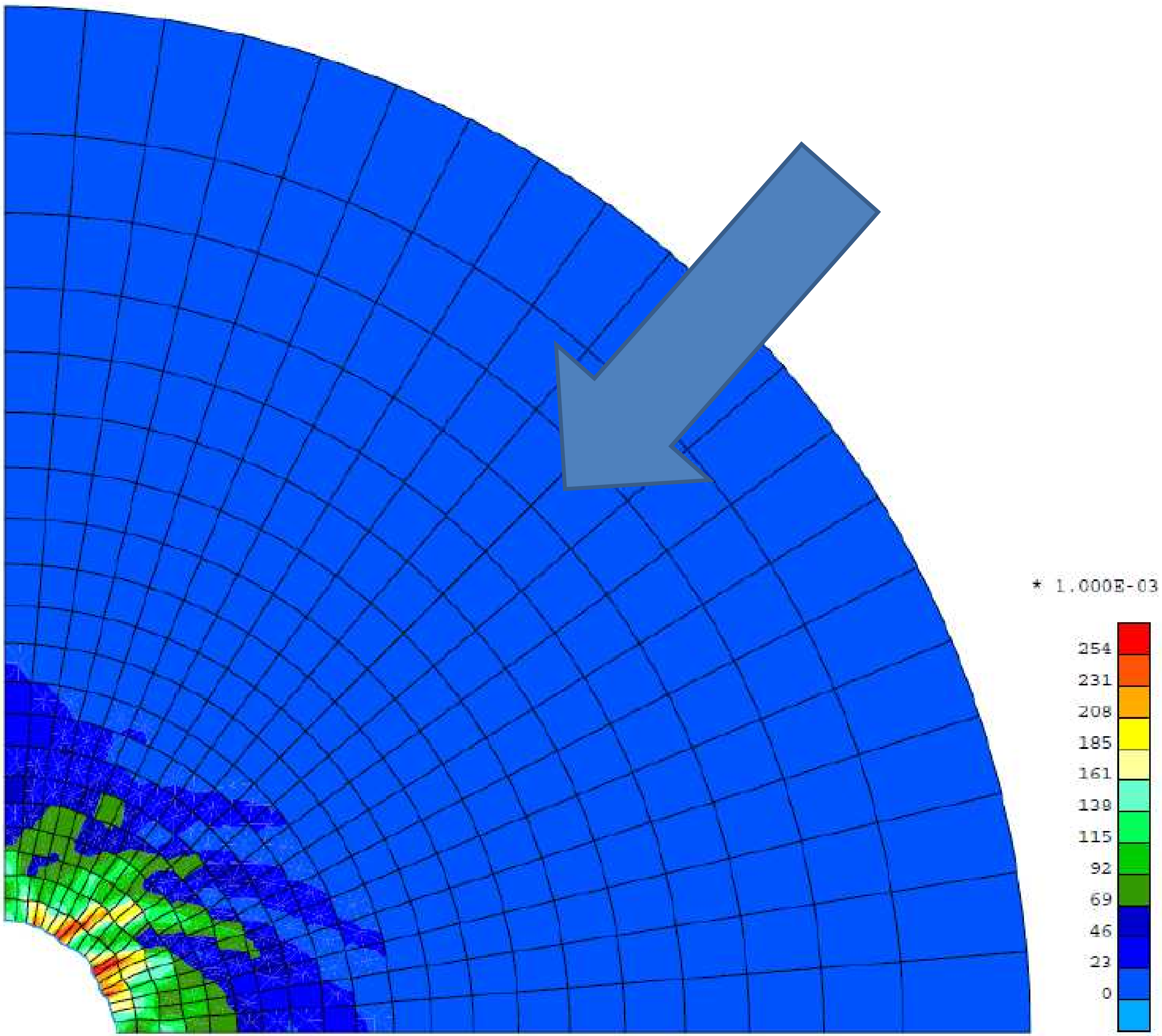
t = 4000
$\varepsilon_{equ\ max}$ = 300. 10^{-5}

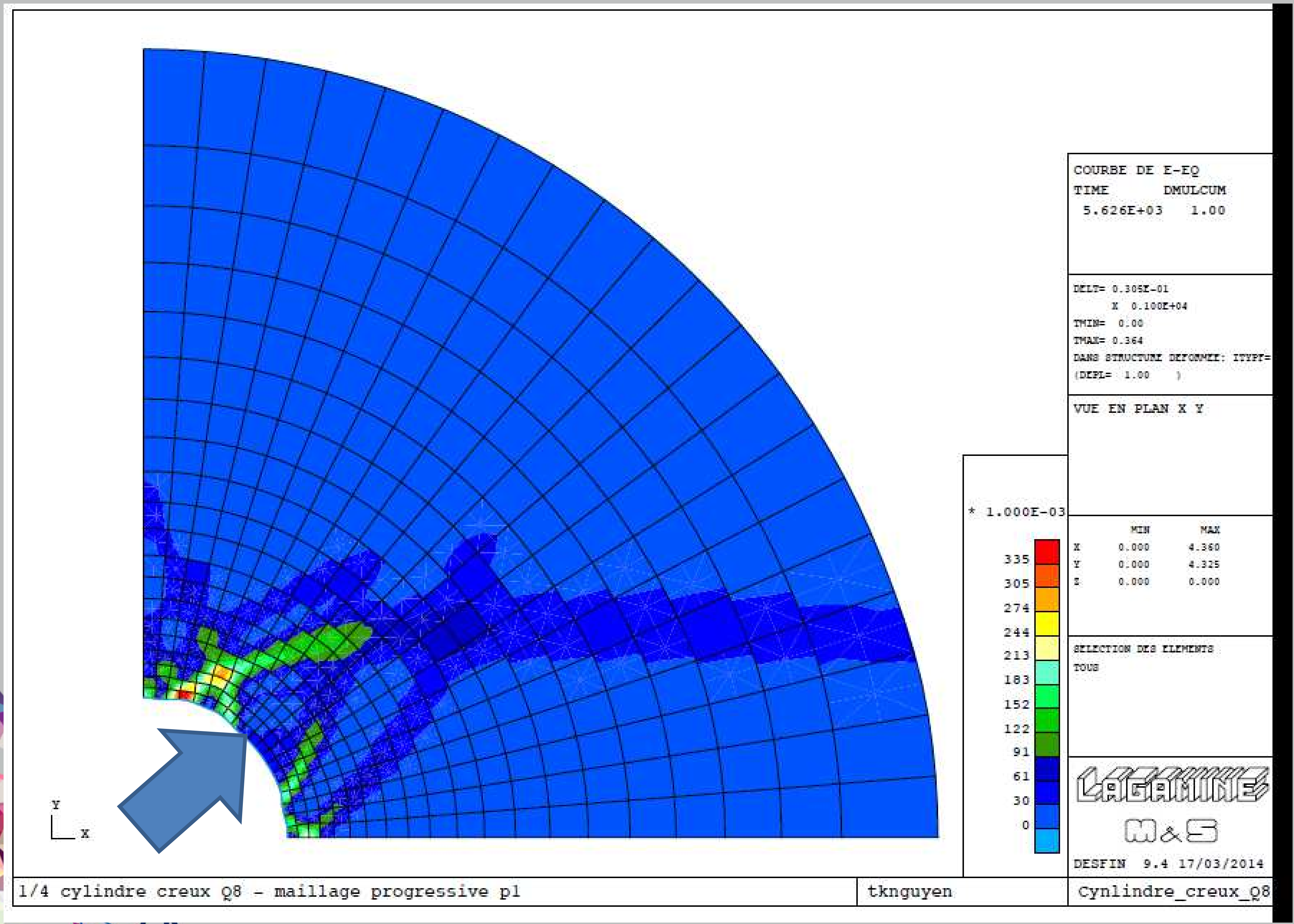


t = 8
$\varepsilon_{equ\ max}$ = 124. 10



$\varepsilon_{equ\ max}$ = 254. 10


* 1.000E-03
254
231
208
185
161
138
115
92
69
46
23
0


COURBE DE E-EQ
TIME DMULCUM
5.626E+03 1.00
DELT= 0.305E-01
X 0.100E+04
TMIN= 0.00
TMAX= 0.364
DANS STRUCTURE DEFORMEE: ITYPF=
(DEPL= 1.00)
VUE EN PLAN X Y
* 1.000E-03
335
305
274
244
213
183
152
122
91
61
30
0
MIN MAX
X 0.000 4.360
Y 0.000 4.325
Z 0.000 0.000
SELECTION DES ELEMENTS
TOUS
Y
X
DESFIN 9.4 17/03/2014
1/4 cylindre creux Q8 - maillage progressive p1
tknguyen
Cynlindre_creux_Q8

Conclusions & Perspectives

CONCLUSIONS

- We have presented a Two-scale numerical approach for granular materials: combining FEM (at macro scale) and DEM (at micro scale).
- Illustration by two examples of BVP :
- a biaxial compression test and
- a hollow cylinder (analogy of underground excavations and drilling)
- Strain localization was observed in both cases. Mesh dependency confirmed.

NOT PRESENTED

"low / strong" cohesion model to manage assemblies of cemented grains made of glued disks

PERSPECTIVES

-3D approach
- Second gradient regularisation

Open Source DEM-FEM Coupling

Jan Stránský

Czech Technical University in Prague
Faculty of Civil Engineering
Department of Mechanics

7. 7. 2014

Outline

1. Motivation
2. MuPIF
3. Examples
   - Surface coupling
   - FE x DE (multiscale) coupling
   - Contact coupling
4. Conclusion

Outline

1. Motivation
2. MuPIF
3. Examples
   - Surface coupling
   - FE x DE (multiscale) coupling
   - Contact coupling
4. Conclusion

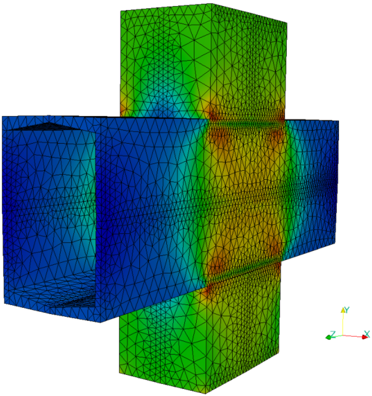
Outline

Outline

Motivation

Numerical methods for solid mechanics

- Finite element method (FEM)
  - approximate solution of PDEs
  - continuous deformable bodies
- Discrete element method (DEM)
  - "discrete" materials
  - contact detection

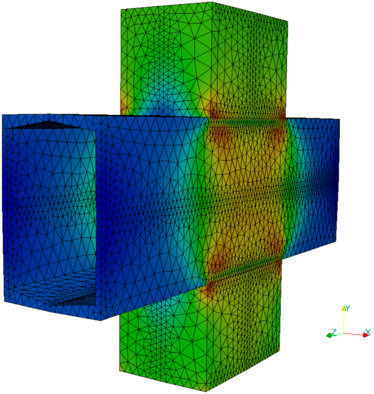


www.oofem.org/wiki/doku.php?id=gallery:steelplasticity

Motivation

Numerical methods for solid mechanics

- Finite element method (FEM)
    - approximate solution of PDEs
    - continuous deformable bodies
- Discrete element method (DEM)
    - "discrete" materials
    - contact detection

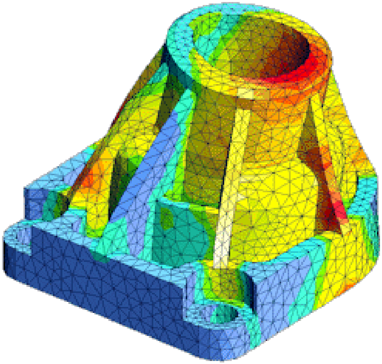


www.oofem.org/wiki/doku.php?id=gallery:steelplasticity

Motivation

Numerical methods for solid mechanics

- Finite element method (FEM)
  - approximate solution of PDEs
  - continuous deformable bodies
- Discrete element method (DEM)
  - "discrete" materials
  - contact detection

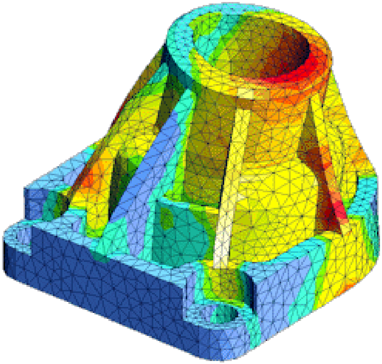


www.hoeft.eu/main_6.html

Motivation

Numerical methods for solid mechanics

- Finite element method (FEM)
  - approximate solution of PDEs
  - continuous deformable bodies
- Discrete element method (DEM)
  - "discrete" materials
  - contact detection

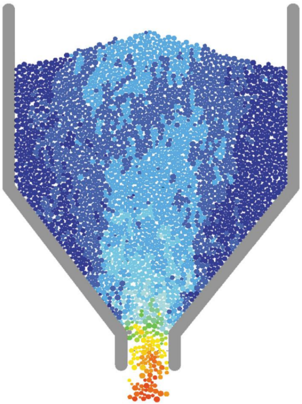


www.hoeft.eu/main_6.html

Motivation

Numerical methods for solid mechanics

- Finite element method (FEM)
  - approximate solution of PDEs
  - continuous deformable bodies
- Discrete element method (DEM)
  - ”discrete” materials
  - contact detection

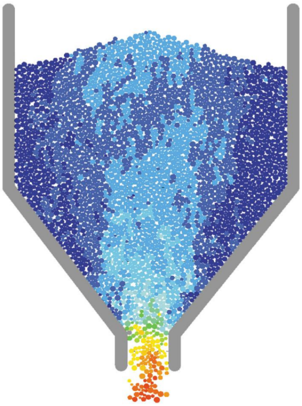


geo.hmg.inpg.fr/frederic/Research_projects_missile_impacts.html

Motivation

Numerical methods for solid mechanics

- Finite element method (FEM)
  - approximate solution of PDEs
  - continuous deformable bodies
- Discrete element method (DEM)
  - ”discrete” materials
  - contact detection

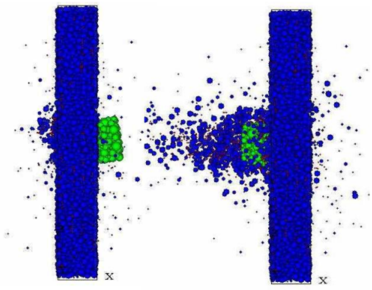


geo.hmg.inpg.fr/frederic/Research_projects_missile_impacts.html

Motivation

Numerical methods for solid mechanics

- Finite element method (FEM)
  - approximate solution of PDEs
  - continuous deformable bodies
- Discrete element method (DEM)
  - ”discrete” materials
  - contact detection



www.epfl.ch/SIC/SA/publications/SCR99/scr11-page23.html

Motivation

Numerical methods for solid mechanics

- Finite element method (FEM)
  - approximate solution of PDEs
  - continuous deformable bodies
- Discrete element method (DEM)
  - ”discrete” materials
  - contact detection

www.featurepics.com/online/Tire-Tread-Beach-Sand-1393666.aspx

Motivation

Numerical methods for solid mechanics

- Finite element method (FEM)
  - approximate solution of PDEs
  - continuous deformable bodies
- Discrete element method (DEM)
  - ”discrete” materials
  - contact detection

www.litomysky.cz/drahy/prazcesterk.htm

Motivation

Numerical methods for solid mechanics

- Finite element method (FEM)
  - approximate solution of PDEs
  - continuous deformable bodies
- Discrete element method (DEM)
  - ”discrete” materials
  - contact detection

gravel-lok.blogspot.com/2009_11_01_archive.html

Motivation

Numerical methods for solid mechanics

- Finite element method (FEM)
  - approximate solution of PDEs
  - continuous deformable bodies
- Discrete element method (DEM)
  - ”discrete” materials
  - contact detection

⇓

Combination

- Monolithic application
  - redoing what already exists
- Combine exsting codes
  - open source code is preferable

gravel-lok.blogspot.com/2009_11_01_archive.html

Motivation

Numerical methods for solid mechanics

- Finite element method (FEM)
  - approximate solution of PDEs
  - continuous deformable bodies
- Discrete element method (DEM)
  - ”discrete” materials
  - contact detection

⇓

Combination

- Monolithic application
  - redoing what already exists
- Combine exsting codes
  - open source code is preferable

gravel-lok.blogspot.com/2009_11_01_archive.html

Motivation

Numerical methods for solid mechanics

- Finite element method (FEM)
  - approximate solution of PDEs
  - continuous deformable bodies
- Discrete element method (DEM)
  - ”discrete” materials
  - contact detection

⇓

Combination

- Monolithic application
  - redoing what already exists
- Combine exsting codes
  - open source code is preferable

gravel-lok.blogspot.com/2009_11_01_archive.html

Motivation

Numerical methods for solid mechanics

- Finite element method (FEM)
  - approximate solution of PDEs
  - continuous deformable bodies
- Discrete element method (DEM)
  - ”discrete” materials
  - contact detection

⇓

Combination

- Monolithic application
  - redoing what already exists
- Combine exsting codes
  - open source code is preferable

gravel-lok.blogspot.com/2009_11_01_archive.html

Motivation

Numerical methods for solid mechanics

- Finite element method (FEM)
  - approximate solution of PDEs
  - continuous deformable bodies
- Discrete element method (DEM)
  - ”discrete” materials
  - contact detection

⇓

Combination

- Monolithic application
  - redoing what already exists
- Combine exsting codes
  - open source code is preferable

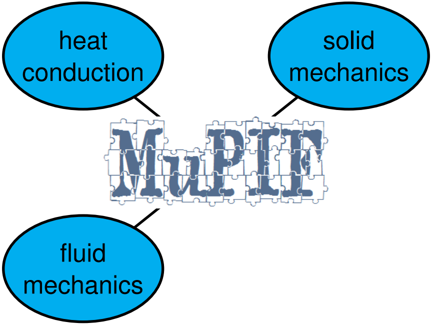
gravel-lok.blogspot.com/2009_11_01_archive.html

MuPIF

MuPIF = Multi-Physic Integration Framework

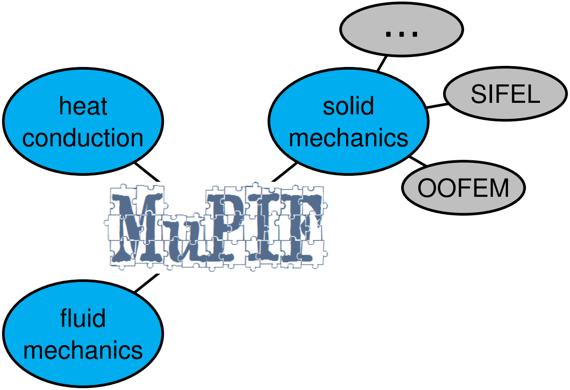
MuPIF

MuPIF = Multi-Physic Integration Framework



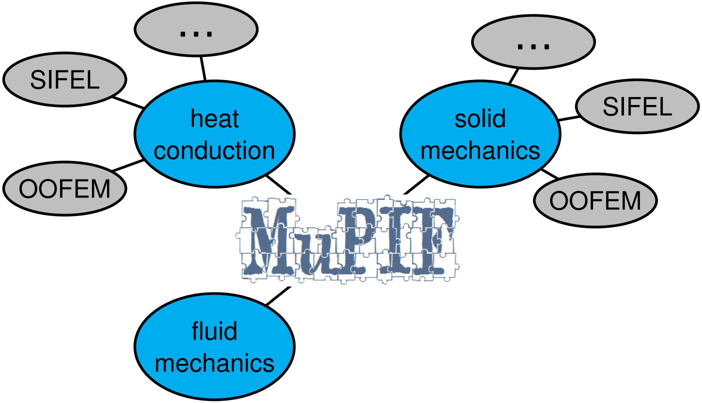
MuPIF

MuPIF = Multi-Physic Integration Framework



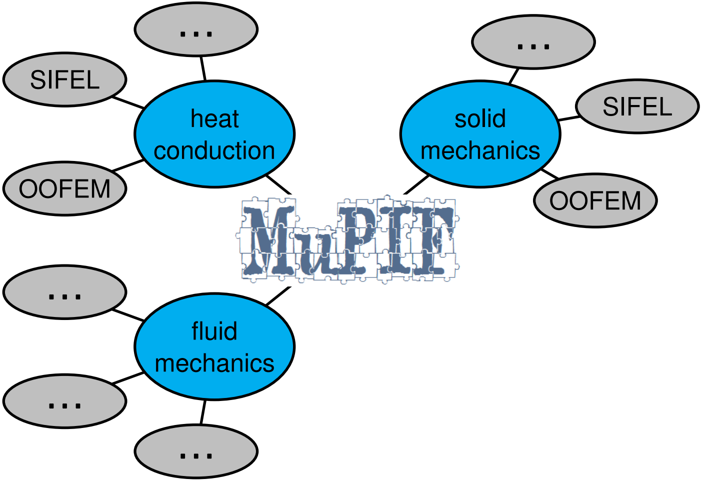
MuPIF

MuPIF = Multi-Physic Integration Framework



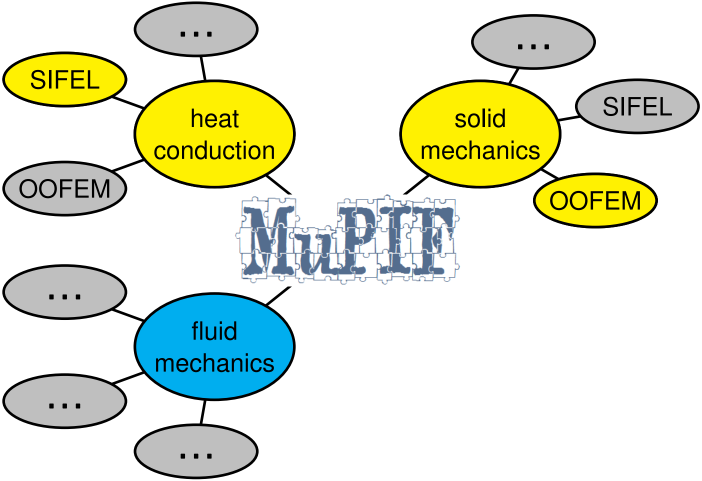
MuPIF

MuPIF = Multi-Physic Integration Framework



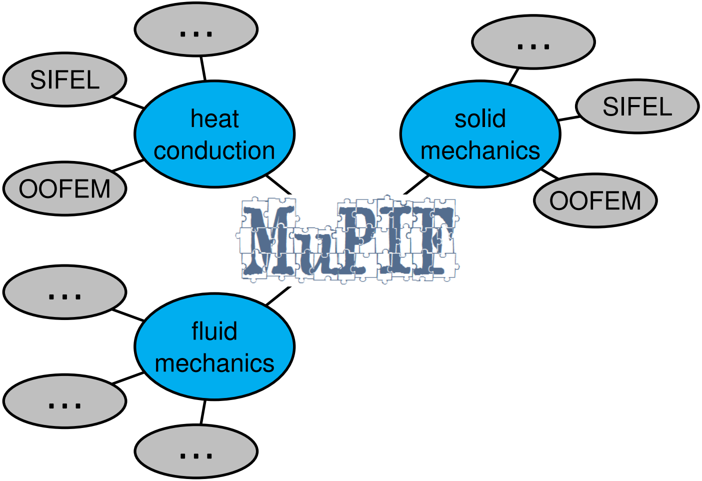
MuPIF

MuPIF = Multi-Physic Integration Framework



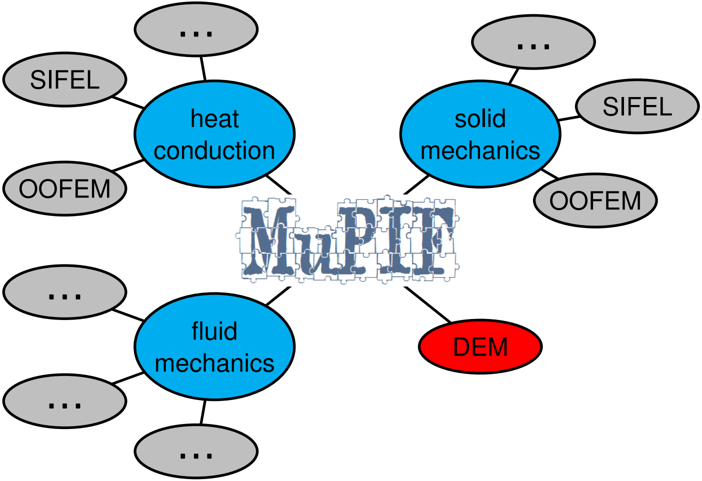
MuPIF

MuPIF = Multi-Physic Integration Framework



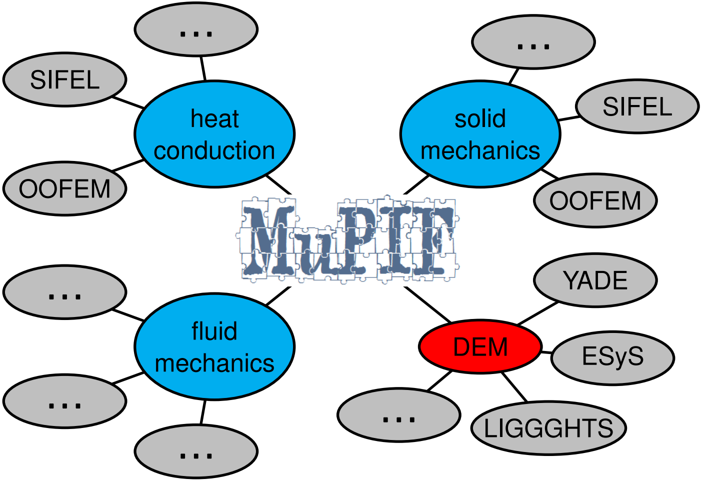
MuPIF

MuPIF = Multi-Physic Integration Framework



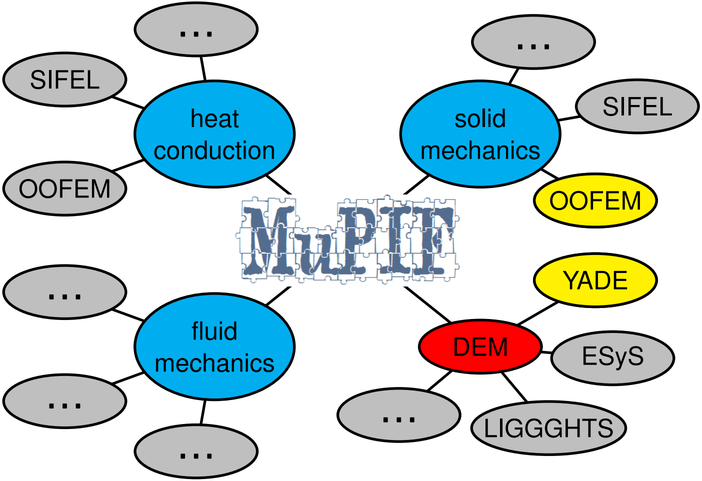
MuPIF

MuPIF = Multi-Physic Integration Framework



MuPIF

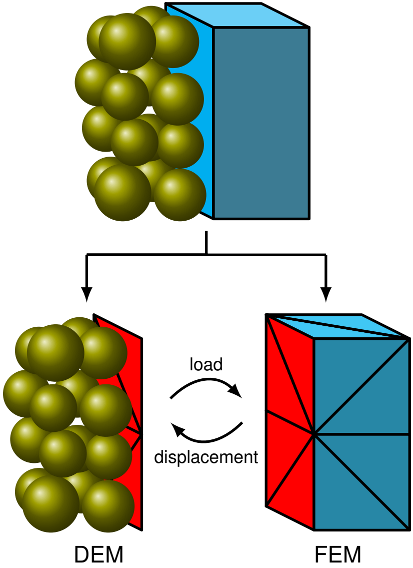
MuPIF = Multi-Physic Integration Framework



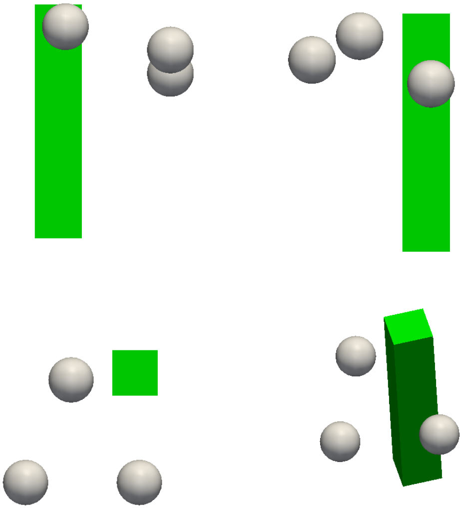
Examples

- Software
  - MuPIF
  - YADE
  - OOFEM
- Solution
  - expicit
  - linear

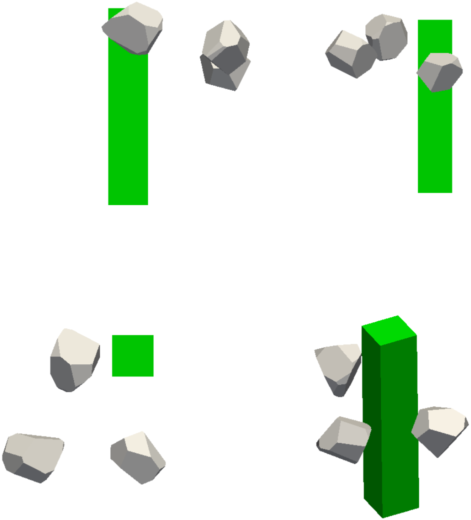
Surface coupling



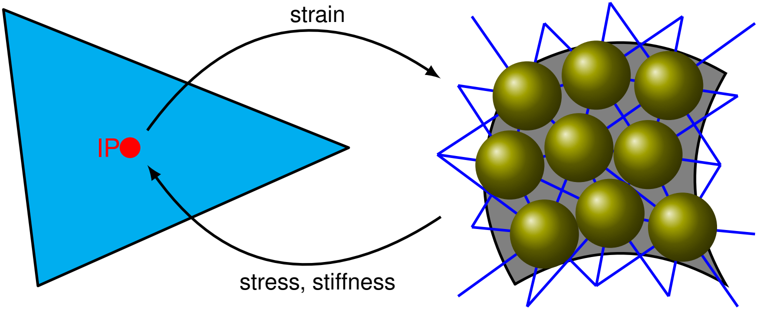
Surface coupling - example



Surface coupling - example 2

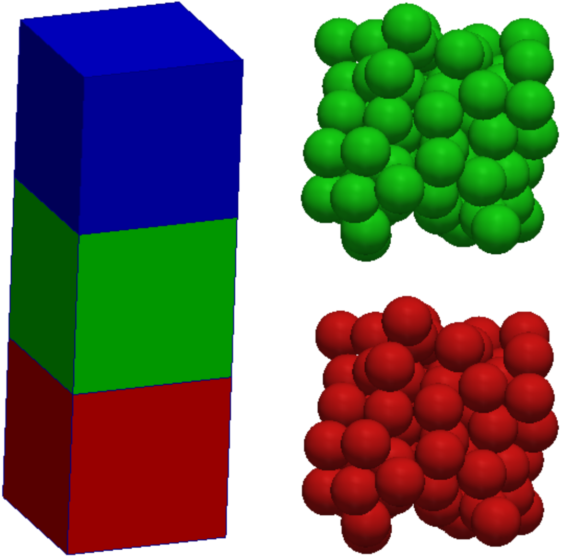


FE x DE (multiscale) coupling

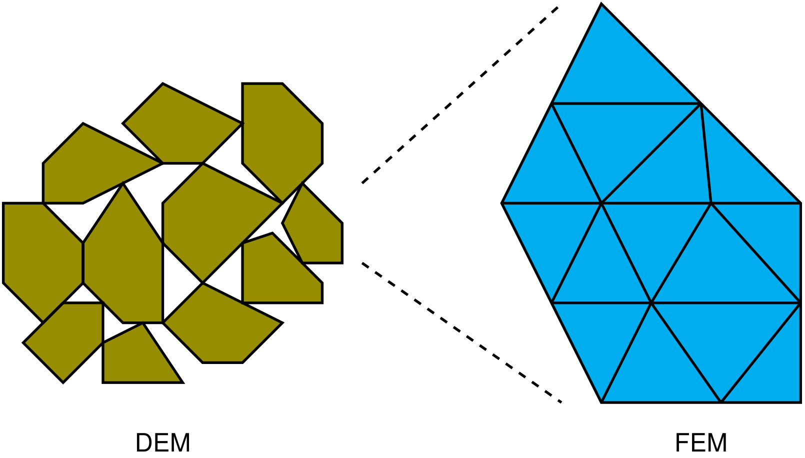


FEM DEM

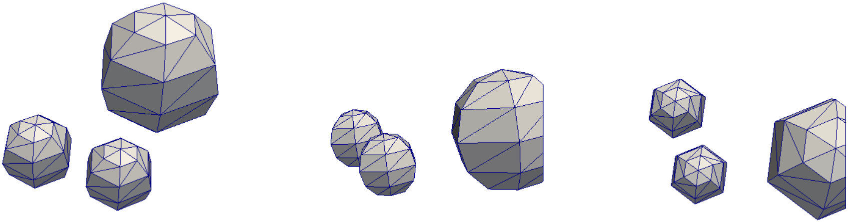
FE x DE (multiscale) coupling - example



Contact coupling



Contact coupling - example



Conclusion

Conclusion

Summary

- FEM–DEM coupling presented
  - using `MuPIF` framework
  - surface coupling
  - FE x DE (multiscale) coupling
  - contact coupling
- all examples are available from Internet

Conclusion

Summary

- FEM–DEM coupling presented
  - using `MuPIF` framework
  - surface coupling
  - FE x DE (multiscale) coupling
  - contact coupling
- all examples are available from Internet

Future work

- volume couplings
- testing of more complicated behavior
- improvement and extension (volume couplings, implicit solutions . . .)
- documentation improvement
- . . .

Acknowledgement

Financial support of the Czech Technical University in Prague under project SGS13/034/OHK1/1T/11 is gratefully acknowledged

jan.stransky@fsv.cvut.cz

Simulation of (rockfall) wire meshes with Yade

K. Thoeni, **A. Giacomini**, **S.W. Sloan**
Centre for Geotechnical and Materials Modelling
The University of Newcastle, Australia

C. Lambert
Department of Civil and Natural Resources Engineering
University of Canterbury, New Zealand

1st Yade Workshop, 7-9 July 2014, Grenoble

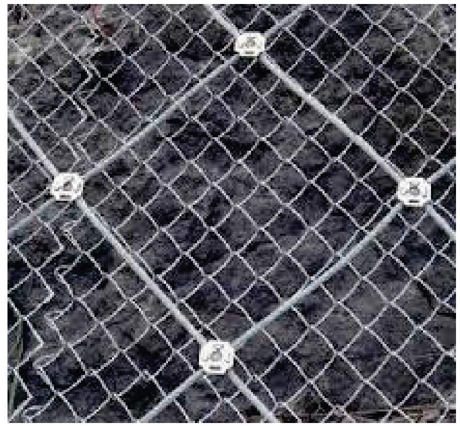
Content

Content

1. Introduction
2. The Wire Model in Yade
3. How to use
4. Some Examples
5. An application: Drapery Systems
6. Future Developments

1. Introduction

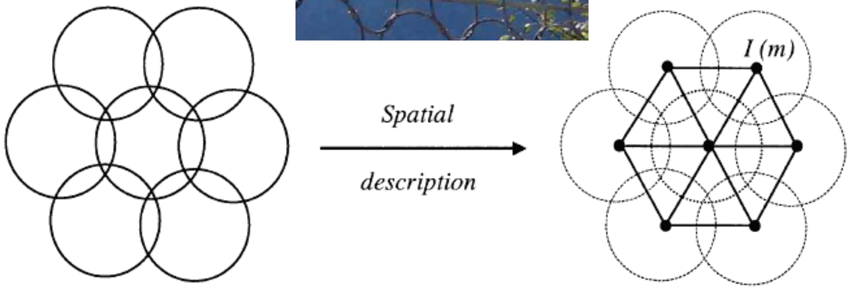
- Various wire meshes are used:
  - double-twist
  - chain-link
  - orthogonal
  - …



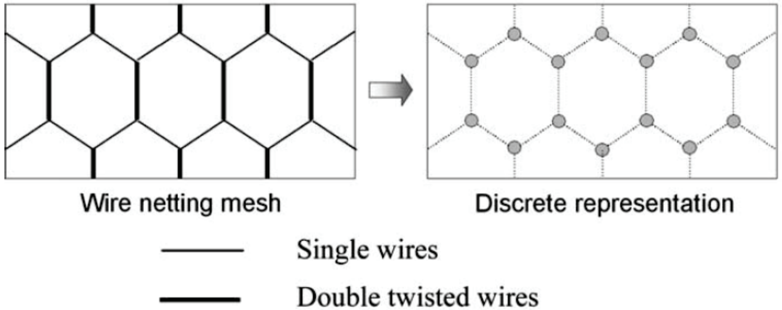
1. Introduction

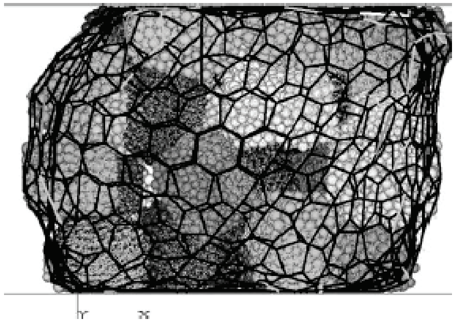
DEM used for modelling of rockfall meshes:

- Nicot et al. (2001):
  - ASM ring net
  - every ring is represented by a particle
  - remote interaction model
  - tensile forces only



- Bertrand et al. (2005, 2008):
  - double-twisted hexagonal wire mesh
  - particle at physical nodes of the mesh
  - remote interaction model
  - tensile forces only

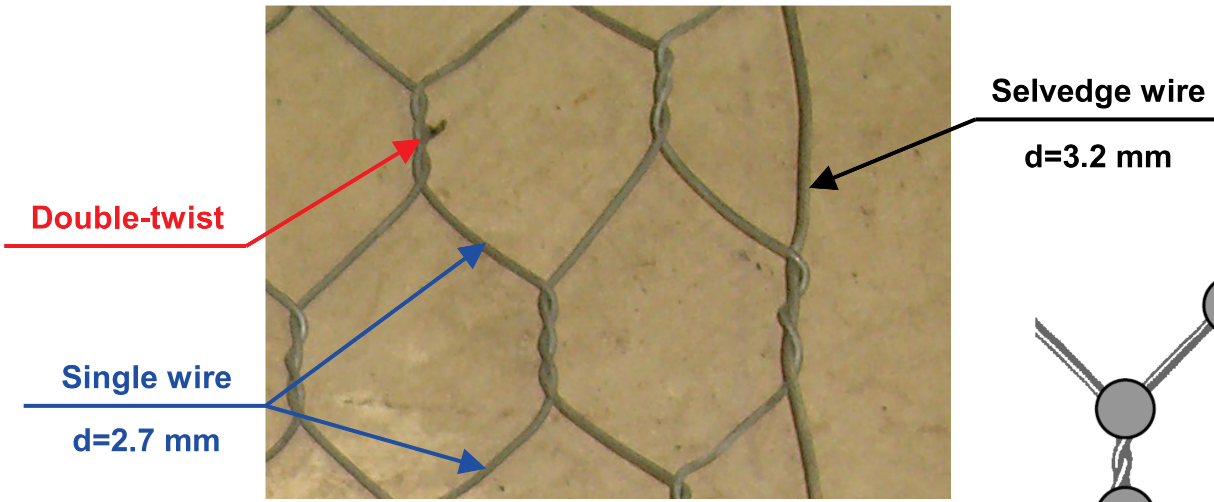


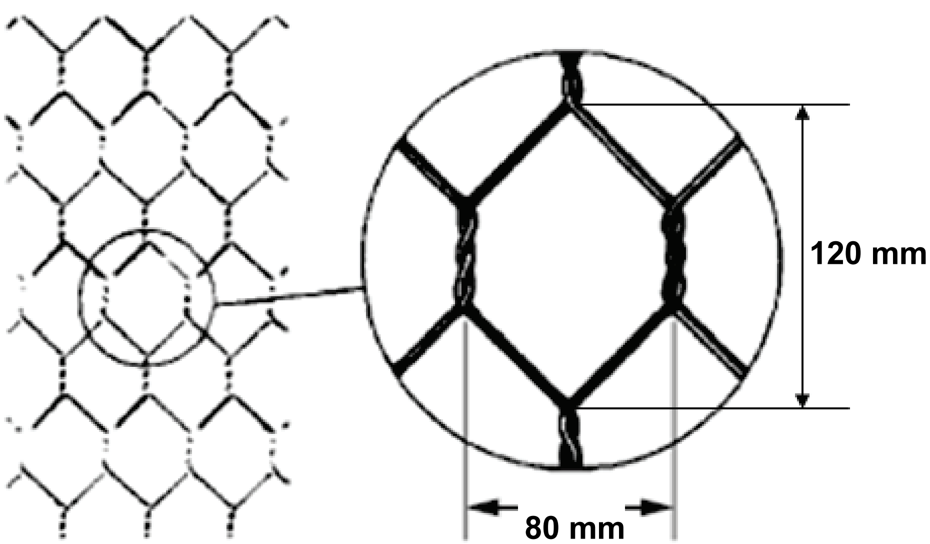


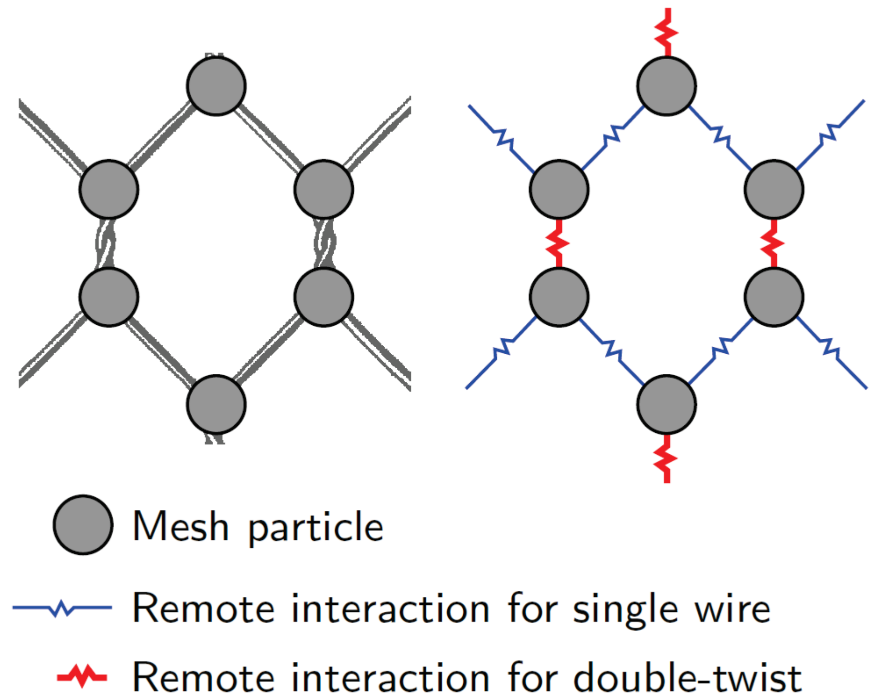
Content

2. The Wire Model in Yade: Concept

Maccaferri double-twisted hexagonal wire mesh:







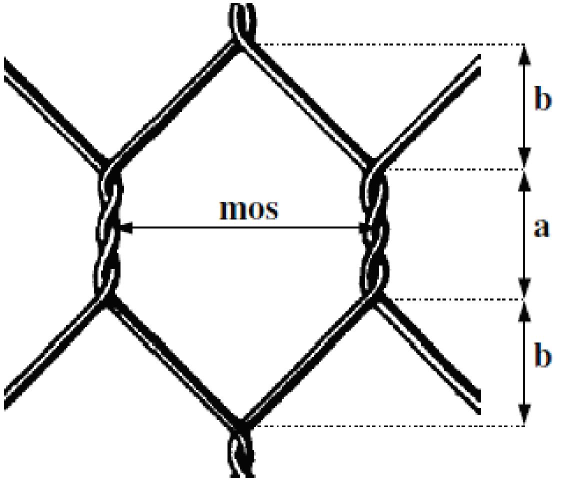
THE UNIVERSITY OF NEWCASTLE AUSTRALIA

2. The Wire Model in Yade: Representation

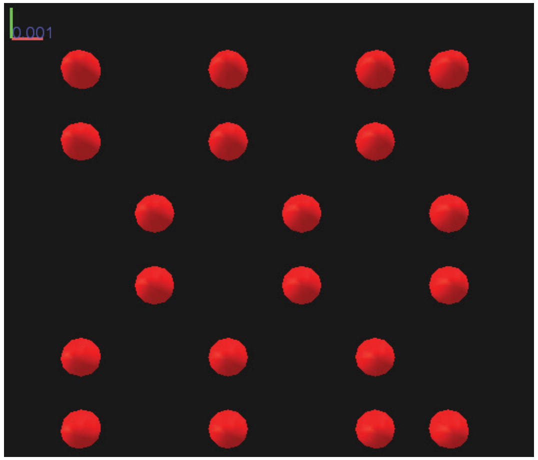
Generation of particles:

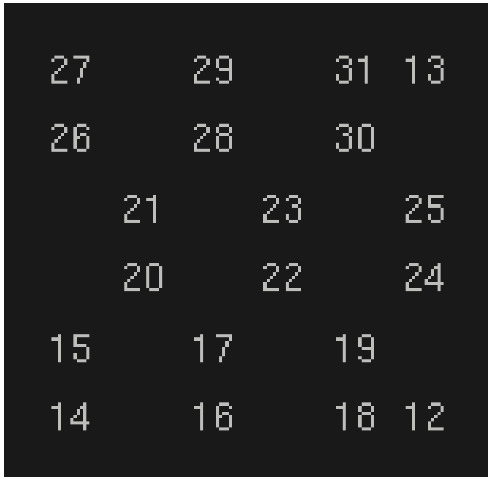
- Generation of particle at physical nodes of the mesh:

```
pack.hexaNet( radius, cornerCoord,
   xLength, yLength, mos, a, b,
   startAtCorner, isSymmetric, **kw )
```



- Corner particles are automatically generated
- Numbering of particles defines type of interaction





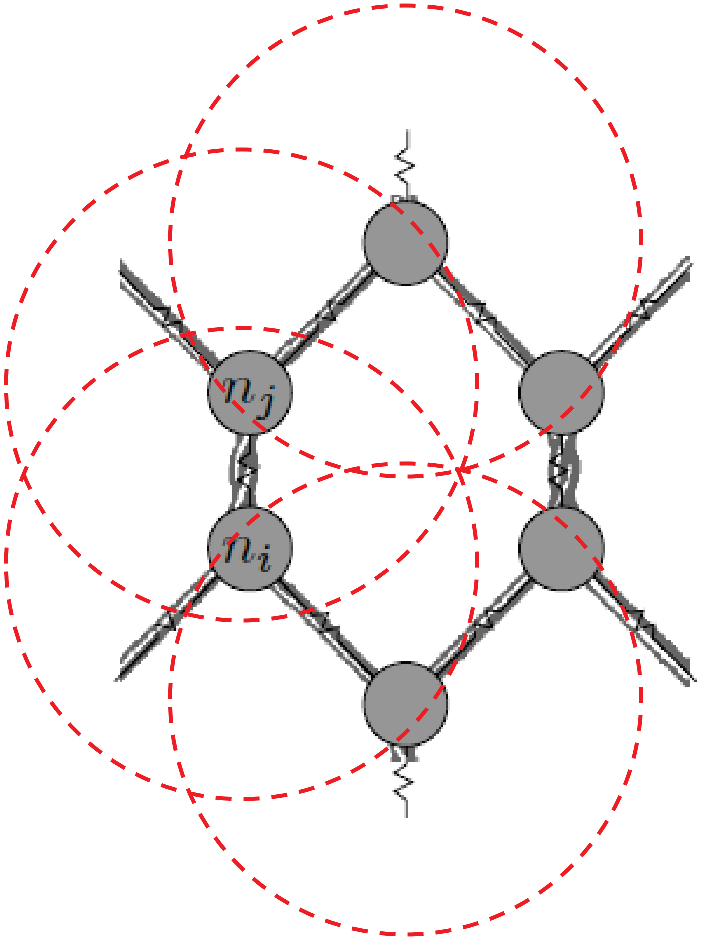
2. The Wire Model in Yade: Representation

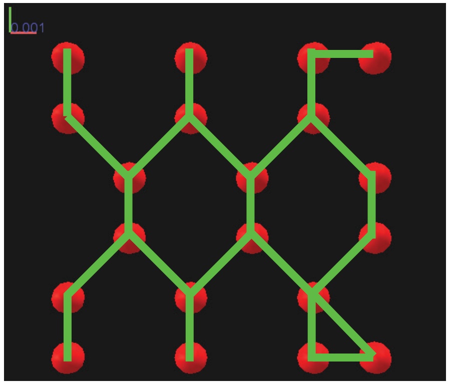
<u>Generation of wires/interactions/links:</u>

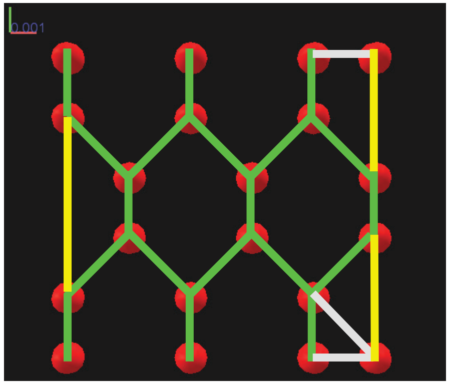
- Automatic definition of interaction via an interaction radius (initialisation step is required)
- Automatic identification of double twist interactions with

$$|n_i - n_j| = 1 \quad \rightarrow \text{double twist}$$

- Interactions at the border need to be modified manually (selvedge wire)



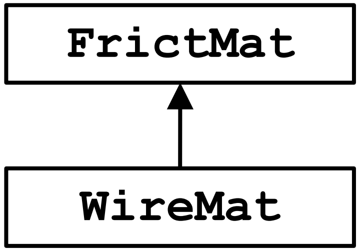




2. The Wire Model in Yade: `WireMat`

Definition of material **`WireMat`**:

```
WireMat( young, poisson,
   frictionAngle, density,
   type, ...,
   )
```



Three different types are implemented:

- **`type=0`**: Bertrand's approach (Bertrand et al. 2005, 2008)
- **`type=1`**: EWM - Elementary wire model (Thoeni et al. 2013)
- **`type=2`**: SDWM - Stochastically distorted wire model (Thoeni et al. 2013)

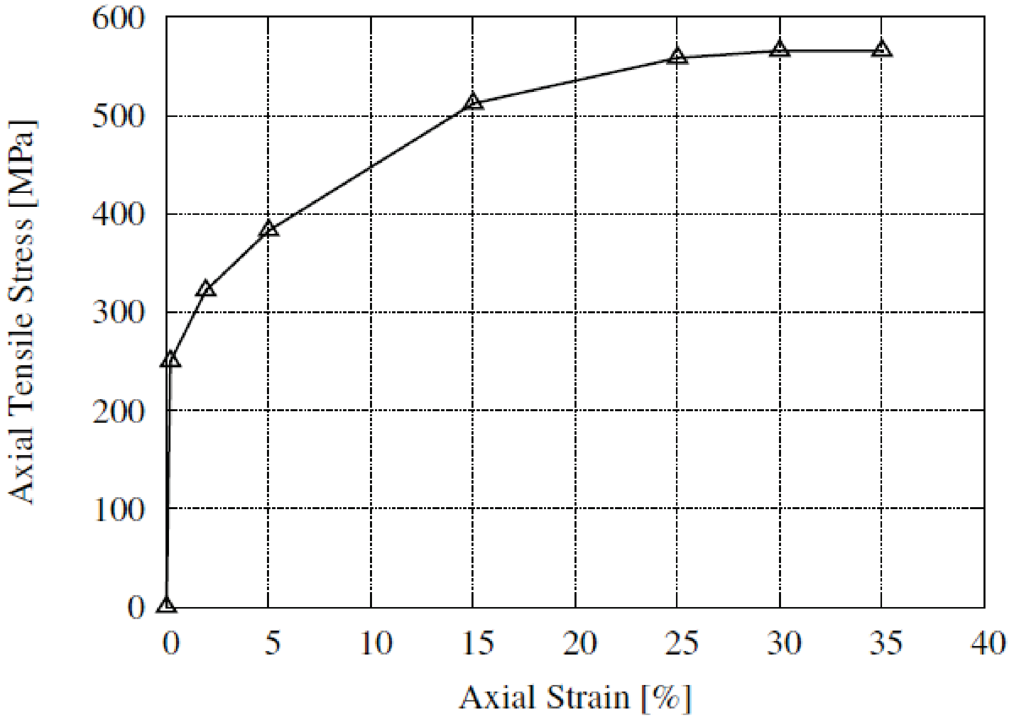
Default: **`type=0`**

2. The Wire Model in Yade: `WireMat(type=0)`

Bertrand's approach (Bertrand et al. 2005, 2008; Thoeni et al. 2011):

```
WireMat( ..., type=0,
   isDoubleTwist, diameter, strainStressValues,
   lambdaEps, lambdak
   )
```

- `strainStressValues`: single wire
- Stress-strain curve of double twist is calculated from calibrated parameters `lambdaEps` and `lambdak`
- Contact law is derived from stress-strain curves

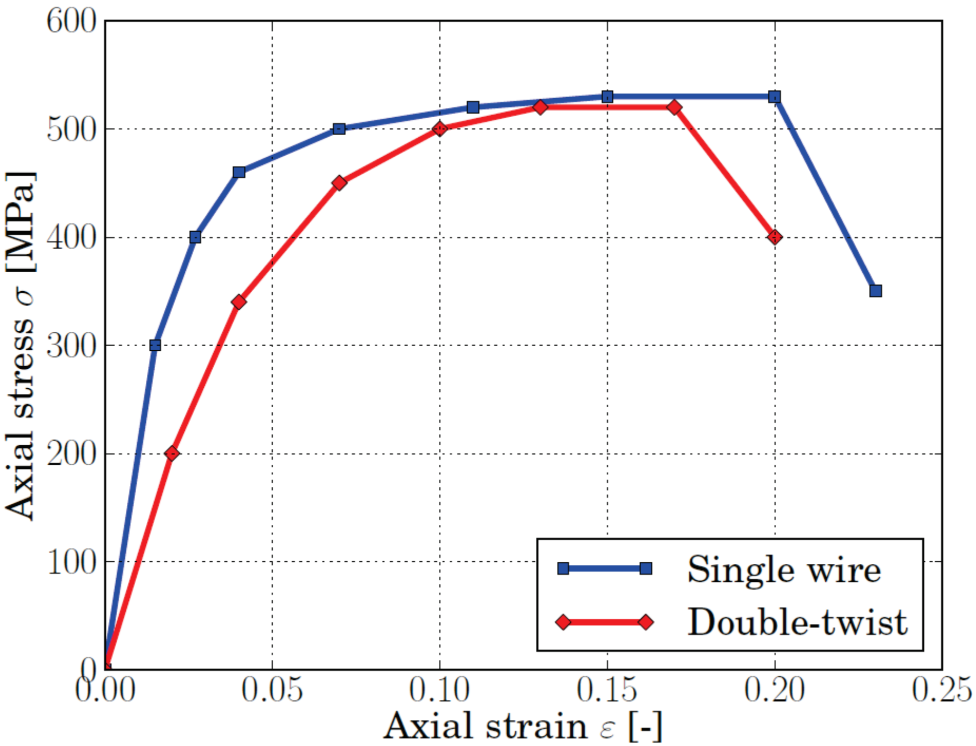


2. The Wire Model in Yade: `WireMat(type=1)`

EWM - Elementary wire model (Thoeni et al. 2013):

```
WireMat( ..., type=1,
    isDoubleTwist, diameter,
    strainStressValues, strainStressValuesDT
    )
```

- Explicitly defined piecewise linear stress-strain curves
- `strainStressValues`: single wire
- `strainStressValuesDT`: double twist
- Contact law is derived from stress-strain curves

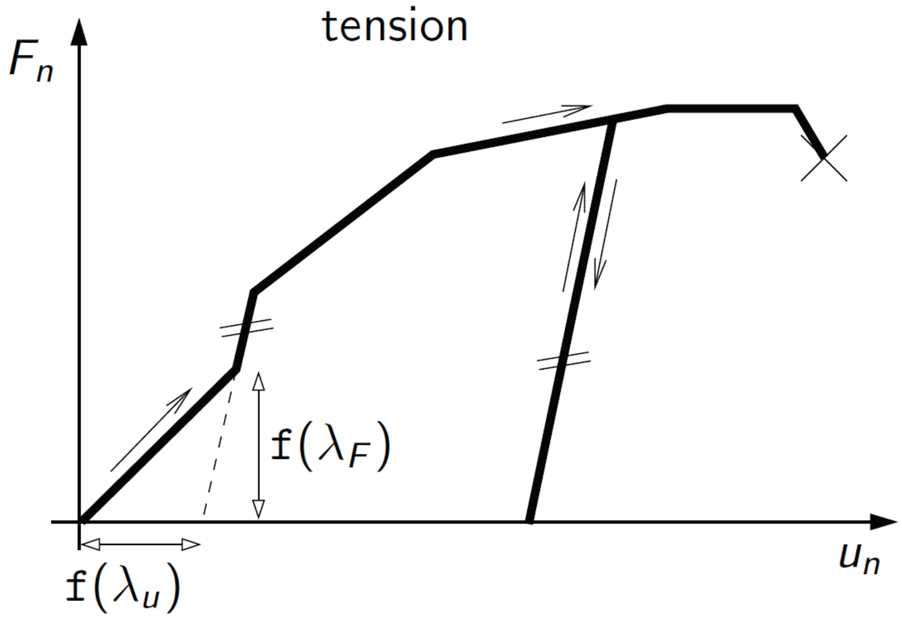


2. The Wire Model in Yade: `WireMat(type=2)`

SDWM - Stochastically distorted wire model (Thoeni et al. 2013):

```
WireMat( ..., type=2,
   isDoubleTwist, diameter,
   strainStressValues, strainStressValuesDT,
   lambdau, lambdaF, seed
   )
```

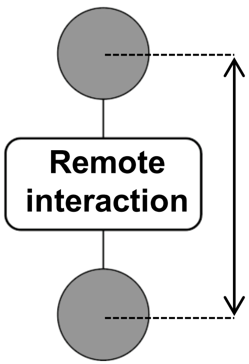
- Stochastically distorted extension of EWM
- Contact law is derived from stress-strain curves
- **`lambdau`**: defines horizontal shift of force-displacement curve
- **`lambdaF`**: defines initial stiffness
- **`seed`**: seed number for initialisation of the random numbers (**`seed=-1`** no distortion, **`seed=0`** internal seed number)



2. The Wire Model in Yade: `WireMat(type=2)`

<u>SDWM - Stochastically distorted wire model (Thoeni et al. 2013):</u>

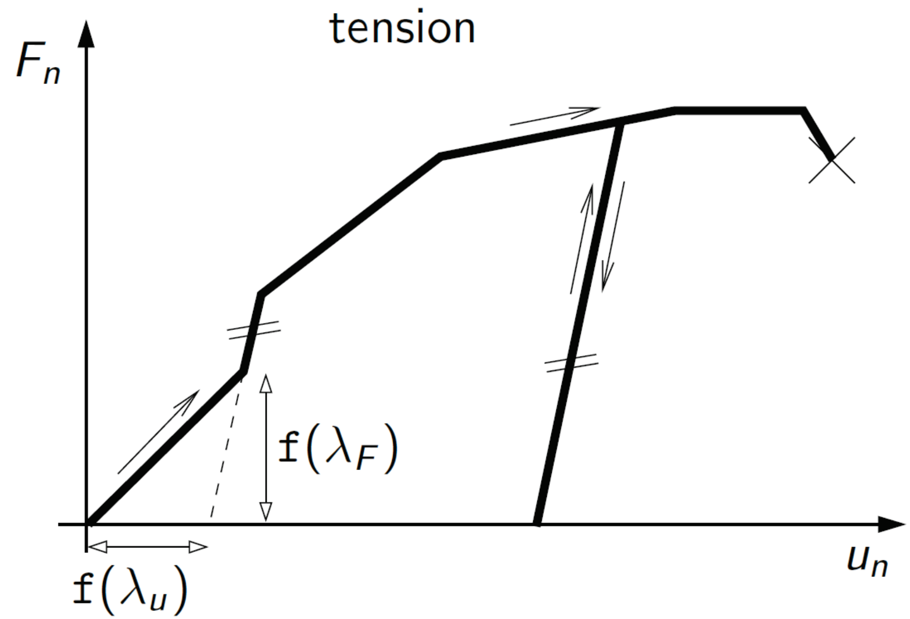
- Stochastically distorted contact model accounts for global mesh characteristics (e.g. distortion of wires and hexagons, variation of Young's modulus)
- Random distortion of the inter-particle length L_0



$$L_0 \rightarrow L = \mathrm{f}(\lambda_u)$$

$$L_0 < L < L_0 + \Delta L$$

$$\Delta L = \lambda_u \tilde{r} L_0 \quad \tilde{r} = 0 \ldots 1$$



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Content

3. How to use

Create the mesh:

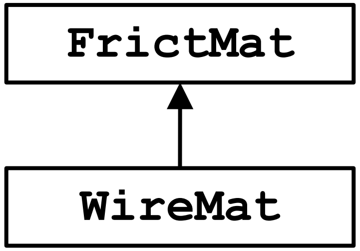
- Define material
- Define engines to initialise the wire interactions

```
iR=2.8
O.engines=[
   ForceResetter(),
   InsertionSortCollider(
   [Bo1_Sphere_Aabb(aabbEnlargeFactor=iR)]
   ),
   InteractionLoop(
   [Ig2_Sphere_Sphere_ScGeom(interactionDetectionFactor=iR)],
   [Ip2_WireMat_WireMat_WirePhys(linkThresholdIteration=1)],
   [Law2_ScGeom_WirePhys_WirePM(linkThresholdIteration=1)]
   ),
]
O.step()
aabb.aabbEnlargeFactor=-1.
Ig2ssGeom.interactionDetectionFactor=-1.
```

3. How to use

<u>Interaction with other particles:</u>

- Interaction on **`FrictMat`** basis




```
O.engines=[
   ForceResetter(),
   InsertionSortCollider(
   [Bo1_Sphere_Aabb()]
   ),
   InteractionLoop(
   [Ig2_Sphere_Sphere_ScGeom()],
   [Ip2_WireMat_WireMat_WirePhys(),
   Ip2_FrictMat_FrictMat_FrictPhys()],
   [Law2_ScGeom_WirePhys_WirePM(),
   Law2_ScGeom_FrictPhys_CundallStrack()]
   ),
   ...
   )]
```

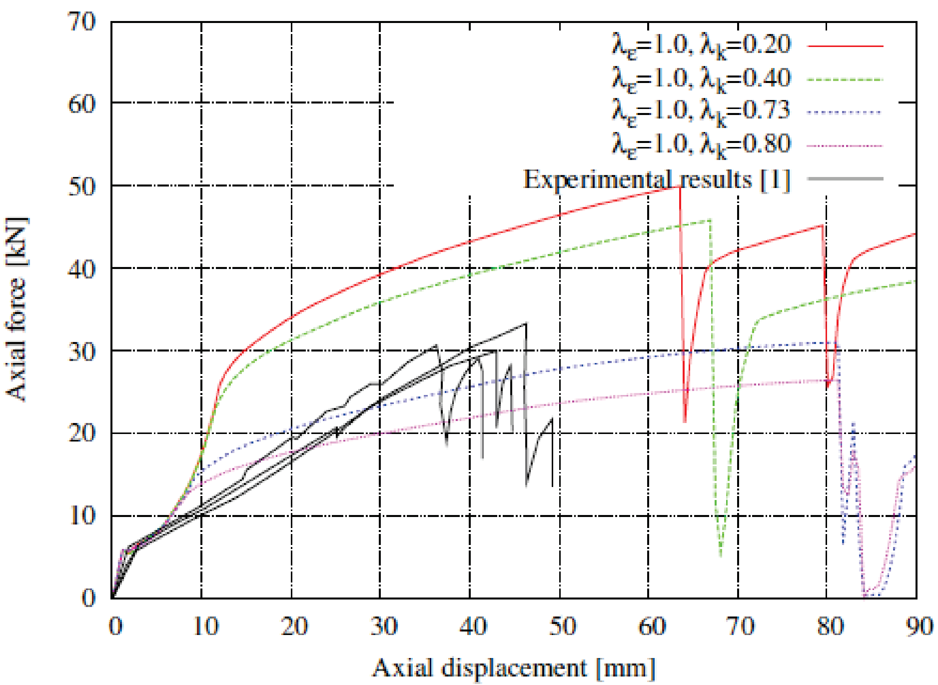
Content

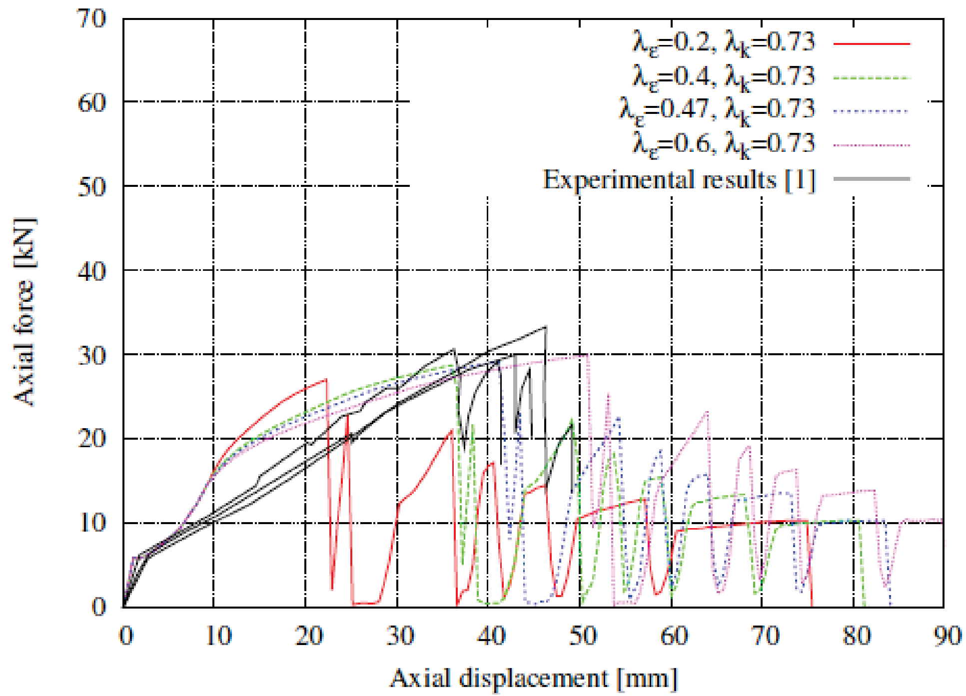
THE UNIVERSITY OF NEWCASTLE AUSTRALIA

4. Some Examples

Tensile test after Bertrand et al. (2008) and Thoeni et al. (2011):

- Tensile test of plane net sheet of 0.5m x 1.0m
- Test is used to calibrate parameters λ_k and λ_ε

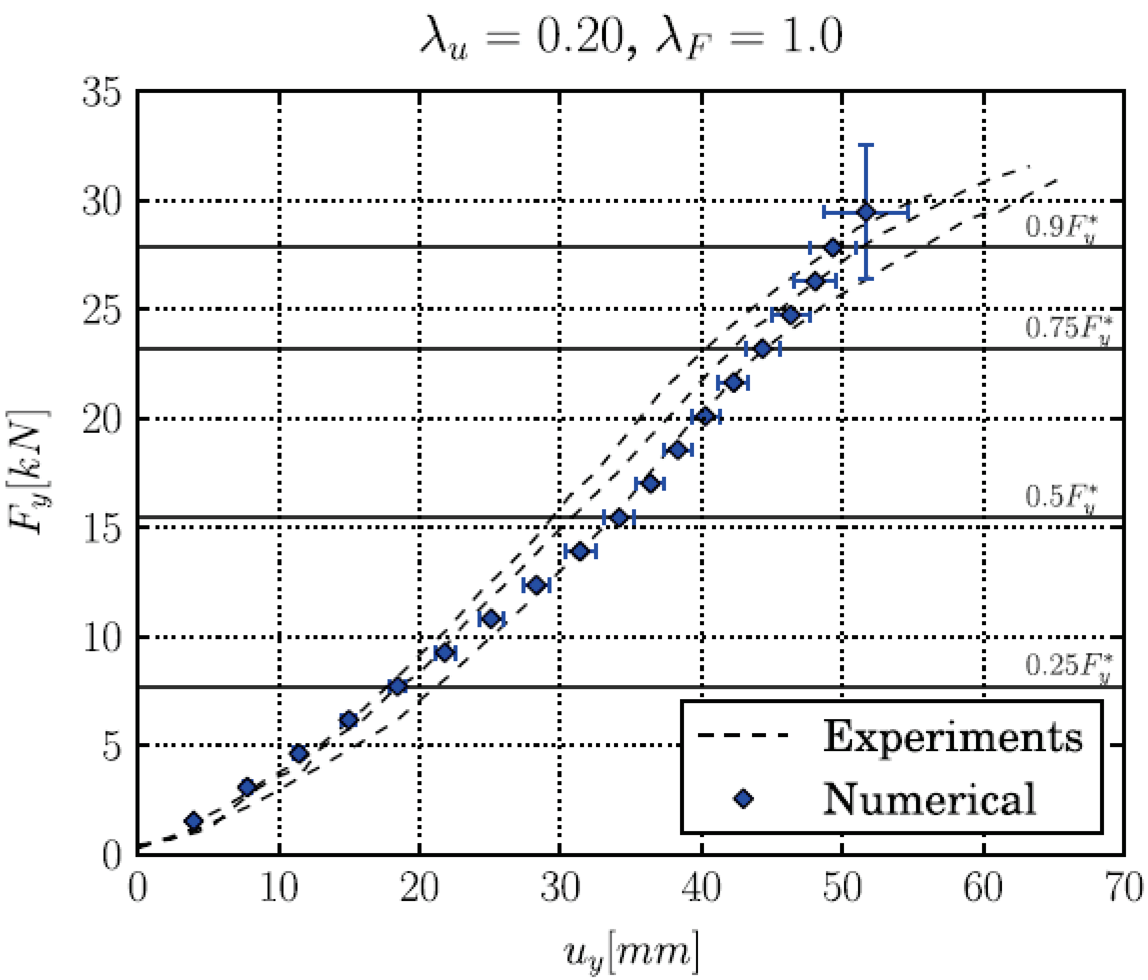


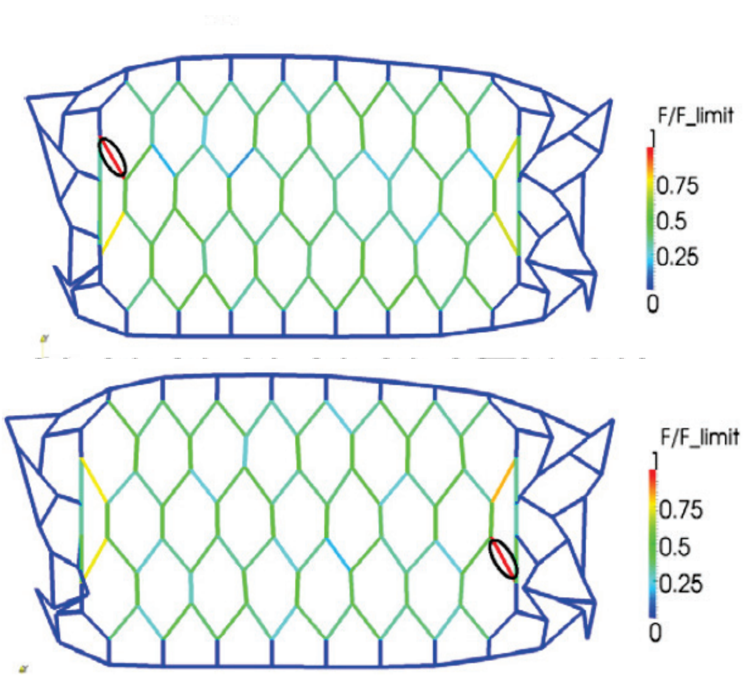


4. Some Examples

Tensile test after Thoeni et al. (2013):

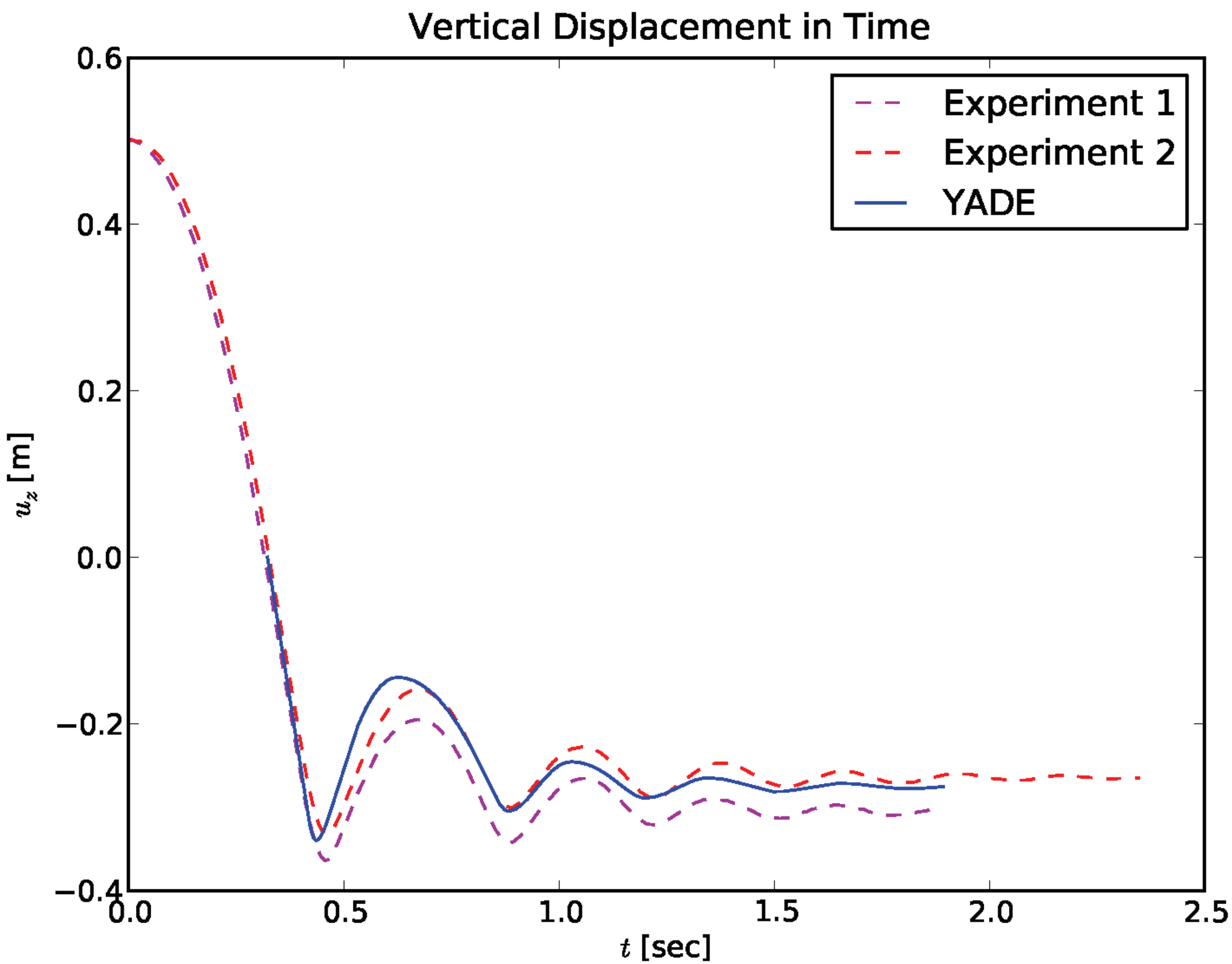
- Tensile test of plane net sheet of 0.36m x 0.95m
- Test is used to calibrate parameters λ_u and λ_F

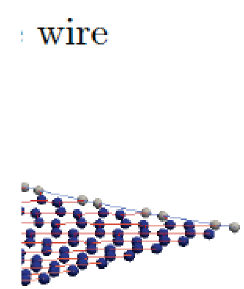




4. Some Examples

Block bouncing on mesh (Thoeni et al. 2013):





Content

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

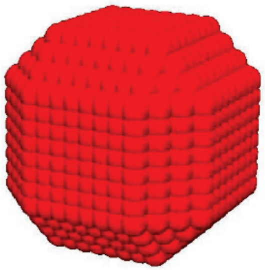
5. An application: Drapery Systems

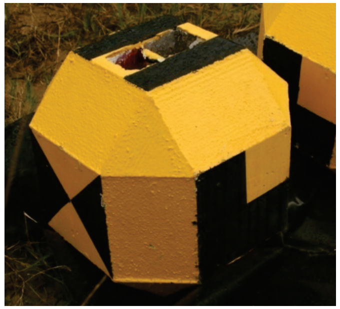
Simple drapery systems for rockfall protection along highwalls:

5. An application: Drapery Systems

Representation of the block:

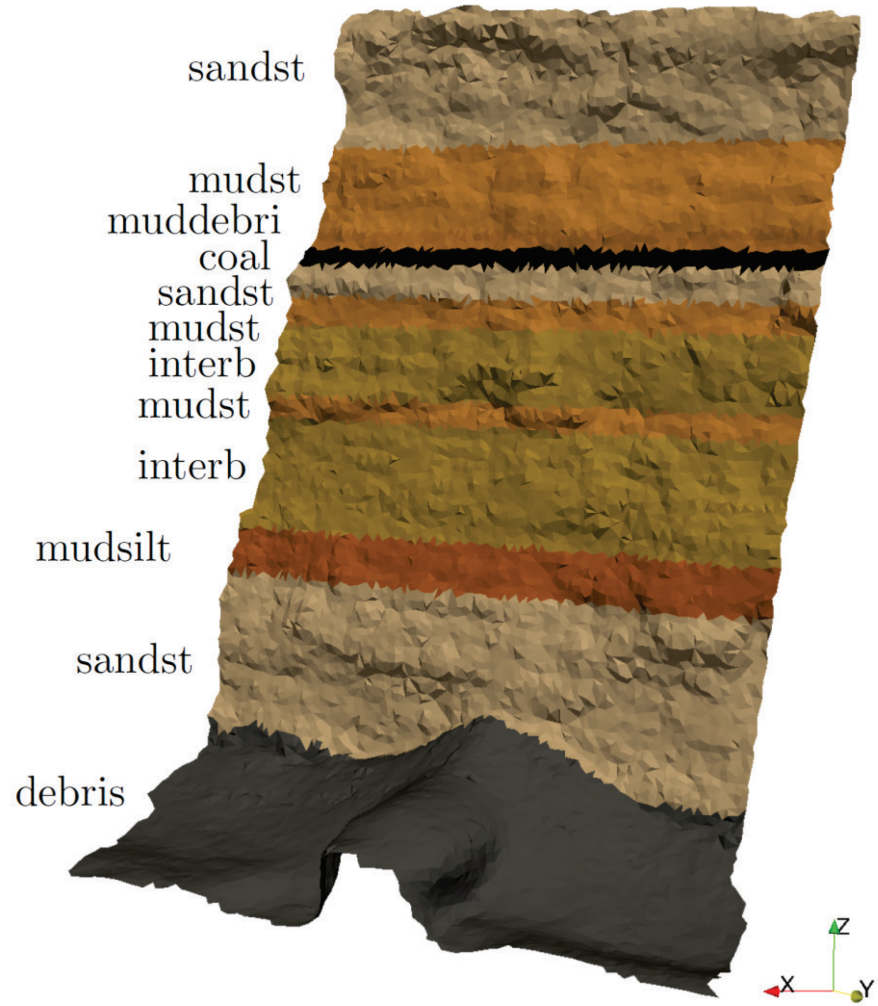
- Rigid clumps of spherical particle





Representation of the slope:

- Facets
- Identification of the material layers
- Material properties (e.g. restitution coefficients) according to geological survey



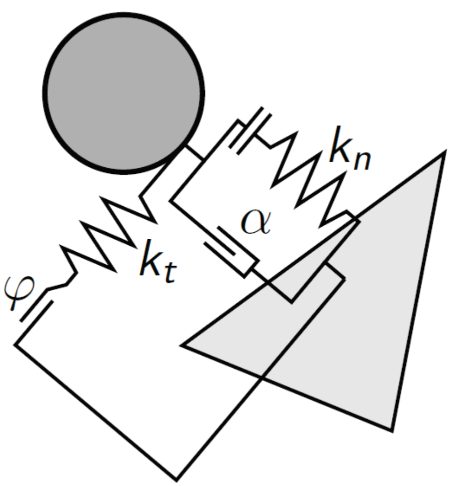
5. An application: Drapery Systems

<u>Contact model for non wire-wire interactions:</u>

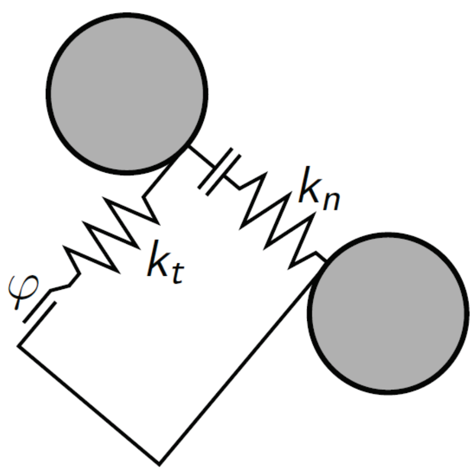
- Classical linear elastic-plastic law with friction and viscous damping (**`FrictViscoPM`**)

$$F_n = k_n\, u_n + \alpha \frac{\mathrm{d}u_n}{\mathrm{d}t} \qquad \alpha = 2\sqrt{k_n m}\, \frac{-\ln e_n}{\sqrt{\pi^2 + \ln^2 e_n}}$$

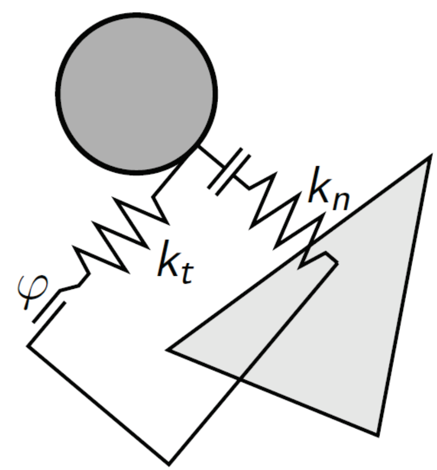
$$\Delta F_t = k_t\, \Delta u_t \qquad |F_t| \leq |F_n| \tan\varphi$$



Block-Slope



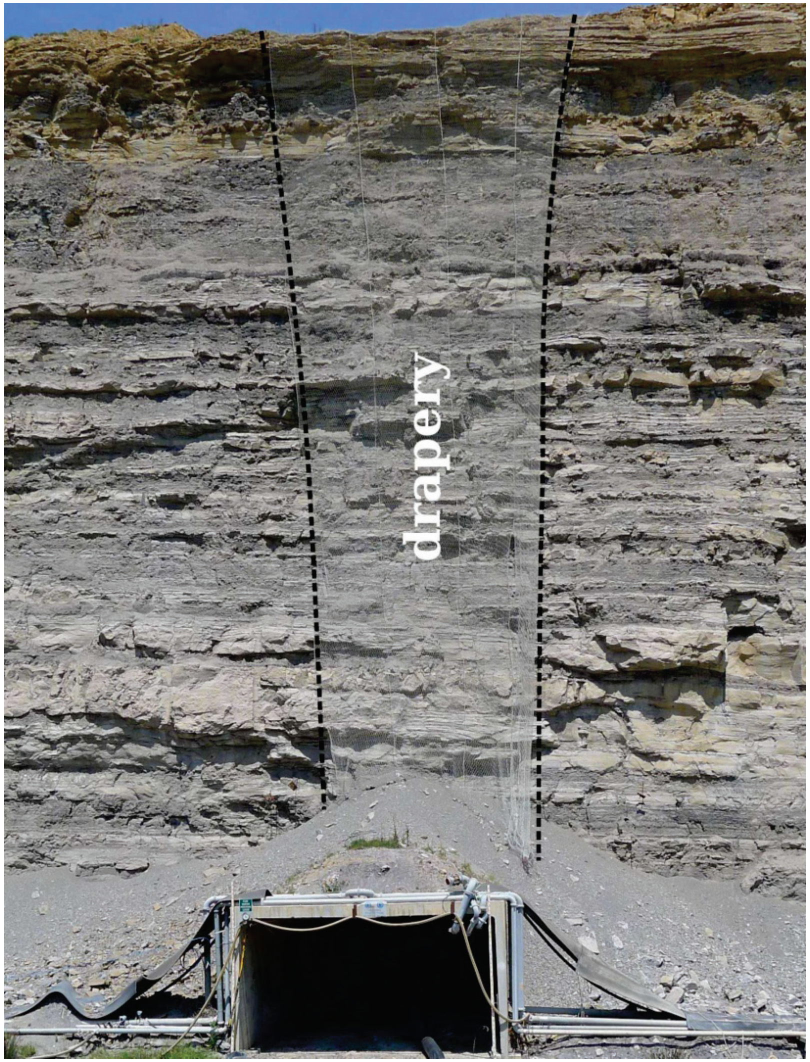
Block-Drapery



Slope-Drapery

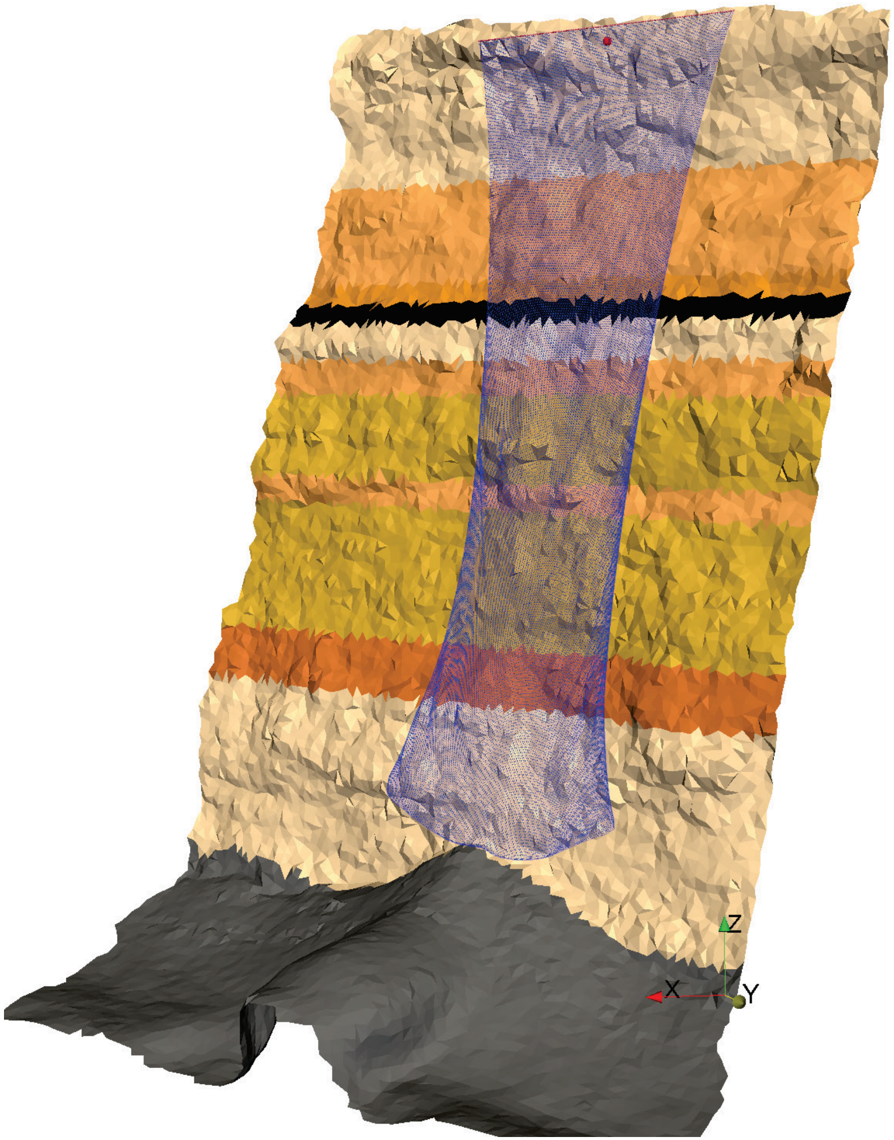
5. An application: Drapery Systems

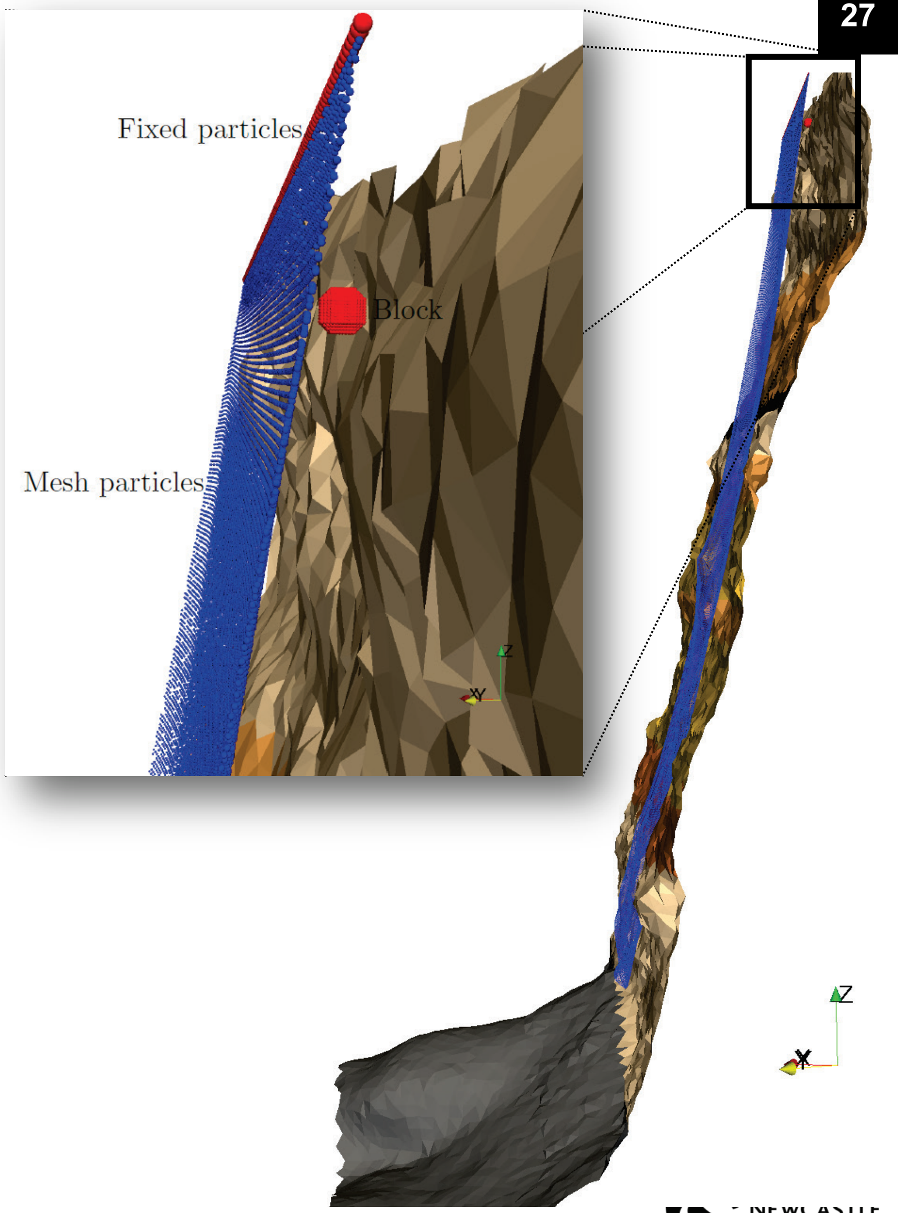
Experimental tests (Giacomini et al., 2012):



5. An application: Drapery Systems

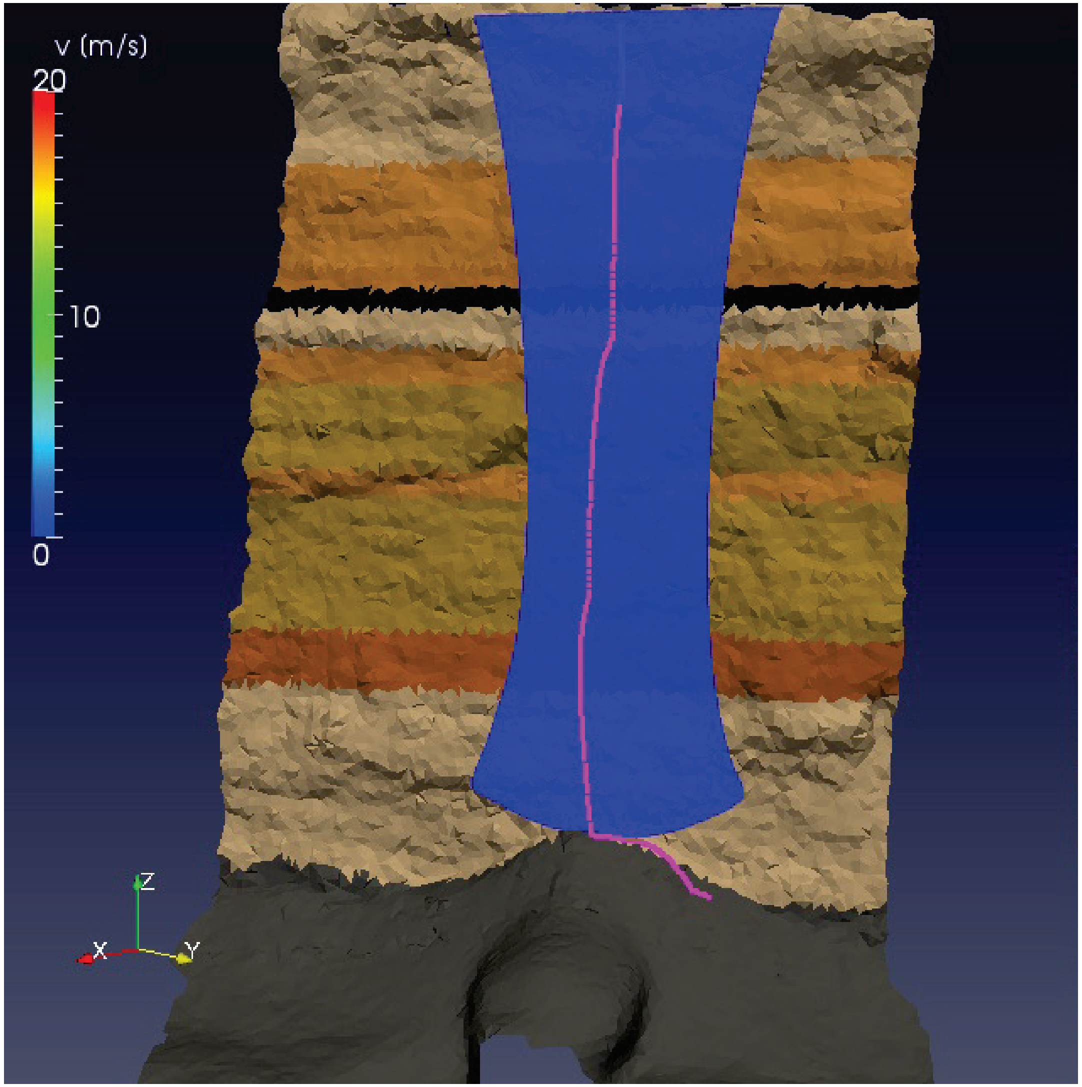
Representation of full model:





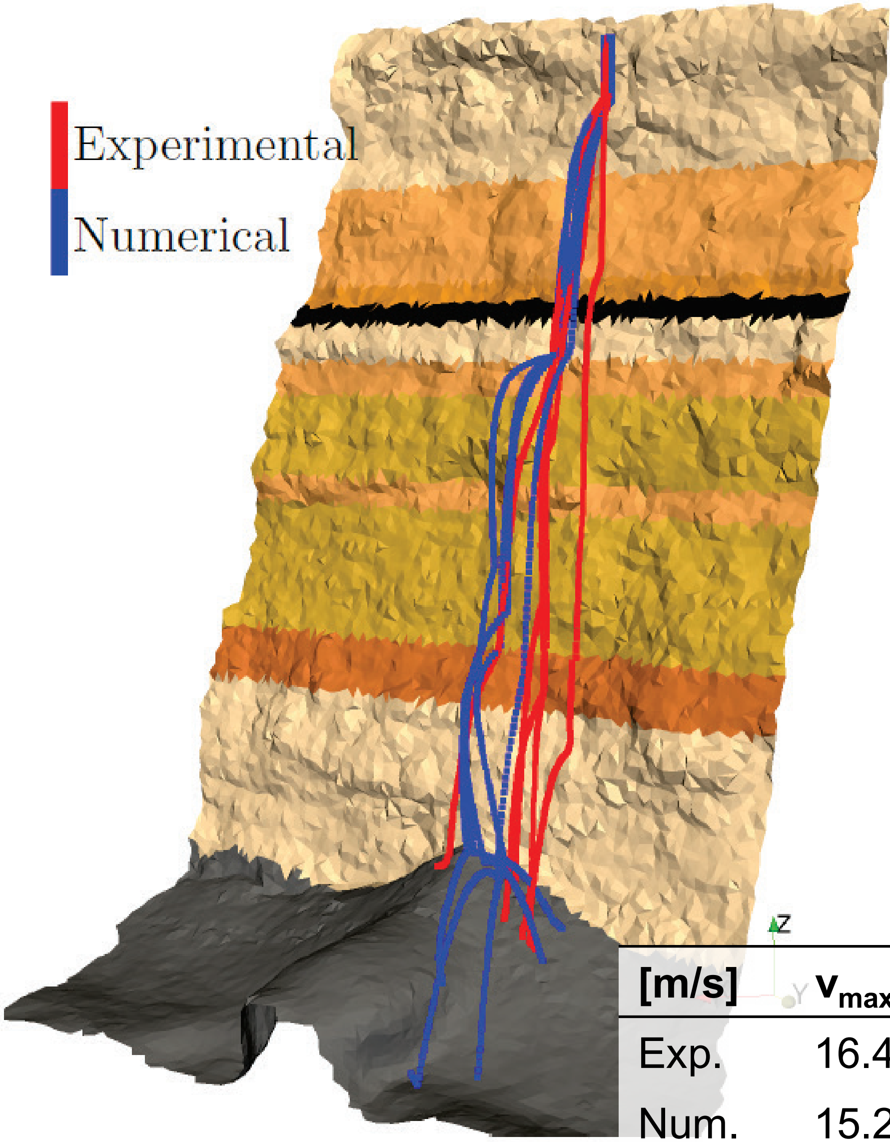
5. An application: Drapery Systems

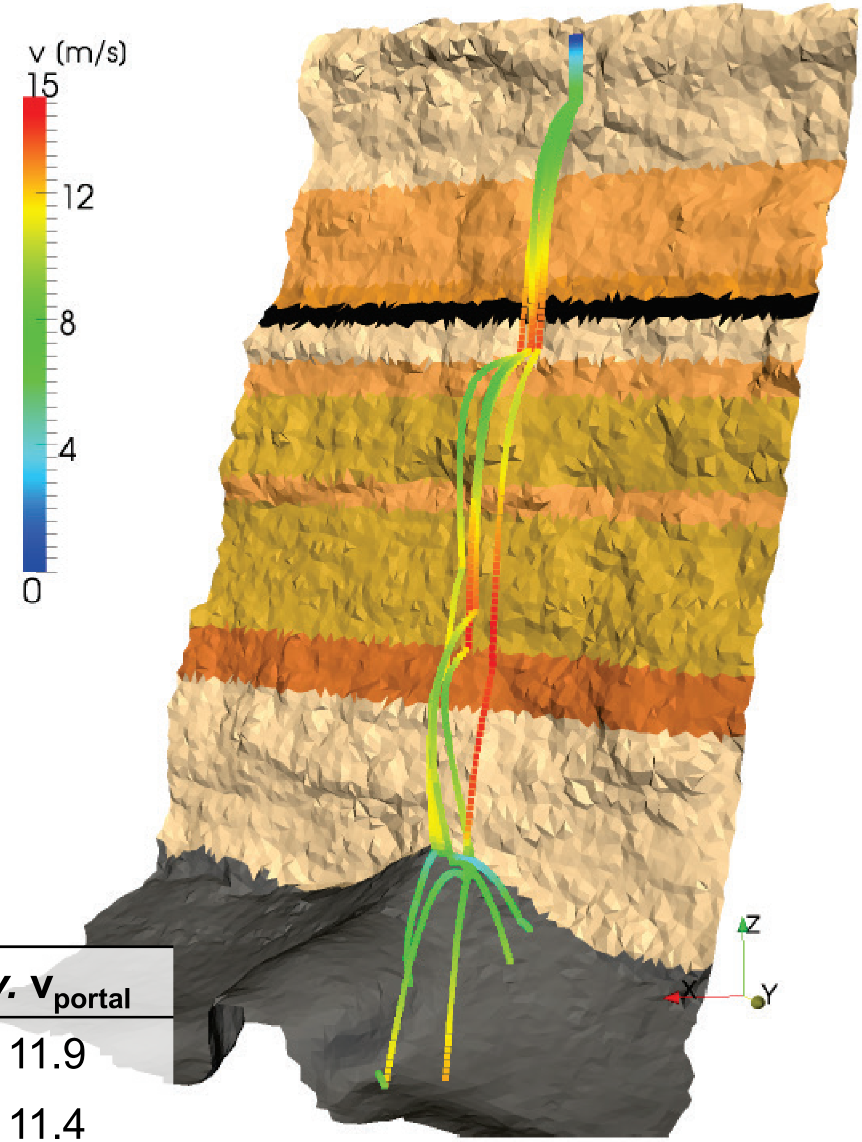
Video experimental test vs. numerical prediction:



5. An application: Drapery Systems

Comparison between experimental results and numerical predictions:



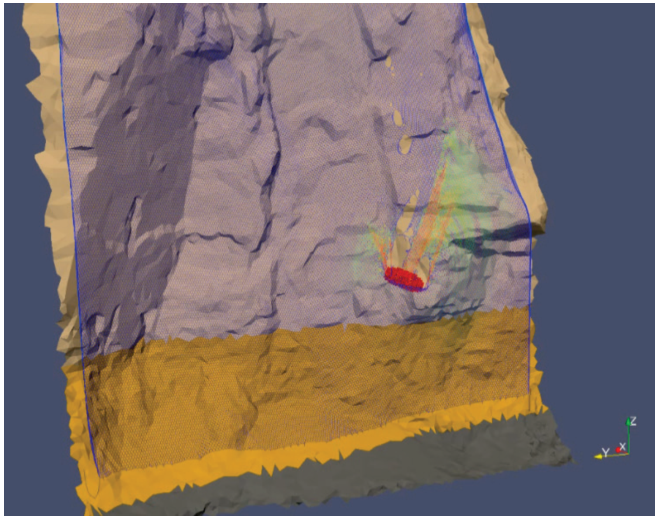


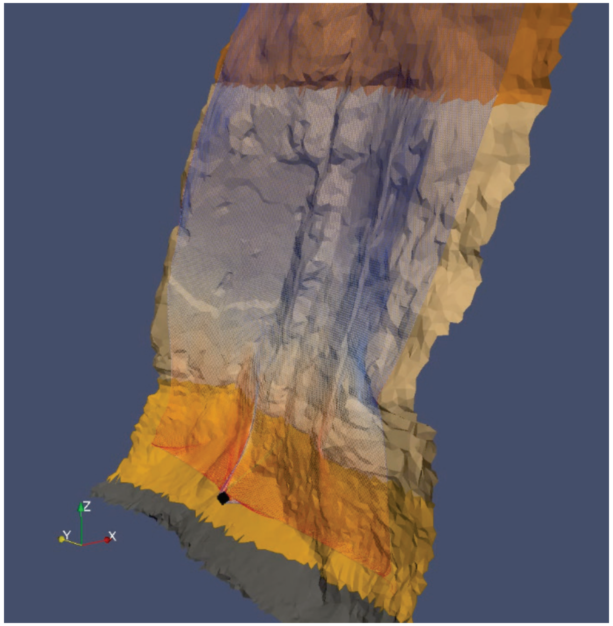
[m/s]	v_{max}	*av.* v_{portal}
Exp.	16.4	11.9
Num.	15.2	11.4

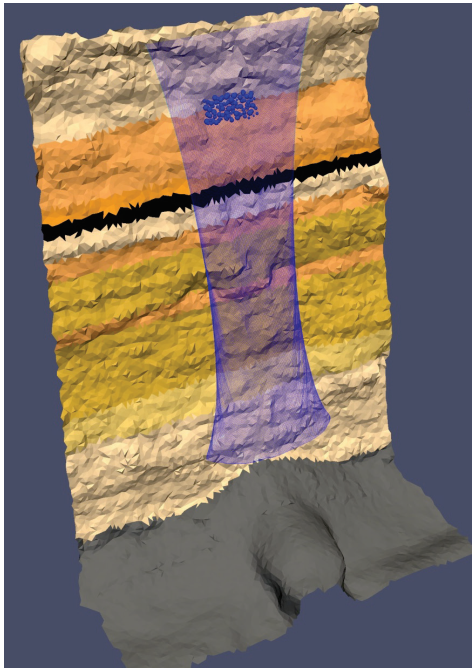
5. An application: Drapery Systems

Some capabilities of the model:

- Mesh failure
- Integration of wire ropes
- Multiple falling blocks







Content

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

7. Future Developments

- Implementation of particle-particle contact for wire particles
- Better integration of contacts between various materials / different wires
- Better integration of wire ropes
- Implementation of more advanced block-slope impact models
- Application to other types of meshes

Bibliography

- Bertrand, D., Nicot, F., Gotteland, P., Lambert, S.: **Discrete element method (DEM) numerical modeling of double-twisted hexagonal mesh**, *Canadian Geotechnical Journal* 45(8), 1104–1117, 2008
- Bertrand, D., Nicot, F., Gotteland, P., and Lambert, S.: **Modelling a geo-composite cell using discrete analysis**. *Computers and Geotechnics*, 32(8):564–577, 2005.
- Giacomini, A., Thoeni, K., Lambert, C., Booth, S., Sloan, S.W.: **Experimental study on rockfall drapery systems for open pit highwalls**, *International Journal of Rock Mechanics and Mining Sciences* 56, 171–181, 2012
- Nicot, F., Cambou, B., Mazzoleni, G.: **Design of Rockfall Restraining Nets from a Discrete Element Modelling**, *Rock Mechanics and Rock Engineering* 34, 99–118, 2001
- Thoeni, K., Lambert, C., Giacomini, A., and Sloan, S.: **Discrete Modelling of a Rockfall Protective System**. In Oñate, E. and Owen, D., editors, *Particle-Based Methods II: Fundamentals and Applications, II International Conference on Particle-based Methods*, page 24–32. CIMNE International Center for Numerical Methods in Engineering, 2011.
- Thoeni, K., Lambert, C., Giacomini, A., Sloan, S.W.: **Discrete modelling of hexagonal wire meshes with a stochastically distorted contact model**, *Computers and Geotechnics* 49, 158–169, 2013
- Thoeni, K., Giacomini, A., Lambert, C., Sloan, S.W., Carter, J.P.: **A 3D discrete element modelling approach for rockfall analysis with drapery systems**, *International Journal of Rock Mechanics and Mining Sciences*, 2014

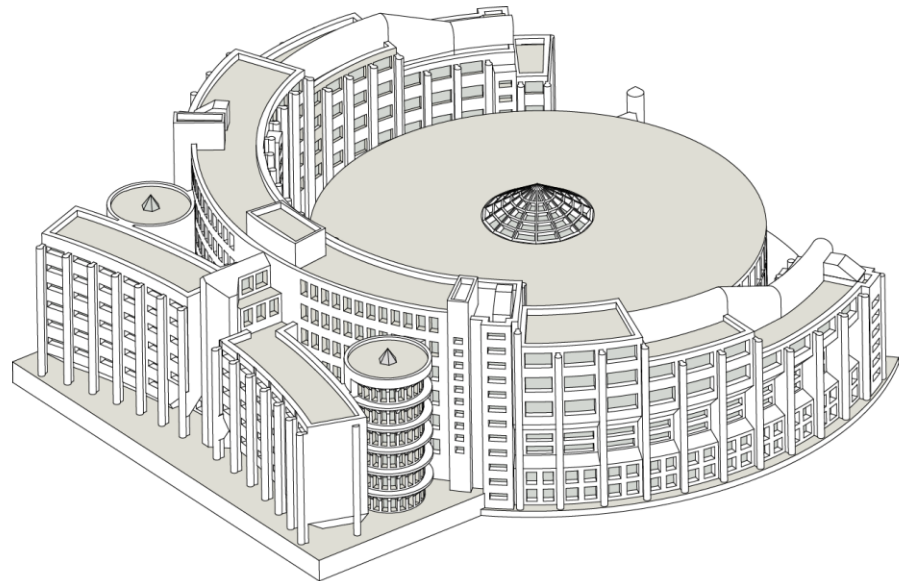
Modeling of Mass Finishing at the IWF

Production Technology Center Berlin
Prof. Dr. h. c. Dr.-Ing. Eckart Uhlmann

M. Sc. Alexander Eulitz

Institute for Maschine Tools and Factory Management
Technische Universität Berlin

Fraunhofer-Institute Production Systems and Design Technology



Agenda

- Production Technology Center
- Introduction to Mass Finishing
- DEM Modeling of Mass Finishing
- Summary and Outlook

Production Technology Center

Robot Cell for Finishing Operations at IWF, TU Berlin

Production Technology Center - Organization

Fraunhofer-Gesellschaft
Institute for
Production Systems and Design Technology (IPK)

Corporate Management
Temporary Administration: Dr.-Ing. H. Kohl

Virtual Product Creation
Prof. Dr.-Ing. R. Stark

Production Systems
Prof. Dr. h. c. Dr.-Ing. E. Uhlmann

Automation Technology
Prof. Dr.-Ing. J. Krüger

Joining and Coating Technology
Prof. Dr.-Ing. Michael Rethmeier

Quality Management
Prof. Dr.-Ing. R. Jochem

Medical Technology
Prof. Dr.-Ing. E. Keeve

University of Technology Berlin
Institute for
Machine Tools and Factory Management (IWF)

Assembly Technology and Factory Management
Prof. Dr.-Ing. G. Seliger

Industrial Information Technology
Prof. Dr.-Ing. R. Stark

Machine Tools and Manufacturing Technology
Prof. Dr. h. c. Dr.-Ing. E. Uhlmann

Industrial Automation Technology
Prof. Dr.-Ing. J. Krüger

Joining and Coating Technology
Temporary Administration: Prof. Dr.-Ing. R. Stark

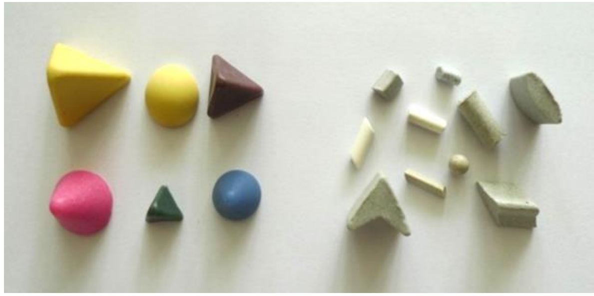
Quality Science
Prof. Dr.-Ing. R. Jochem

Micro Production Technology
Prof. Dr. h. c. Dr.-Ing. E. Uhlmann

Introduction to Mass Finishing

Process Fundamentals

- Process based on an relative motion between loose abrasive media and workpieces
- Workpiece movement can be controlled (e. g. drag finishing) or guided by the media flow (e. g. vibratory finishing)
- Widely used technology for surface roughness improvement, deburring and edge rounding



Selection of abrasive media

Car rim, BBS

Robot guided drag finishing and vibratory finishing facilities at IWF, TU Berlin

Introduction to Mass Finishing

Process Fundamentals – Video (non-vibrating media)

Video

Robot guided workpiece in non-vibrating media

Introduction to Mass Finishing

Process Fundamentals – Video (vibrating media)

Video

Movement of vibrating media (no lubrication)

Video

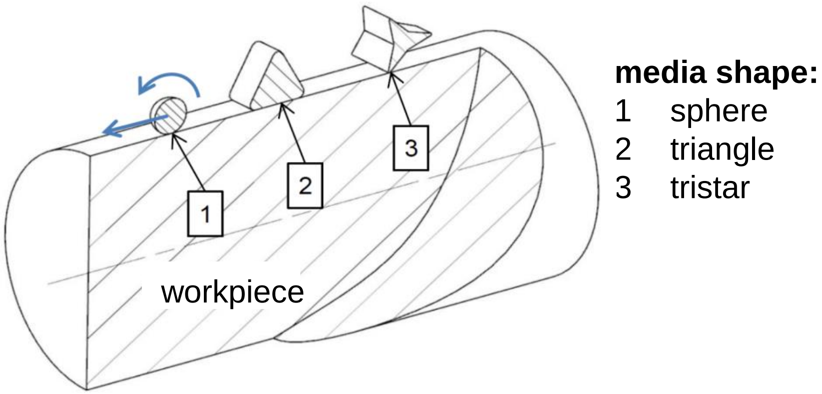
Robot guided workpiece in vibrating lubricated media

Introduction to Mass Finishing

Research Focus

- Fundamental mechanisms of surface evolution
- Development of empirical model to predict surface roughness
- Modeling of the workpiece-media interaction with the Discrete Element Method (DEM)

Smoothed impeller [Walther Trowal GmbH & Co. KG]



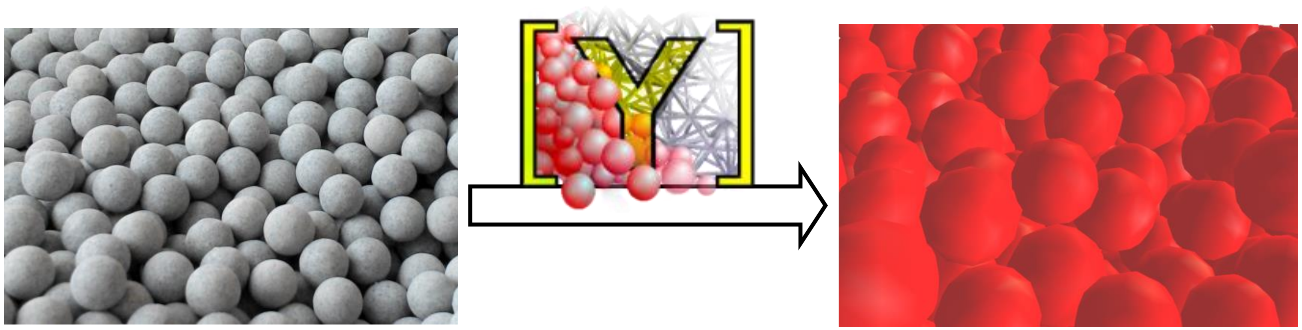
Contact between workpiece and media

Fitting before and after vibratory finishing [Walther Trowal GmbH & Co. KG]

DEM Modeling of Mass Finishing

Motivation

- Up to now process design is very costly because of trial-and-error runs
- Lack of fundamental knowledge about workpiece-media interations
- Goal: simulate number, type and intensity of contacts for finite areas of any workpiece



Real spherical media

Modeled spherical media

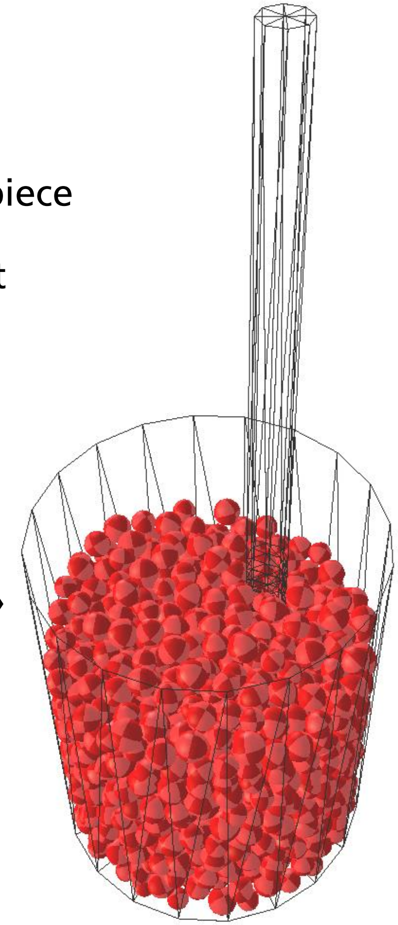
DEM Modeling of Mass Finishing

Modeling Approach 1/2

- Goal: simulate number, type and intensity of contacts for finite areas of any workpiece
- Modeling of a scaled simplified process with spherical because of calculation effort

Real process

Scaled simplified process



Modeled simplified scaled process

DEM Modeling of Mass Finishing

Modeling Approach 2/2

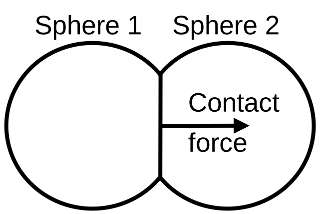
formulation → calibration → validation → utilization

formulation
- boundary conditions
- contact law

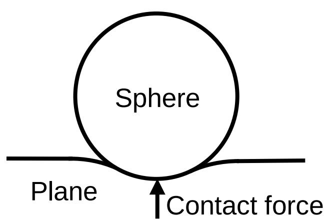
calibration
- material properties
- damping

validation

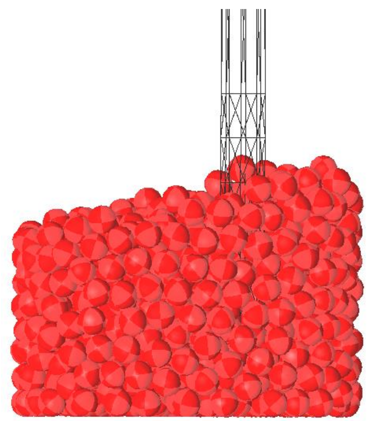
utilization
- determination of local contact intensities at workpiece surface



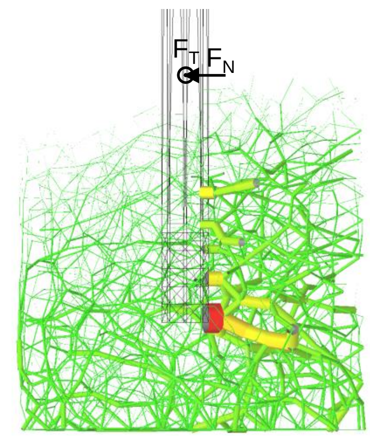
Sphere-sphere contact



Sphere-plane contact



Modeled drag finishing process

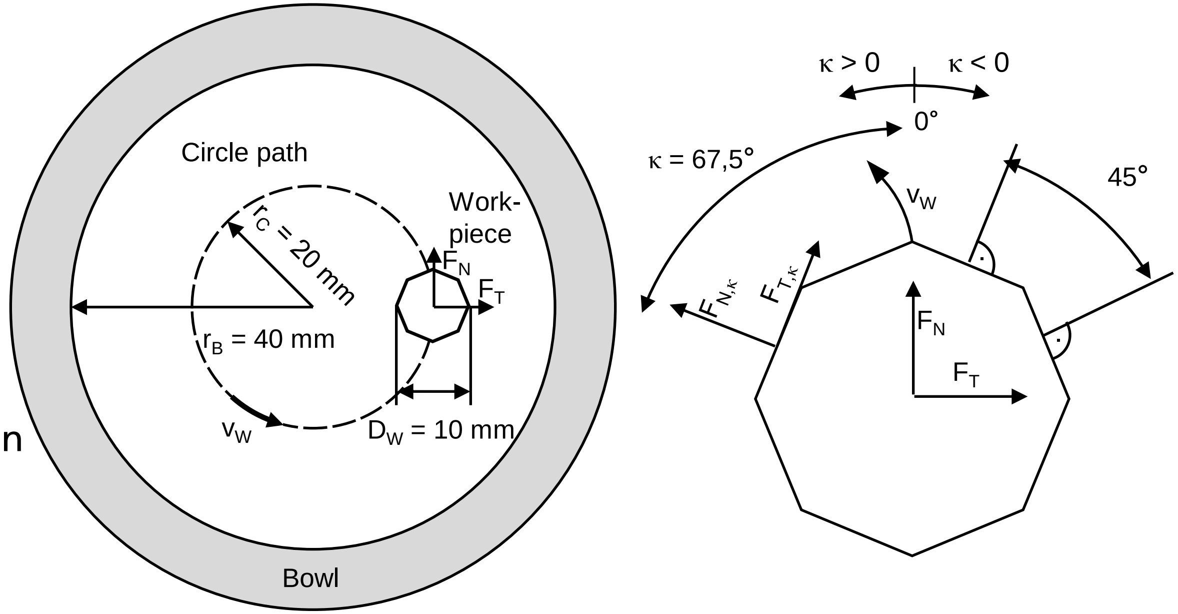


Simulated contact force network for drag finishing

DEM Modeling of Mass Finishing

Model Formulation 1/2

- Media is considered as a bulk of discrete, purely elastic particles
- The dimensions of bowl and workpiece are scaled, boundary conditions are implemented as facets
- Contact forces are calculated according to the non-linear, simplified Hertz-Mindlin contact force model



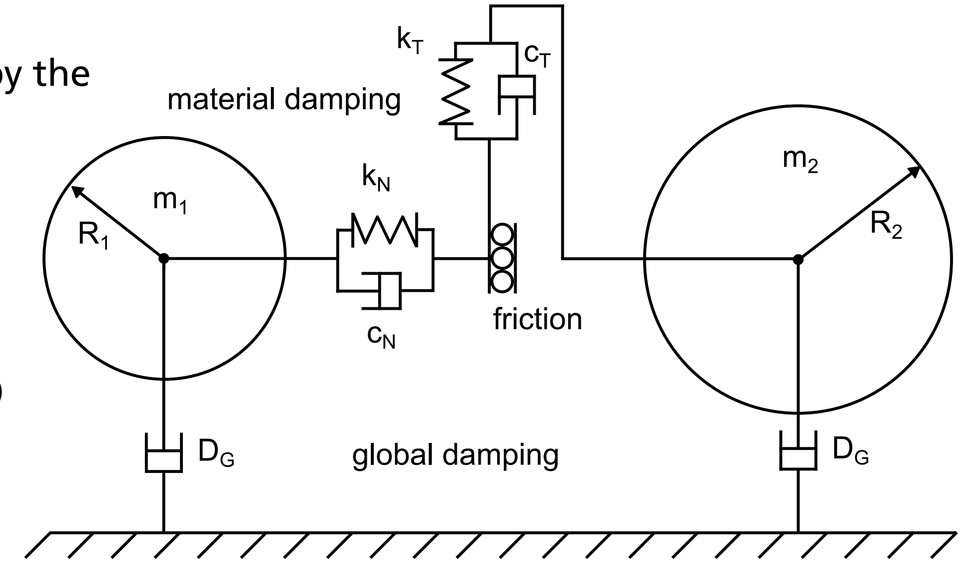
D_W: diameter of the workpiece
F_T : tangential force of the workpiece
r_B : radius of the bowl
v_W : workpiece speed
F_N : normal force of the workpiece
κ : orientation of facets to feed direction
r_C : radius of circle path

Geometric setup of the DEM-model

DEM Modeling of Mass Finishing

Model Formulation 2/2

- Contact forces calculated according to the non-linear, simplified Hertz-Mindlin contact force model
- Force transmitted in tangential direction is limited by the Mohr-Coulomb friction law (plasticity criterion)
- Energy is dissipated through:
  - global damping of the absolute velocities of particles
  - material damping (viscous damping at contacts)
  - friction between particles and particles-facets



Schematic representation of simplified Hertz-Mindlin contact force model

DEM Modeling of Mass Finishing

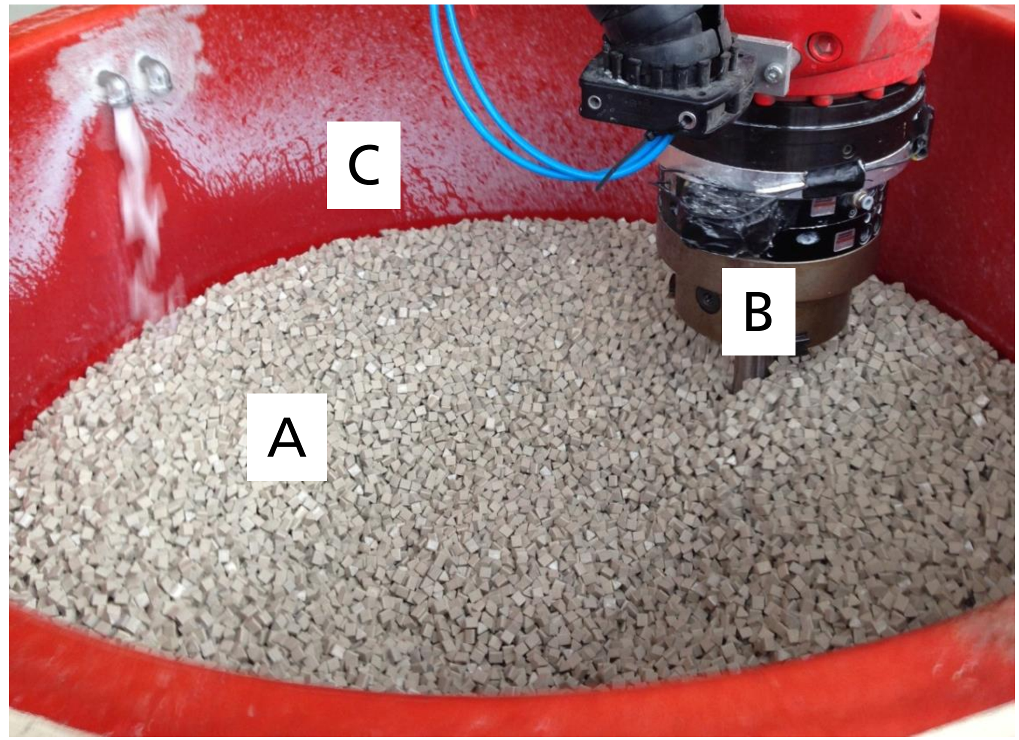
Model Calibration

- Unknown material properties have to be determined
- Calibration approaches
  - Physical (micro-scale):
    determine model parameters according to well-known mechanical relationships on a per-particle base (e.g. Hook's law) → Tensile Test, Compression Test
  - Phenomenological (macro-scale):
    adjust model parameters until certain macroscopic behavior of granular media is observed (e.g. modulus of dense granular media) → Triaxial test, Oedometric Test, Shear Test

DEM Modeling of Mass Finishing

Model Calibration - Materials and Material Combinations

- Three occurring materials:
  - A: media -> ceramic or resin-bound abrasives
  - B: rod -> plastics, steel, ceramic, …
  - C: bowl -> polyurethane coated steel
- Three occurring material combinations:
  - A-A: low relative velocities, mixture of static and dynamic friction
  - A-B: high relative velocities
  - A-C: medium relative velocities



DEM Modeling of Mass Finishing

Model Calibration - Parameters

- Material parameters as input for the contact model:
  - Density ρ_i
  - Young's modulus E_i
  - Poisson's ratio ν_i
  - Friction coefficients $\mu_{i,j}$ for all materials i and material combinations i, j
  - Material damping (coefficient of normal restitution e_N)
- Global damping parameter d (NewtonIntegrator)
- Timestep Δt

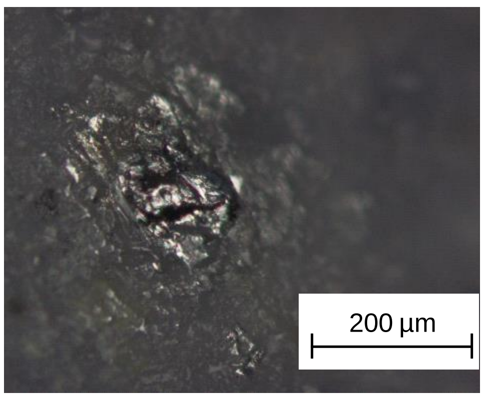
Determined by :

- Physical tests (density, Young's modulus, Poisson's ratio, friction coefficients, material damping)
- Phenomenological tests (material damping, global damping parameter)
- PWaveTimeStepper (timestep)

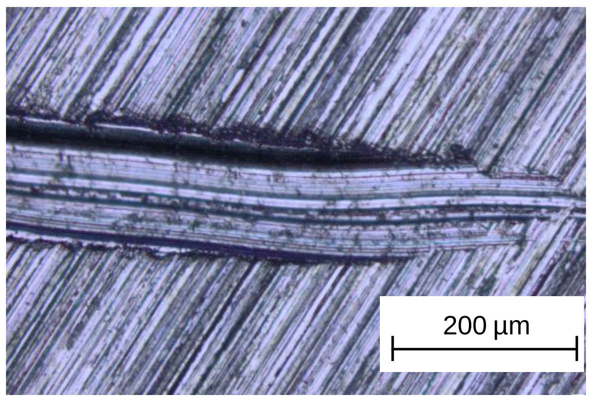
DEM Modeling of Mass Finishing

Model Calibration - Friction Coefficients

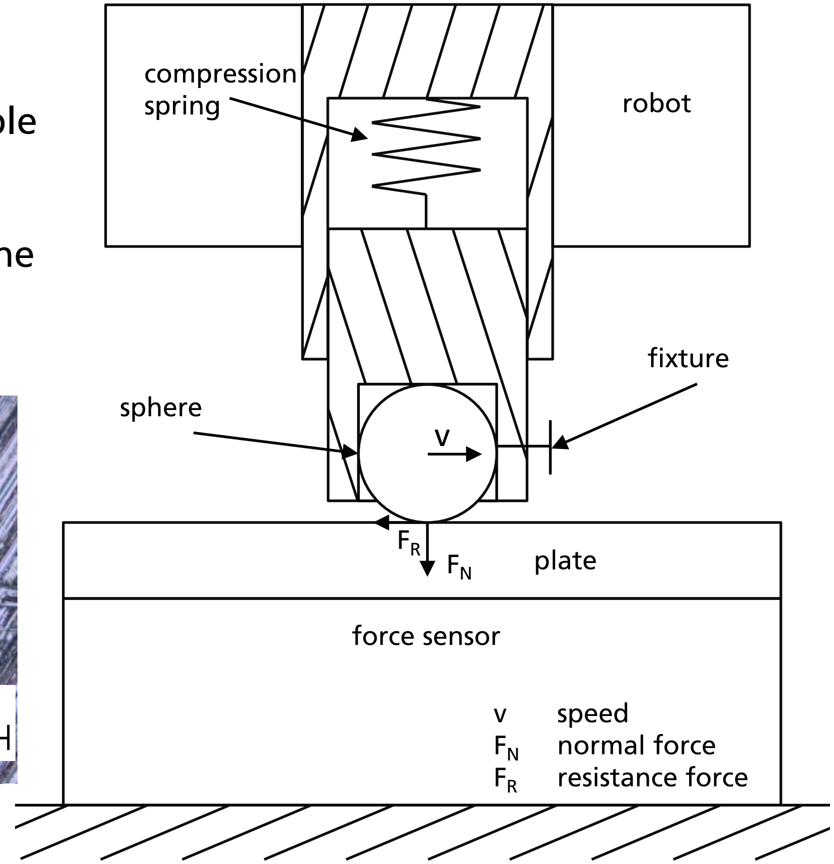
- Friction coefficients $\mu_{i,j}$ have to be determined for all possible combinations of materials in contact
- Conditions during calibration should match conditions of the modeled object (e.g. dry vs. lubricated contact)



Surface of sphere after scratching



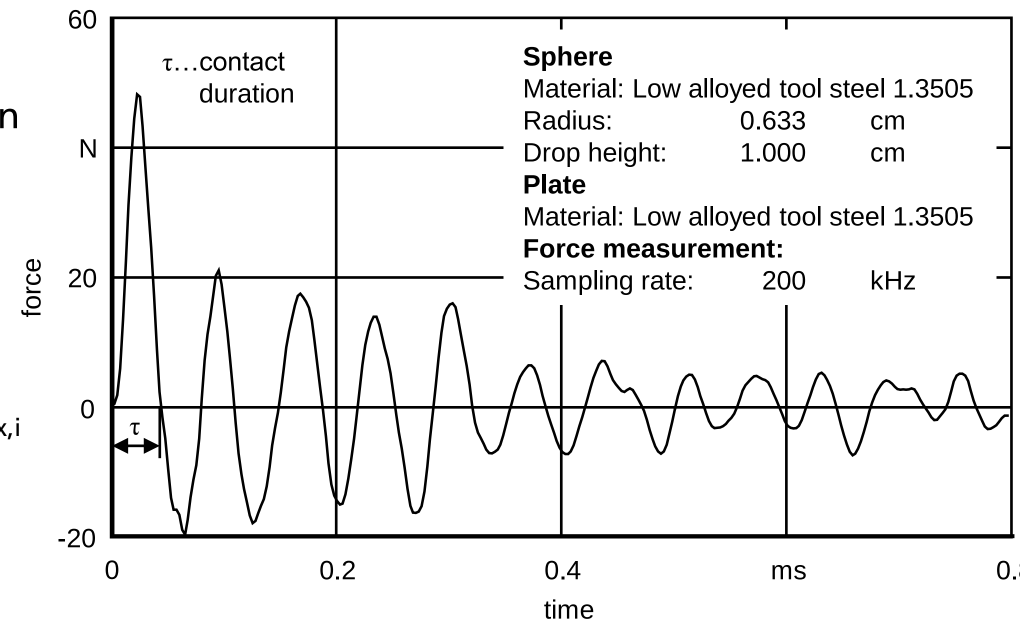
Scratch on the plate



DEM Modeling of Mass Finishing

Model Calibration - Damping and Restitution

- Interdependency between global damping and material damping -> simultaneous determination
- Investigation of single contacts in sphere drop experiments in terms of:
  - Contact duration τ
  - Maximum force magnitude per impact $F_{max,i}$
  - Time between impacts Δt_i



Recorded force signal of first impact during the drop of a low alloyed tool steel 1.3505 from 1cm height

DEM Modeling of Mass Finishing

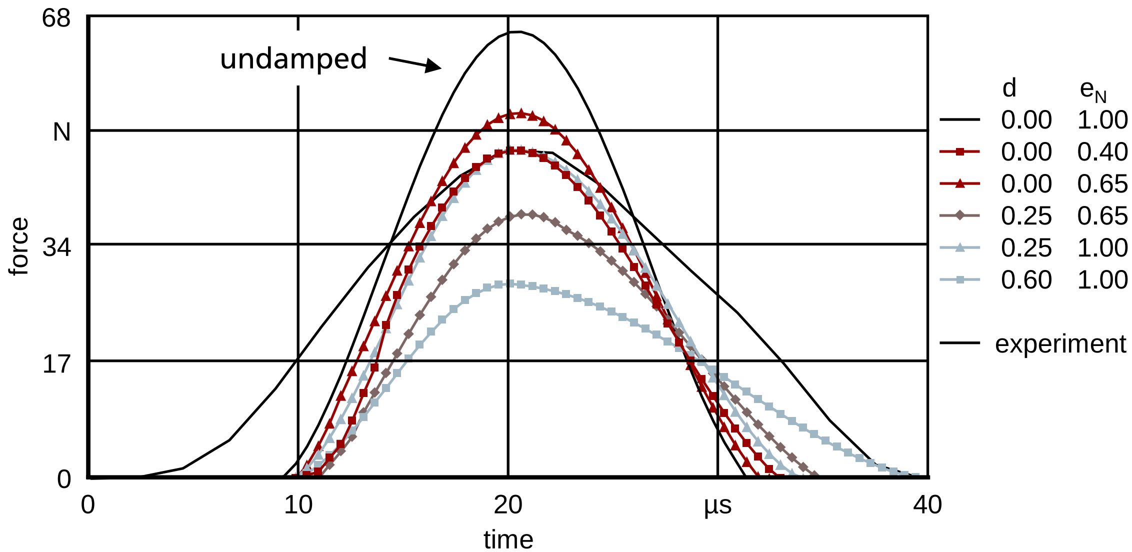
Model Calibration - Damping and Restitution

- "Best" combination of global and material damping is obtained by a weighted linear optimization algorithm
- Force measurement systems with high sampling rates and accuracy are required

Sphere
Material: Low alloyed tool steel 1.3505
Radius: 0.633 cm
Drop height: 1.000 cm
Young's modulus: 210 GPa
Poisson's ratio: 0.3
Density 7900 kg/m³

Plate
Material: Low alloyed tool steel 1.3505

Simulation
Hertz-Mindlin no-slip with non-linear damping. Plate implemented as facet box



Comparison of simulated and recorded force of first impact during the drop of a low alloyed tool steel 1.3505 from 1cm height

DEM Modeling of Mass Finishing

Model Validation

- Results show enough agreement to allow determination of the distribution of contacts
- Deviations in amplitudes and force distribution can mainly be ascribed to inaccuracies of the robot at low workpiece velocities

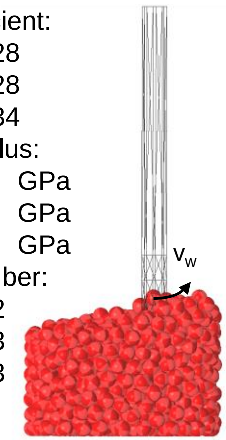
Experiment:
Process:
Robot guided drag finishing
Machine:
Rösler R 220 DL
Comau NJ 370
Workpiece speed:
v_w = 0.015 m/s
Abrasive media:
Walther Trowal
FSG 6 BALLS

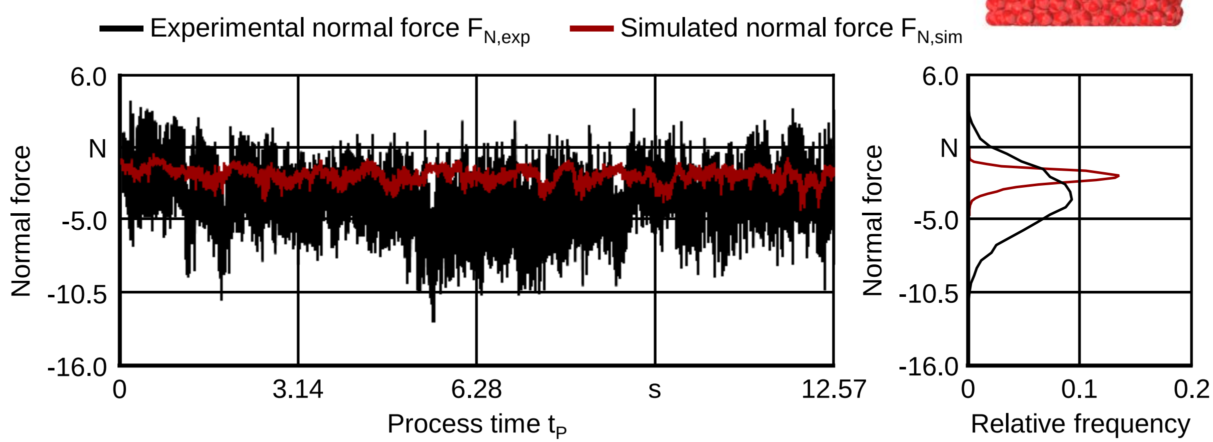
Model setup:
Submersion depth:
d_s = 30 mm
Workpiece speed:
v_w = 0.015 m/s
Radius of spheres:
r_b = 3.27 mm
Number of spheres:
n_b = 877
Filling height:
h_f = 50 mm

Contact law and damping:
Hertz-Mindlin with global damping:
dmp = 0.25

Model material properties:
Density:
ρ_{ball} = 2500 kg/m³
ρ_{rod} = 7900 kg/m³
ρ_{bowl} = 7900 kg/m³

Friction coefficient:
$\mu_{ball,rod}$ = 0.28
$\mu_{ball,bowl}$ = 0.28
$\mu_{ball,ball}$ = 0.34
Young‘s Modulus:
E_{ball} = 90 GPa
E_{bowl} =210 GPa
E_{rod} =210 GPa
Poisson‘s number:
ν_{ball} = 0.2
ν_{bowl} = 0.3
ν_{rod} = 0.3





INSTITUTE PRODUCTION SYSTEMS AND DESIGN TECHNOLOGY

INSTITUTE FOR MACHINE TOOLS AND FACTORY MANAGEMENT TECHNISCHE UNIVERSITÄT BERLIN

DEM Modeling of Mass Finishing

Utilization - Absolute Linear Velocity of Abrasive Media in Drag Finishing

Video

Magnitude of linear velocity in m/s

DEM Modeling of Mass Finishing

Utilization - Interaction Network during Drag Finishing

Normal Force at contacts in N

Video

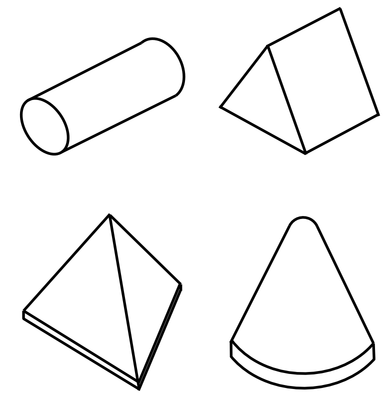
Video

Note that according to Hongyang Alex Cheng there was a bug related to vtkExport of normal forces at interactions.

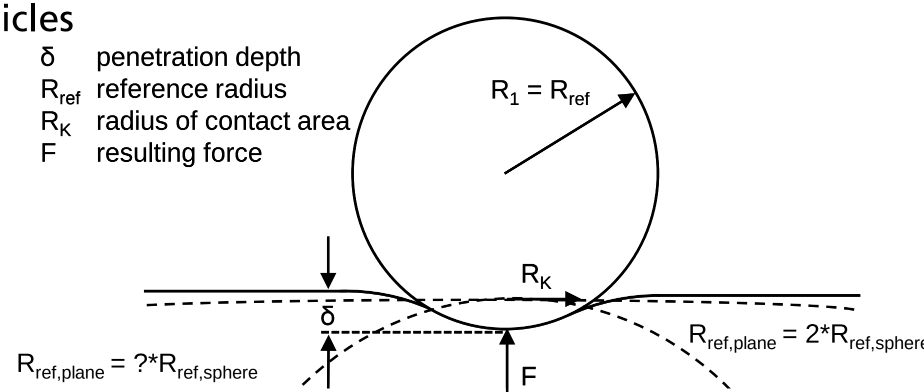
DEM Modeling of Mass Finishing

Wishlist for Yade / Feature Requests / ToDo

- Modeling of non-spherical particles
- Improvement of particle-facet contact (Reference Radius, Facet-Facet-Sphere interaction)
- Static and dynamic (kinetic) friction coefficients
- DEM on GPU for better performance with lots of particles
- VTKRecorder: self defined data for export



Selection of relevant media shapes



Contact of sphere and plane/facet

Summary and Outlook

Summary

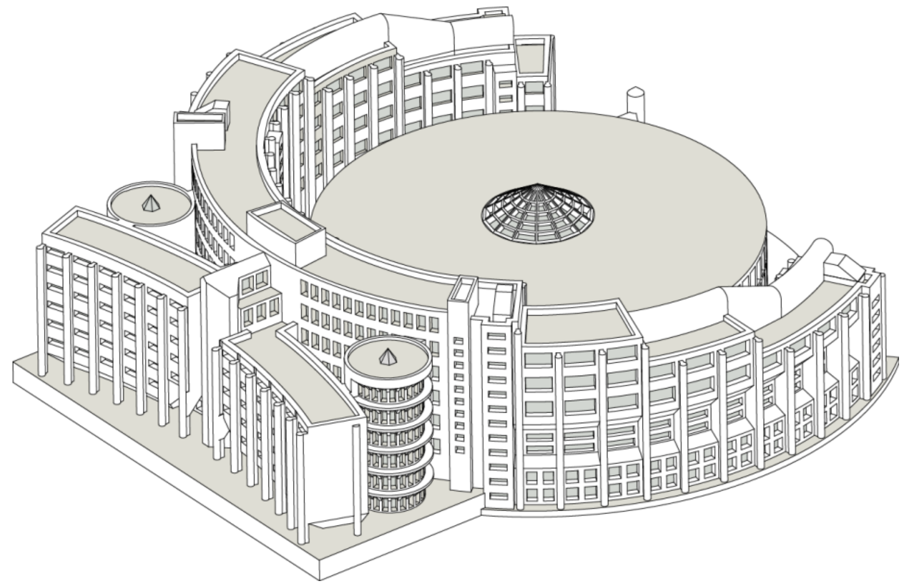
- DEM is suitable to model bulk motion in mass finishing and identify local forces acting on finite areas of workpieces
- Model calibration and validation is important

Outlook

- Modeling of actual mass finishing processes and improvement of model performance
  - Non-spherical media (clumps, exact representation)
  - Lubrication
  - Vibrating bowl
  - Efficiency
- Post processing with ParaView

Thank you!

M. Sc. Alexander Eulitz
Research Engineer
E-Mail: eulitz@iwf.tu-berlin.de
Phone: +49 30 314 24963
Office PTZ 1
Pascalstraße 8-9
10587 Berlin

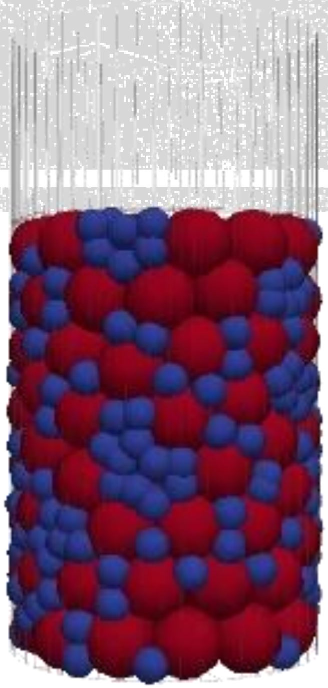


UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Biphasic particles to simulate fresh pervious concrete compaction

Ricardo Pieralisi

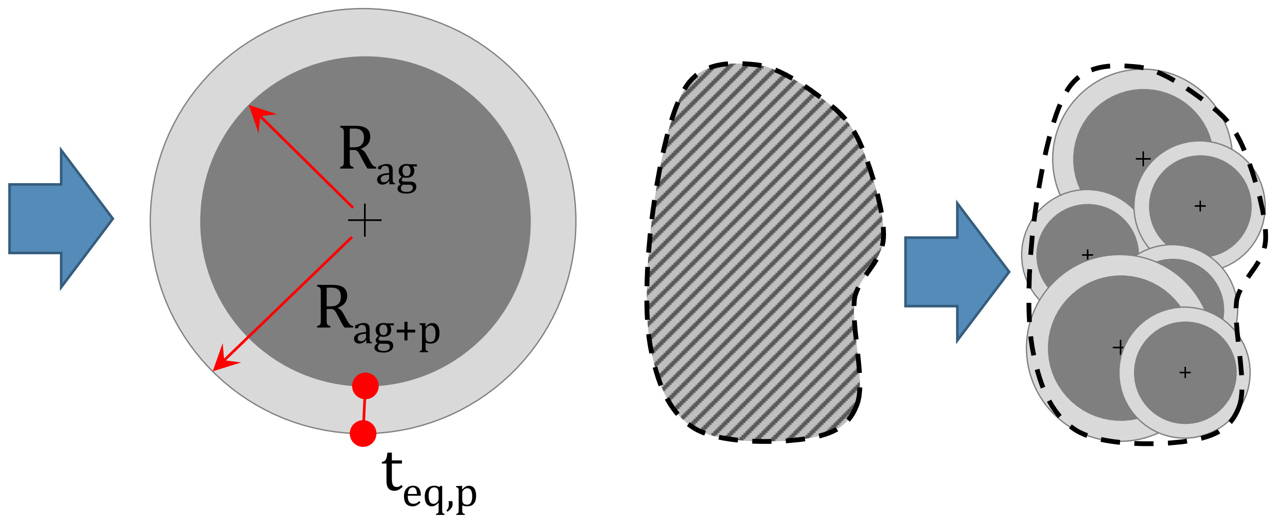
ricardo.pieralisi@upc.edu



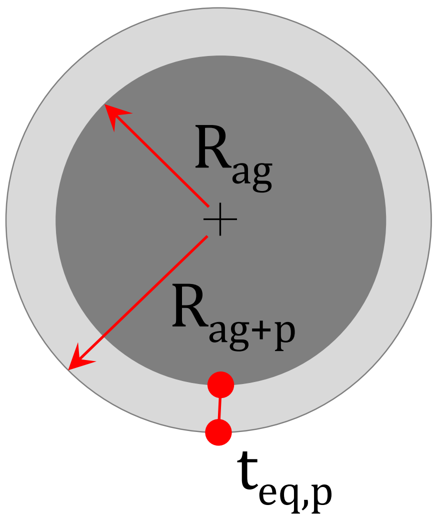
Pervious Concrete

- Low quantity of cement paste
- Interconnected pores
- High permeability coefficient

Particle Definition

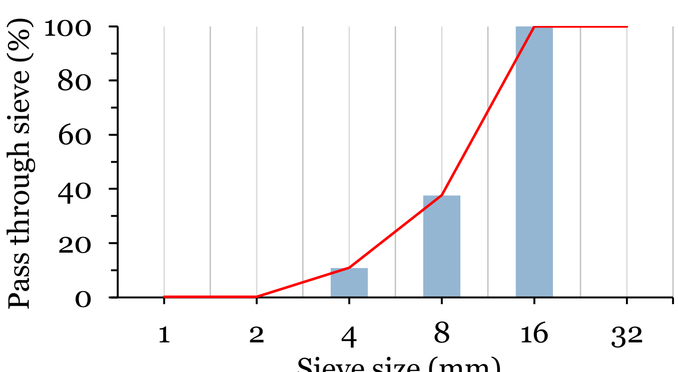


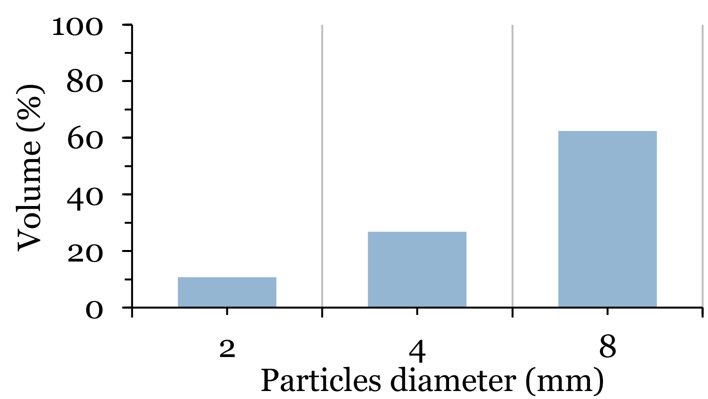
Particle Definition



- Elastic inner core
- Viscoelastic external layer

$$R^{i}_{ag+p} = R^{i}_{ag} \cdot \left(1 + \alpha_{eq}\right)$$

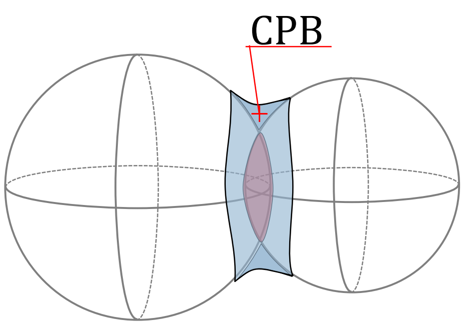




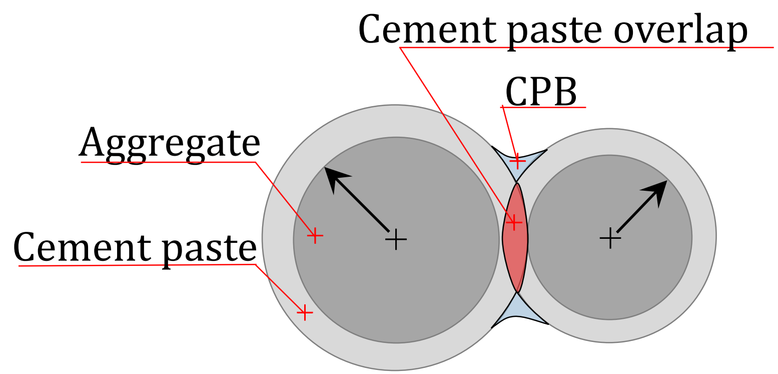
Characteristic of the Contact

- Cement Paste Bridge (CPB) formation

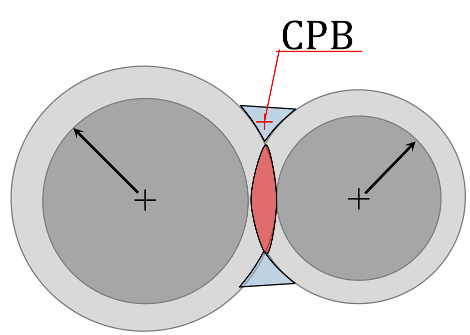
(a) Real format of the CPB



(b) Middle cross section

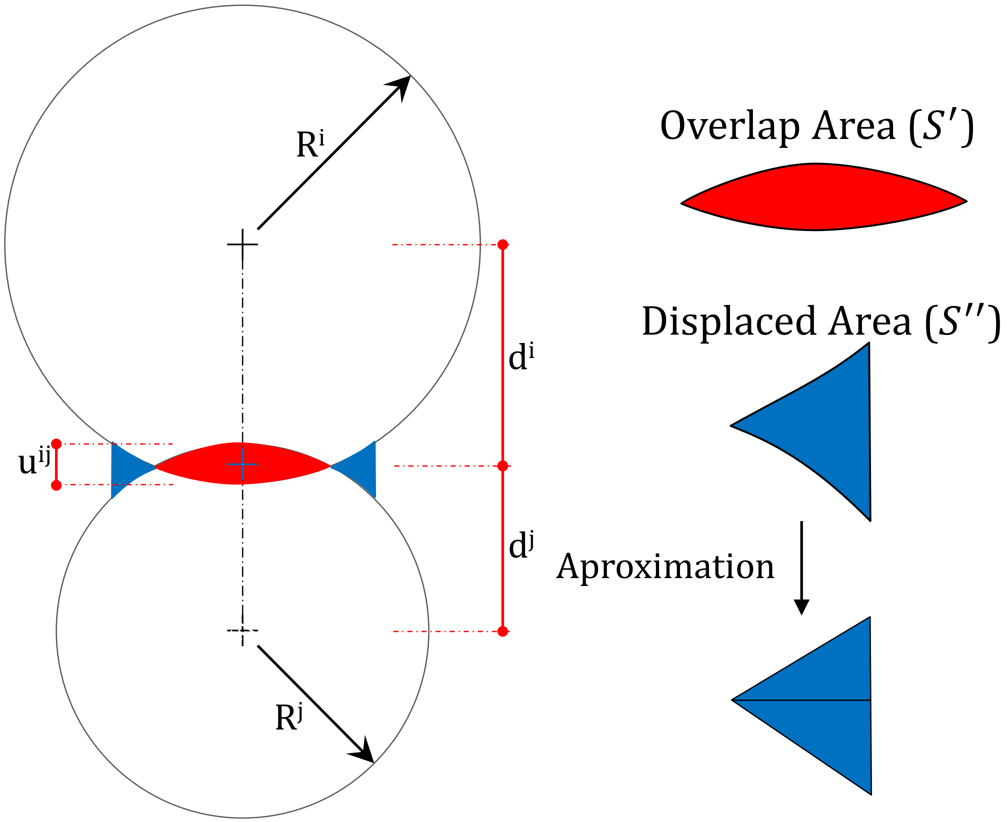


(c) CPB approximation



Characteristic of the Contact

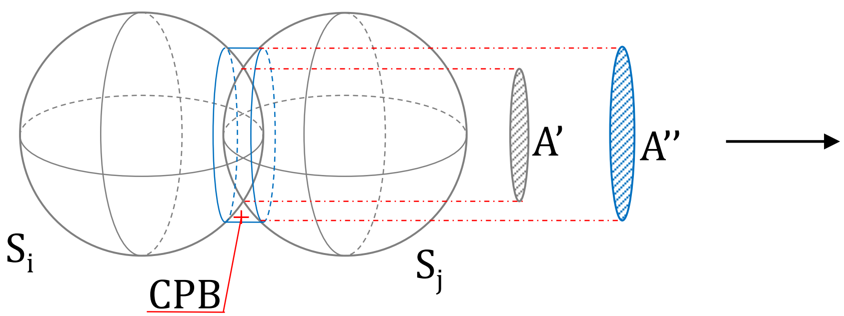
- Cement Paste Bridge (CPB) formation

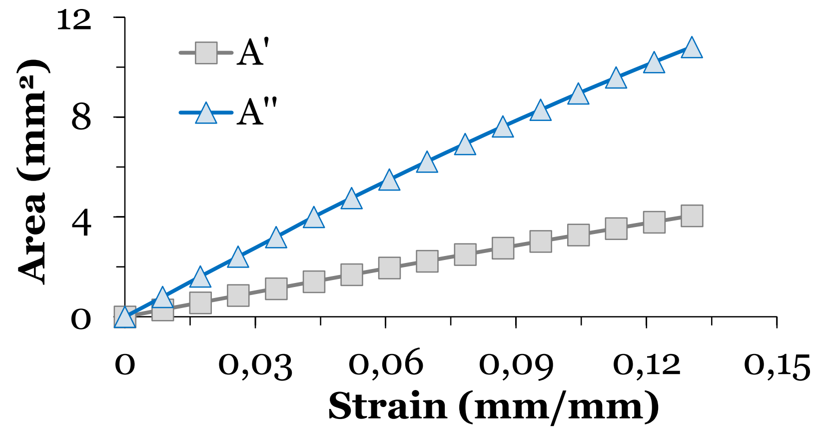


Characteristic of the Contact

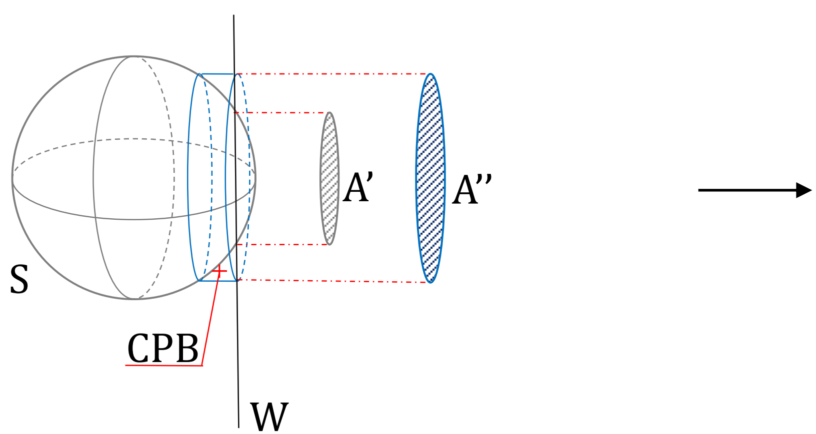
- Cement Paste Bridge (CPB) consideration

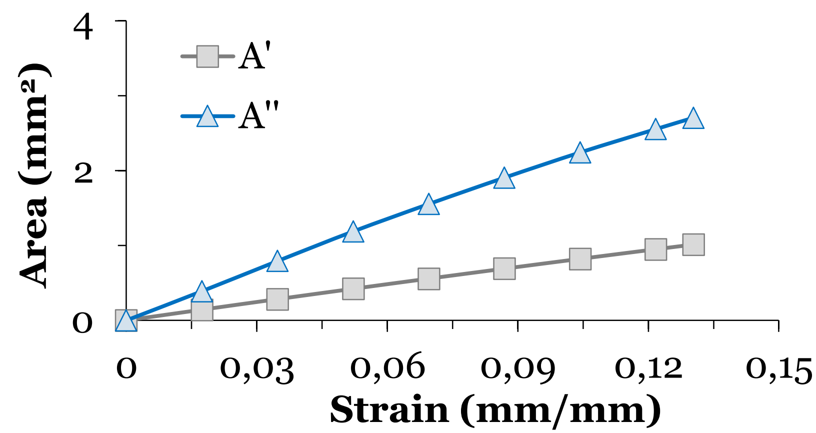
(a) Interaction S- S





(b) Interaction S- W

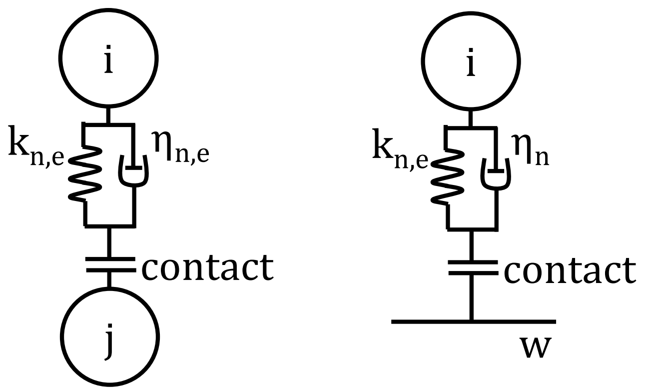




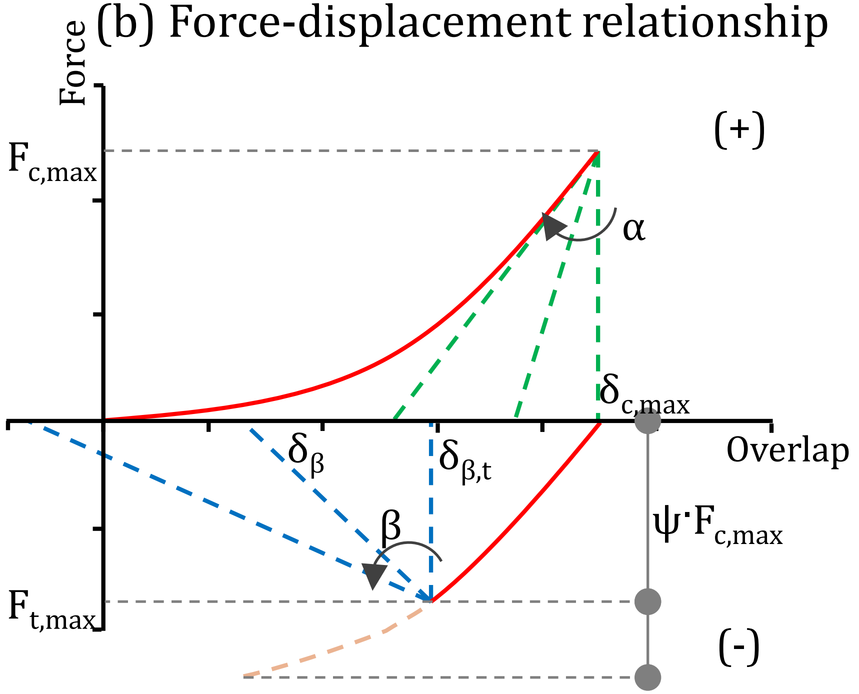
Material Model

- Normal direction

(a) Rheological model



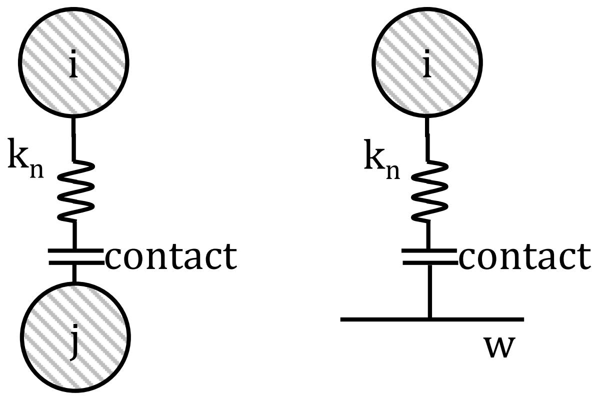
(b) Force-displacement relationship



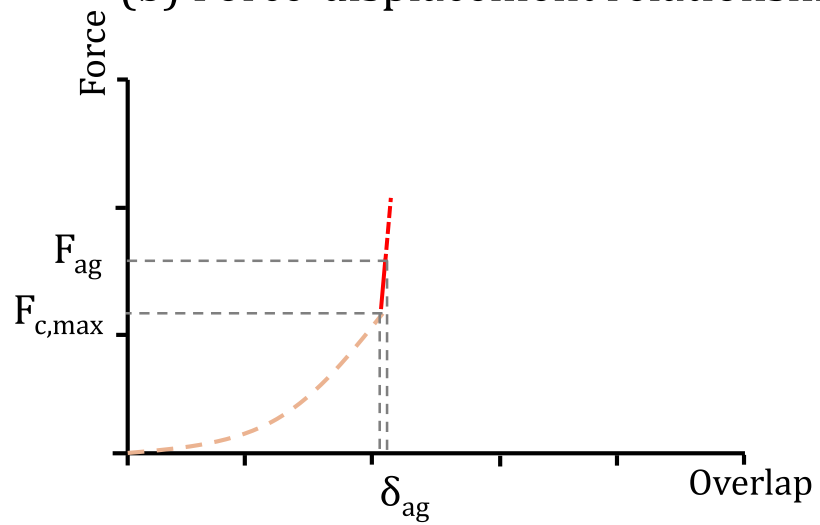
Material Model

- Normal direction

(a) Rheological model



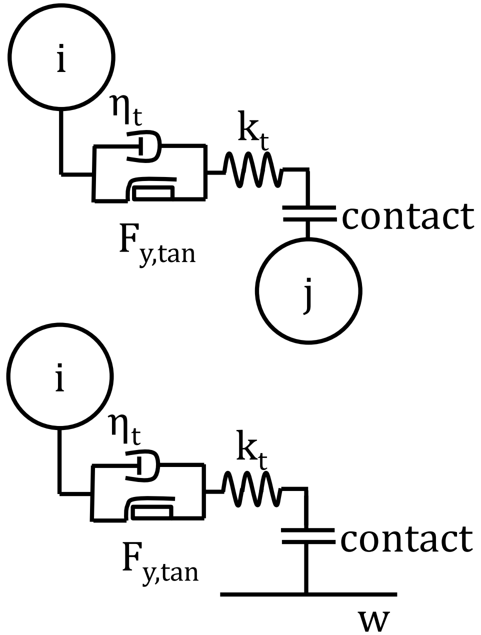
(b) Force-displacement relationship



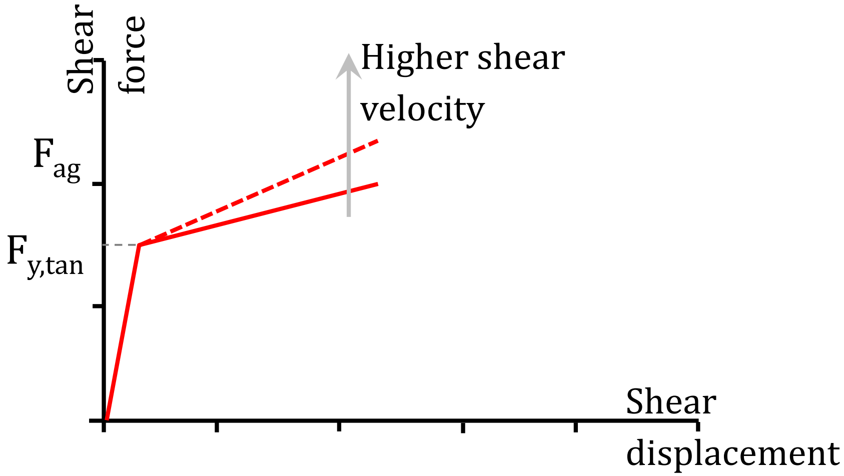
Material Model

- Tangential direction

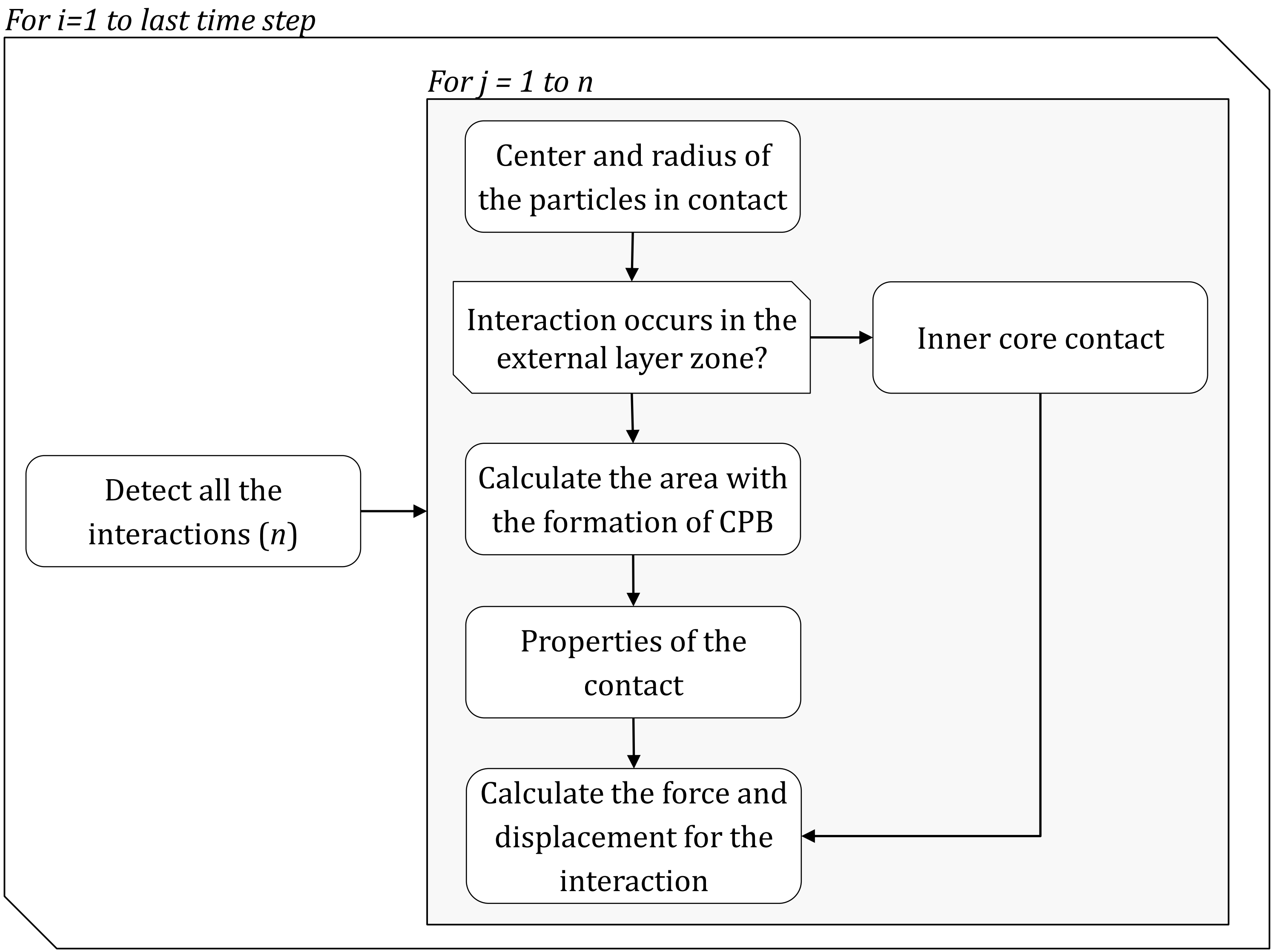
(a) Rheological model



(b) Force-displacement relationship

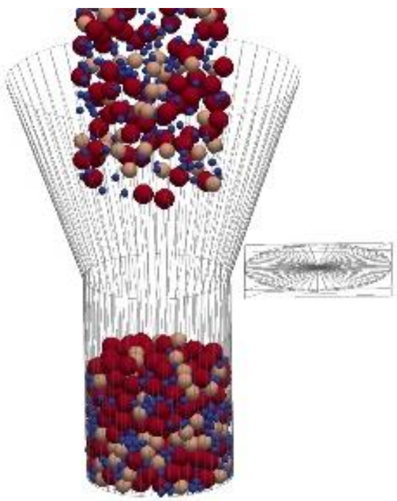


Solution Process

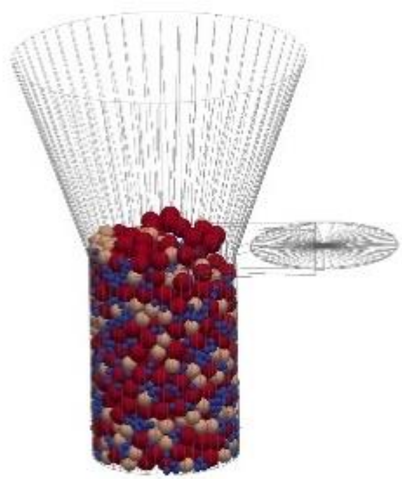


Solution Process

(a) Filling process



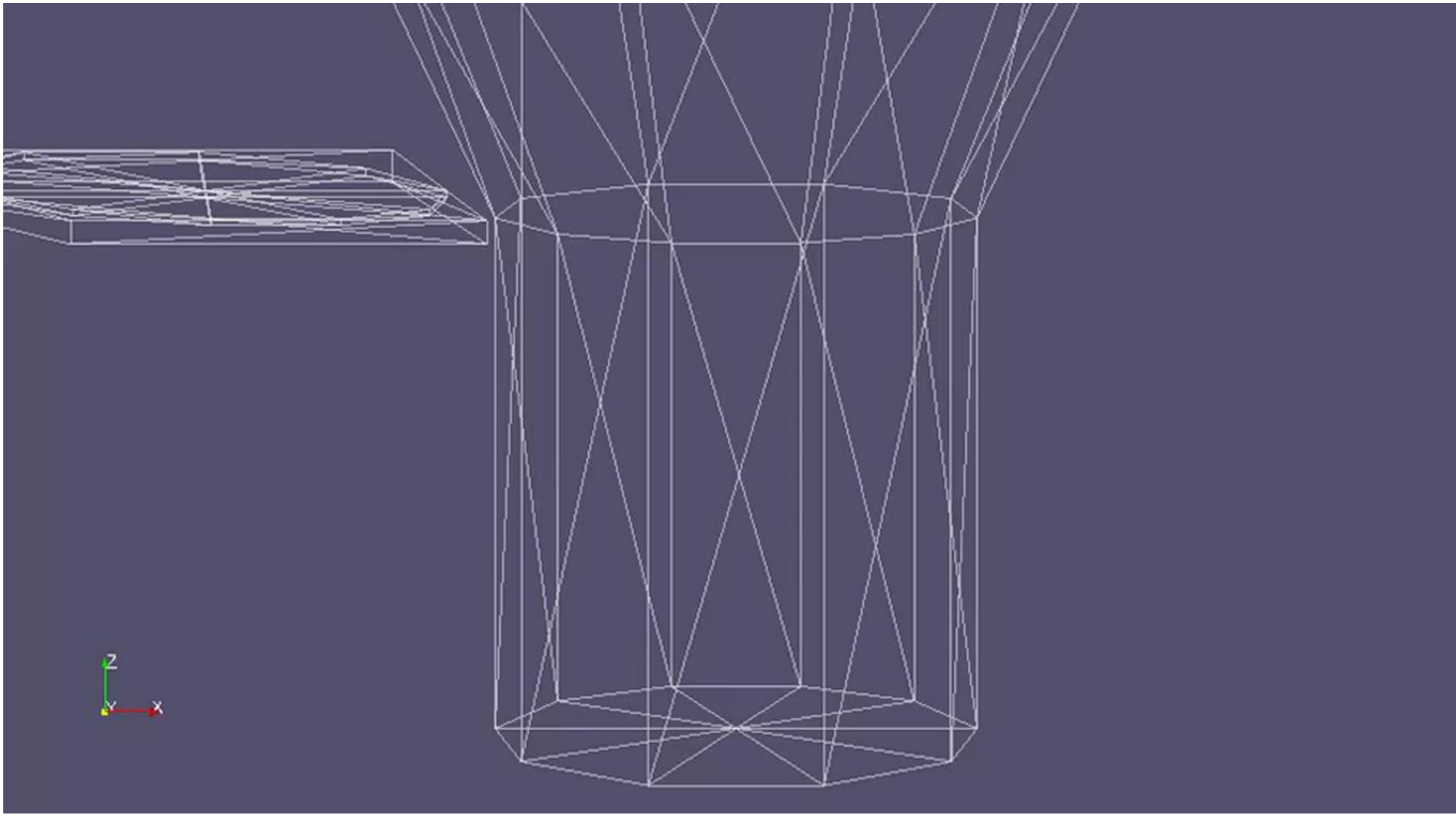
(b) Cleaning process



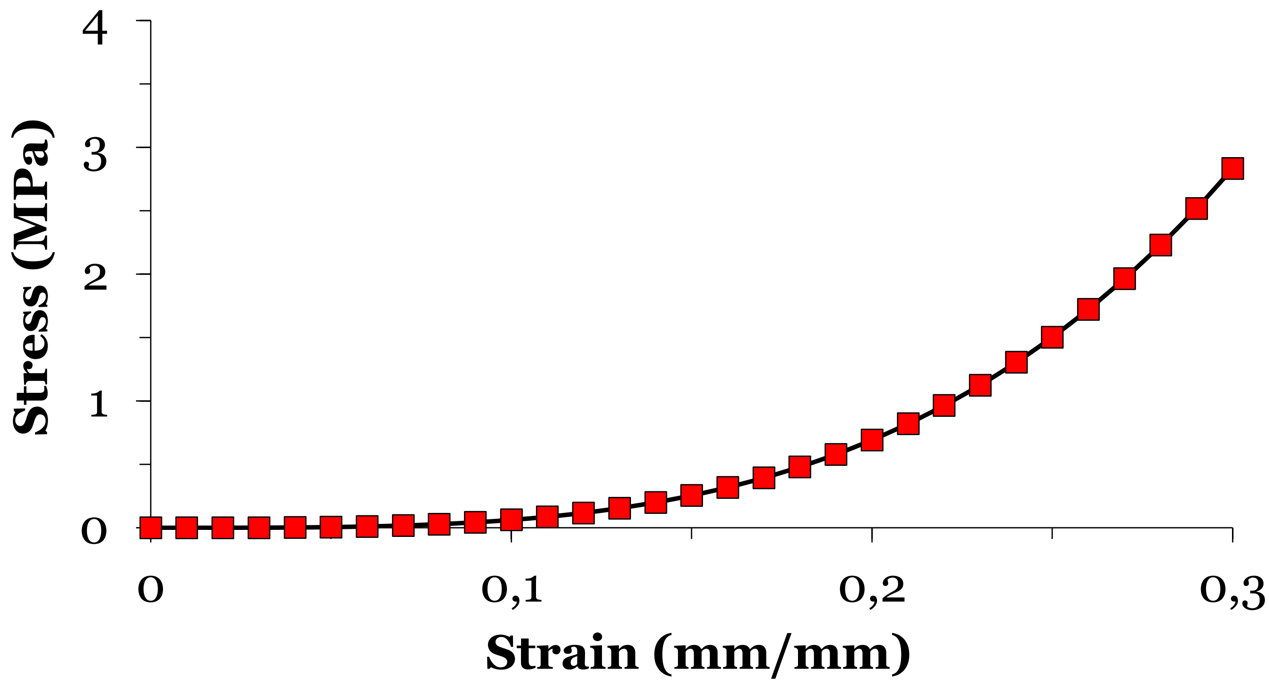
(c) Beggining of compaction

(d) End of compaction

Solution Process

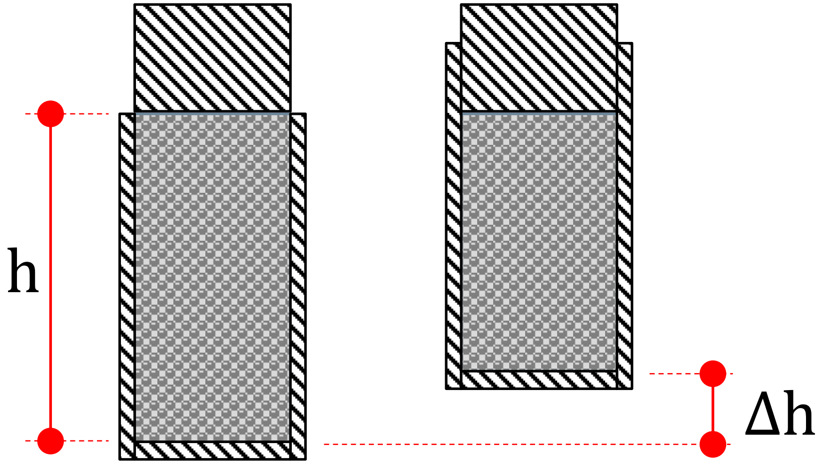


Solution Process



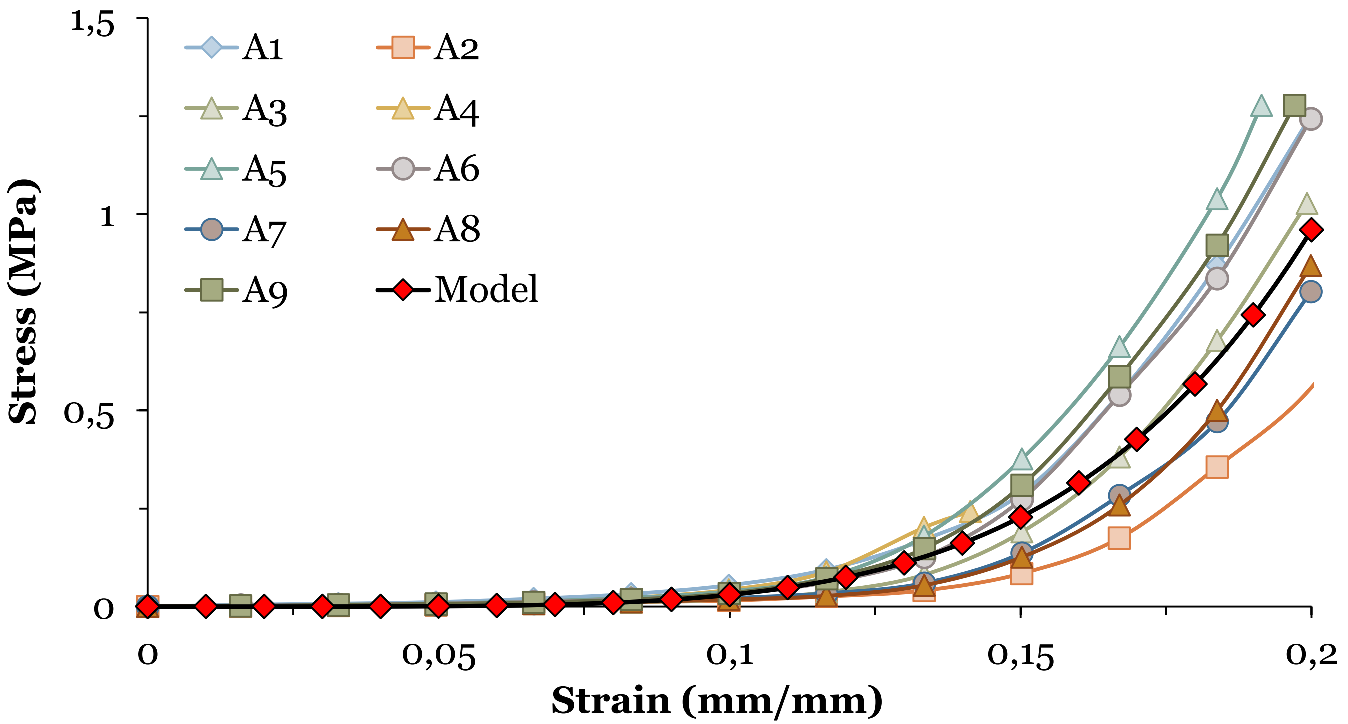
Experimental program

- Equipment

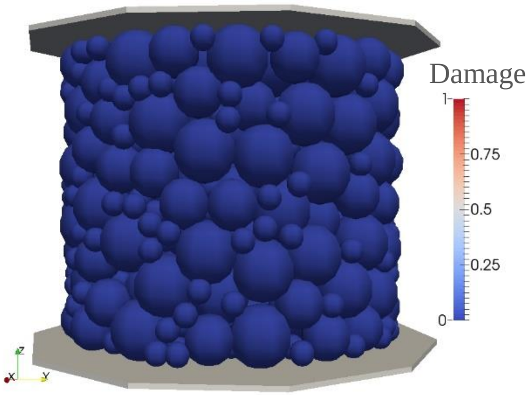


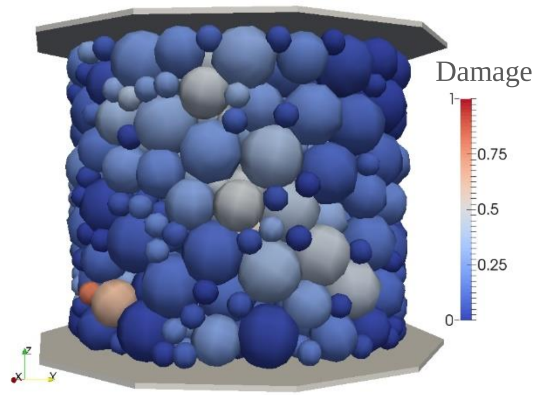
Validation

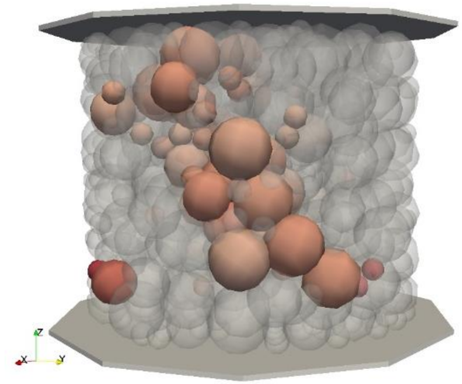
- Equipment

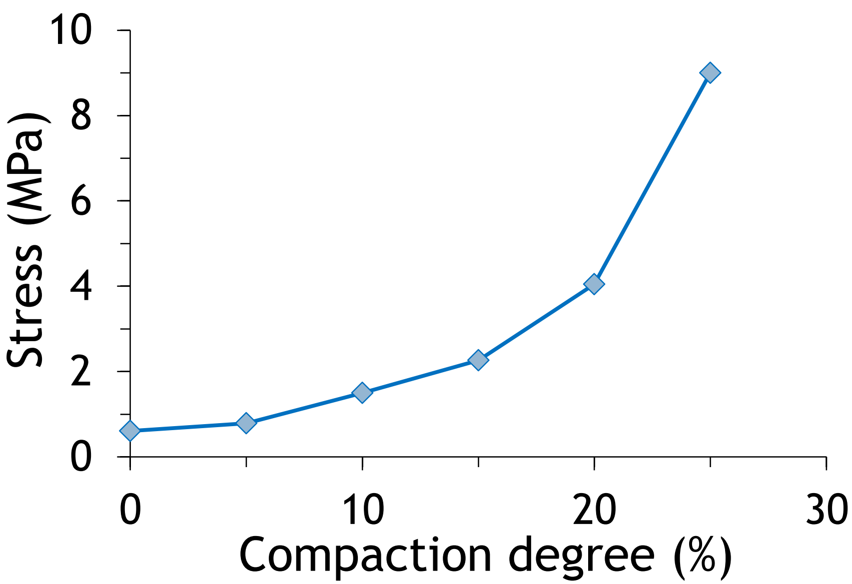


Mechanical Properties

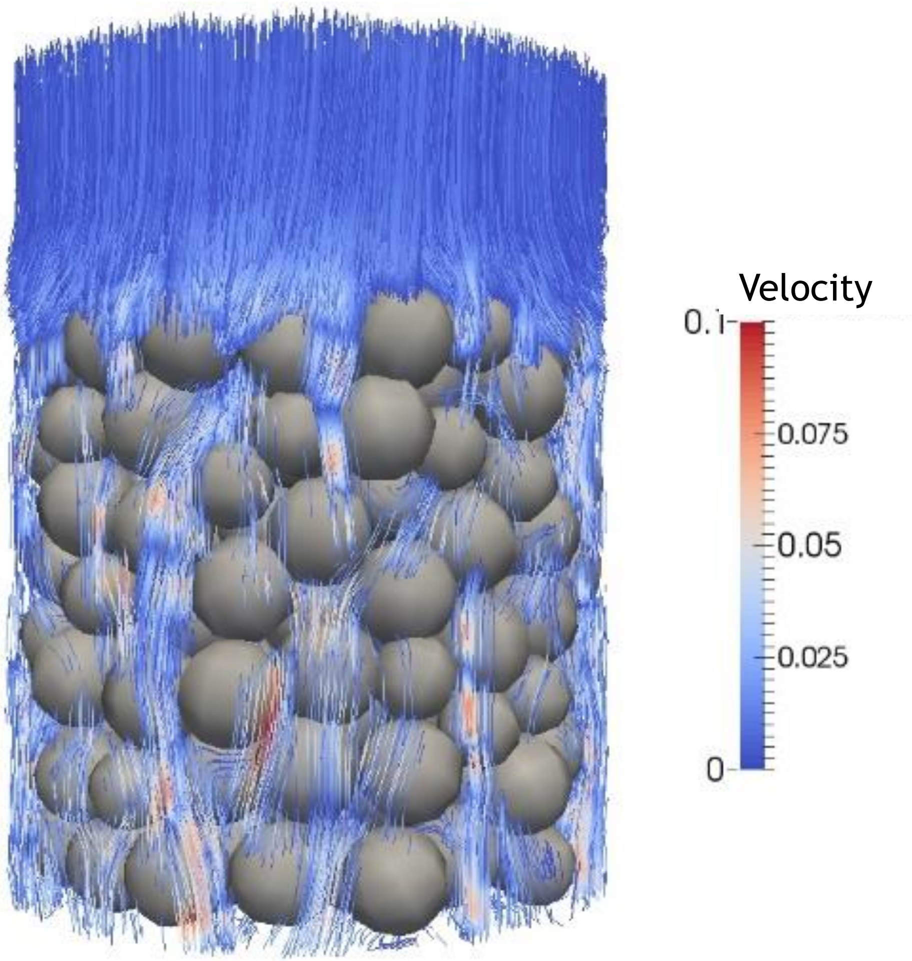


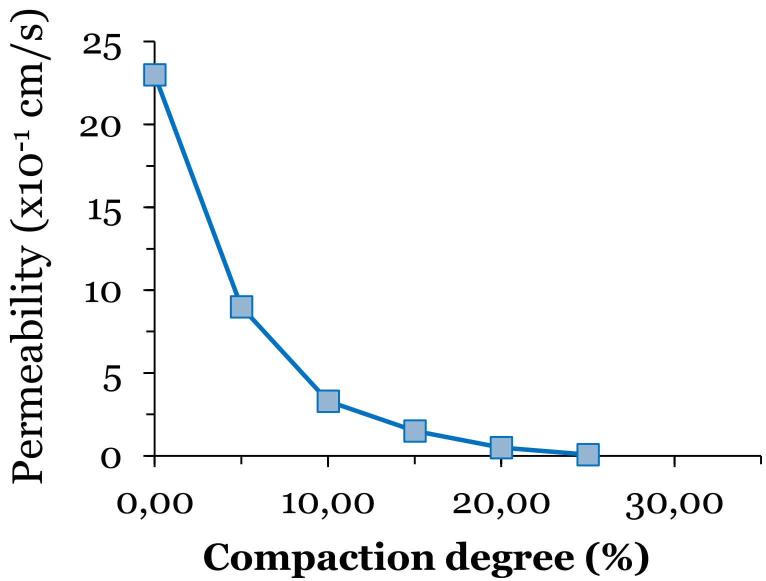






Fluid Transport





UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Ricardo Pieralisi

ricardo.pieralisi@upc.edu

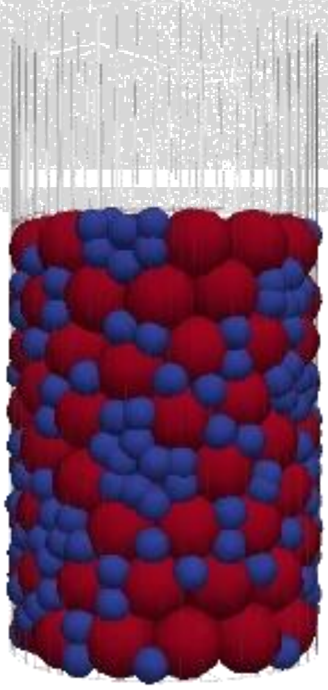
Thank you!

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Biphasic particles to simulate fresh pervious concrete compaction

Ricardo Pieralisi

ricardo.pieralisi@upc.edu



DEM applications in rock (hydro-)mechanics

Frédéric Donzé,
Zoheir Boukria, *Jérôme Duriez*
Viviana Bonnilla-Sierra, Efthymios Papachristos
Alexandra Tsopela

In collaboration with :
Bruno Chareyre, Luc Scholtès, Marc Elmouttie

3SR, Université Grenoble-Alpes, CNRS
GeoRessources, Université de Lorraine, CNRS

Structure Fédérative RNVOR

TOTAL

ANR GeoSMEC

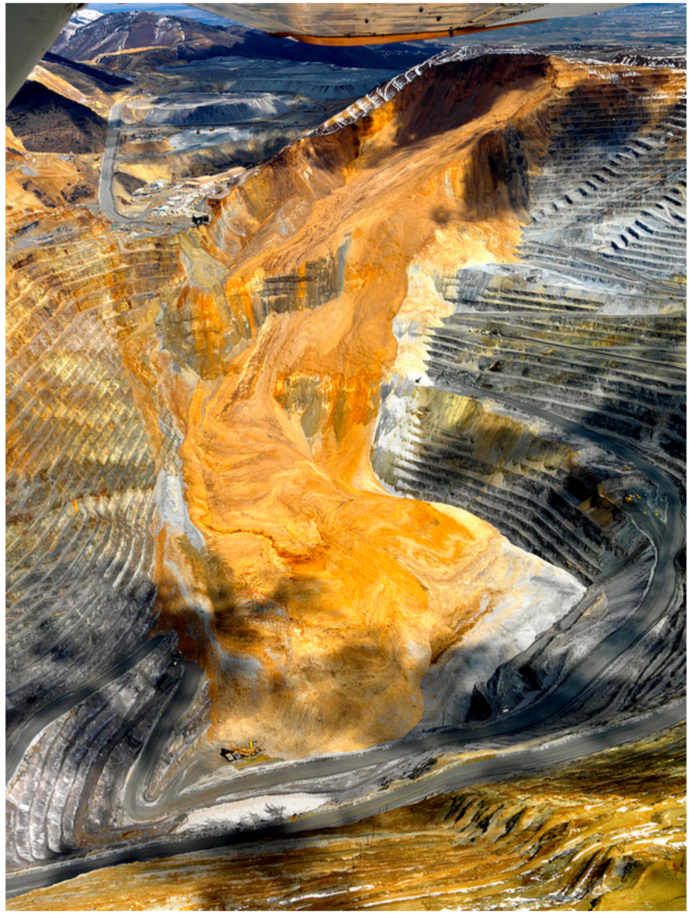
CSIRO

IMS RN Ingénierie des Mouvements de Sol et des Risques Naturels

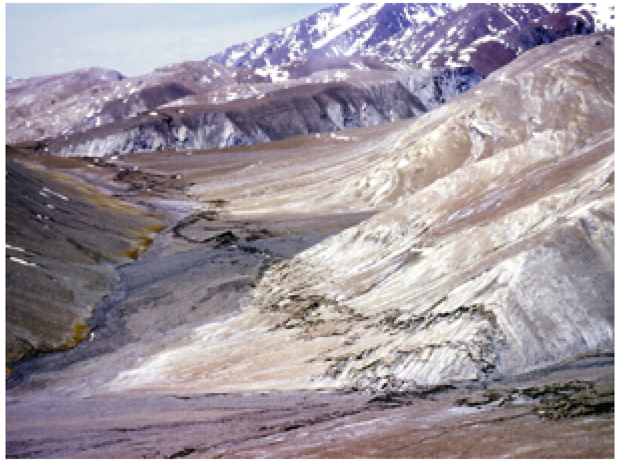
3SR

1st YADE workshop, July 7th 2014

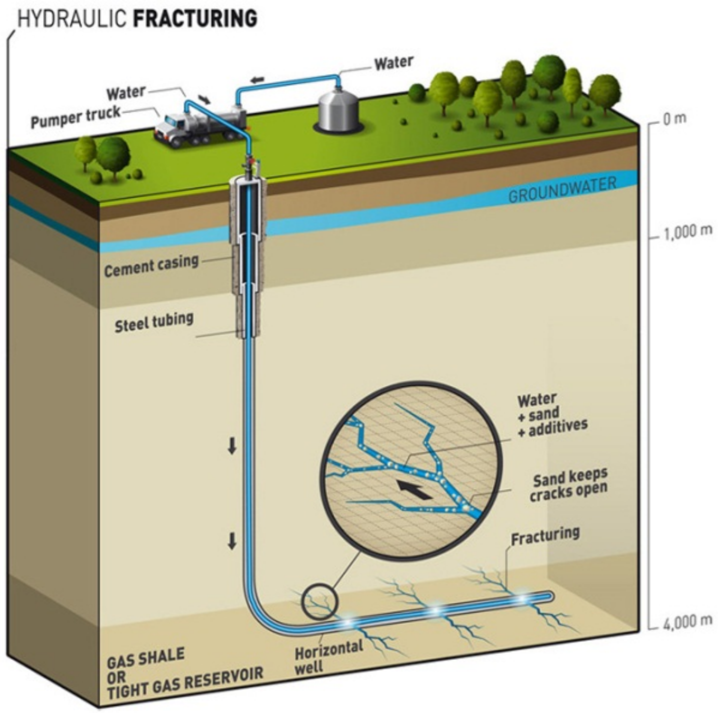
Rock hydro-mechanics case studies



Bingham canyon mine collapse: 0 casualties (Rio Tinto)



Fault after M 7,6 earthquake in Tibet (Y. Klinger, IPGP)



Exploitation of various oil and gas reservoirs

Outline

~~I. DEM modelling of rock~~ (tomorrow morning)

I. Rockfall risk assessment
V. Bonilla-Sierra

II. Crack propagation analysis
J. Duriez

III. Hydro-mechanics for fluid extraction/injection processes
A. Tsopela, Z. Boukria — E. Papachristos

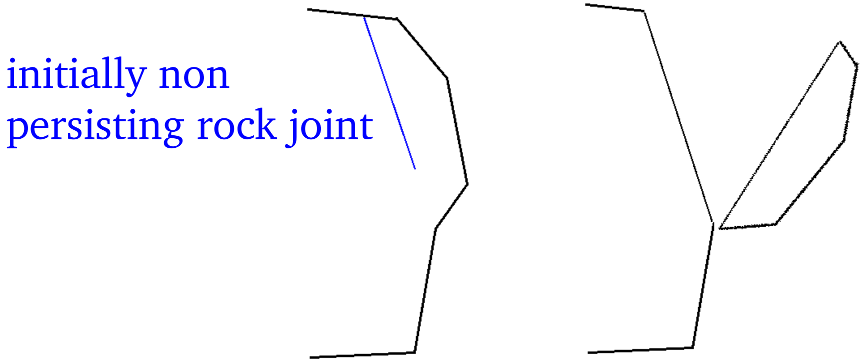
3DEC™

I. Rockfall risk assessment

Take advantages of the DEM for

- a straightforward description of a discontinuous media

- a straightforward description of progressive failure in the rock matrix (rock bridges) through debonding between DE



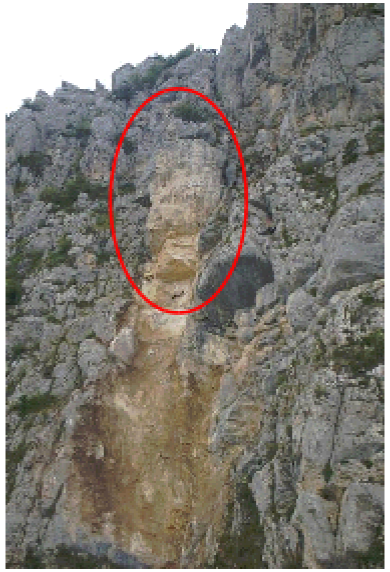
In order to

predict the rockfall !
(triggering conditions and volume)

Rockfall in Arly Gorges (CG73)

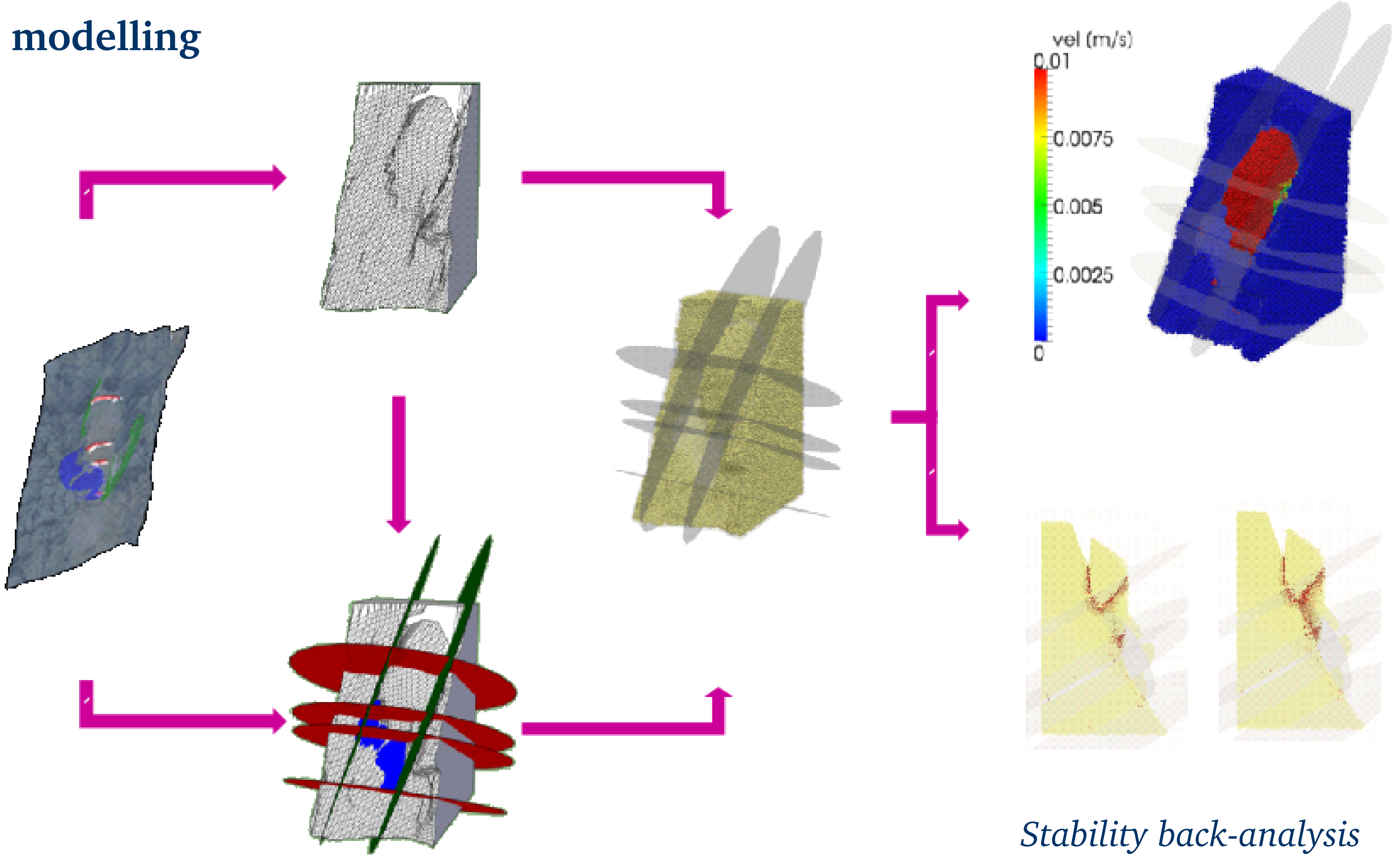
I. Rockfall risk assessment

From in-situ observations to 3D modelling



Unstable rock mass in Néron (Grenoble)

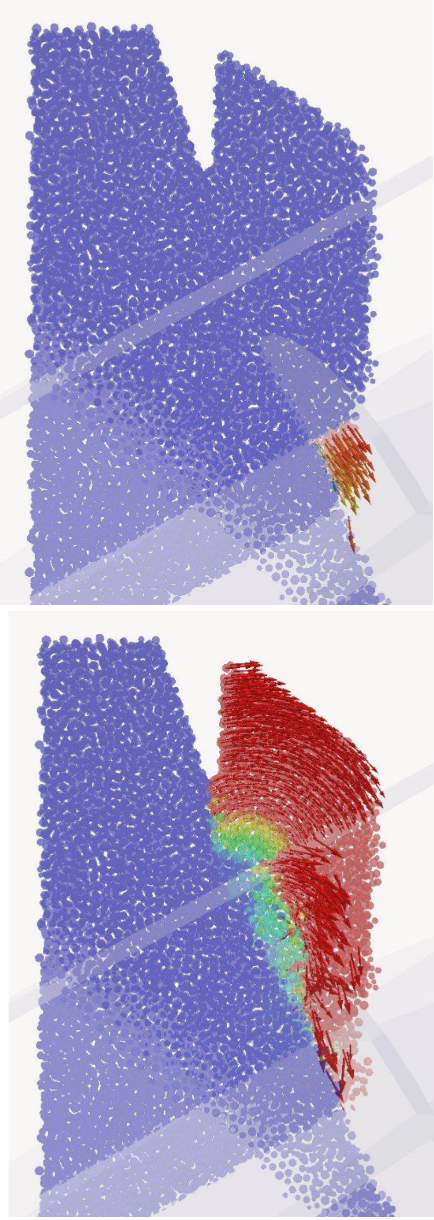
+ Sirovision code



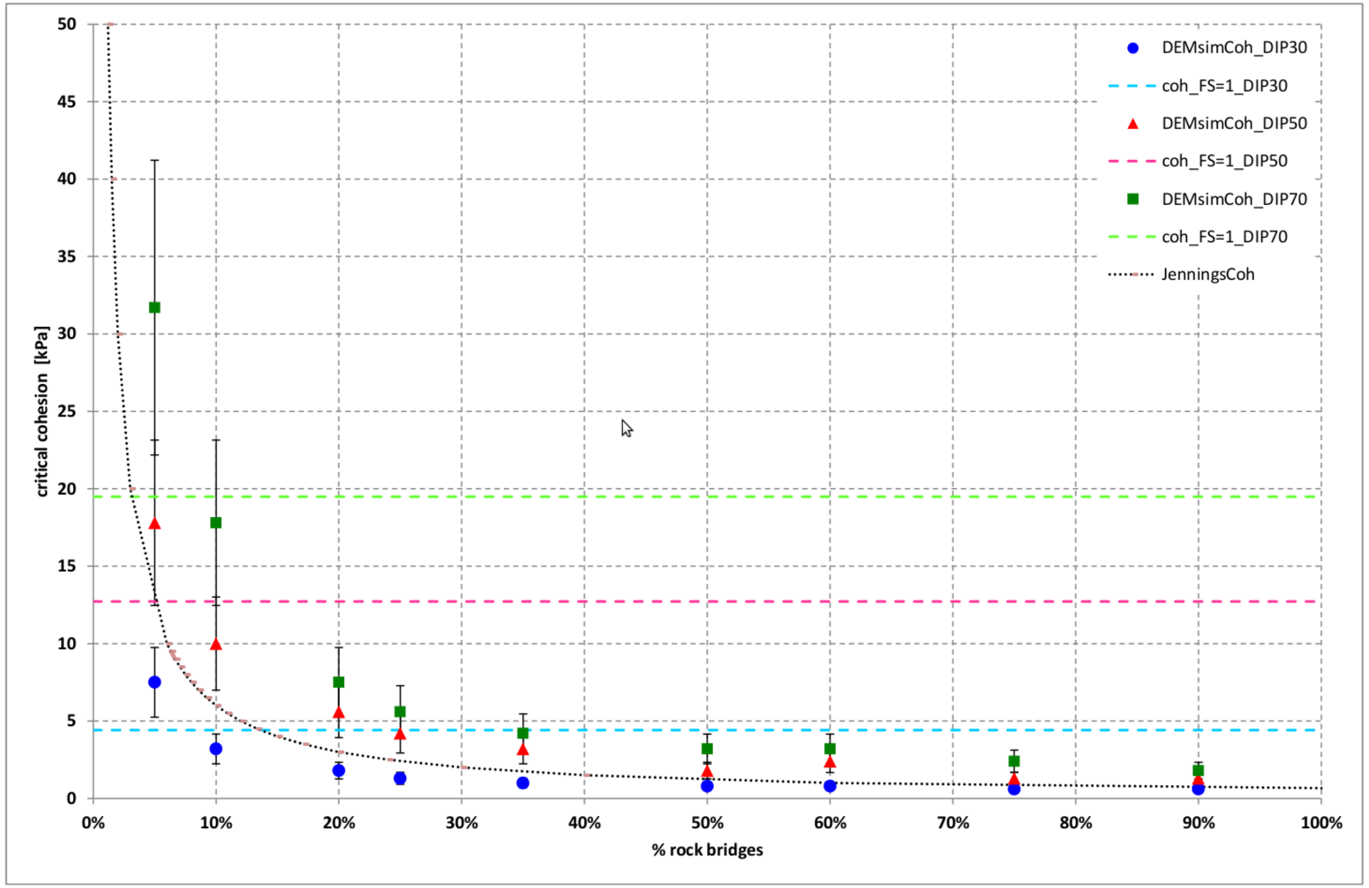
Stability back-analysis

DEM model of the cliff: rock matrix and surface discontinuities (cf tomorrow morning)

I. Rockfall risk assessment



Different failure mechanisms



Limit cohesion according to analytical and DEM stability analysis

II. Crack propagation analysis

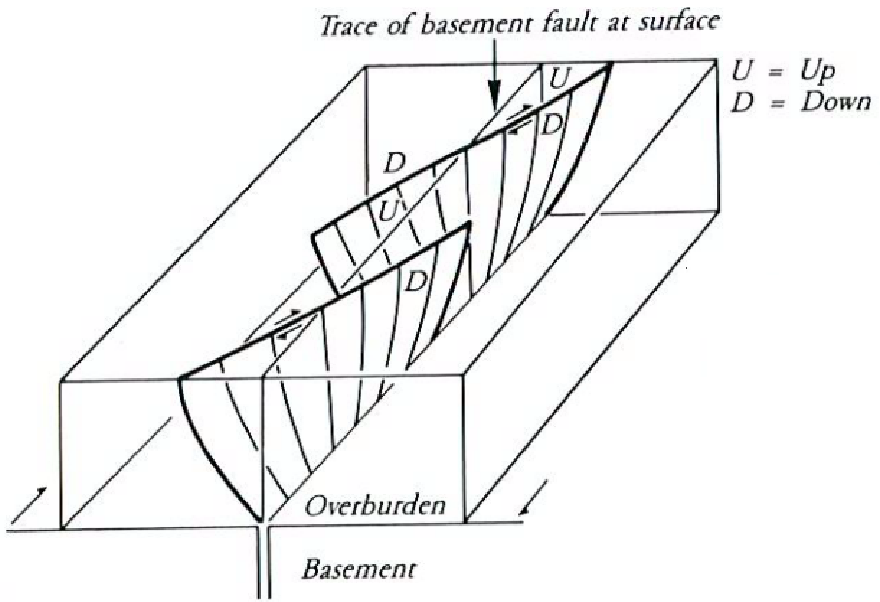
Take advantages of the DEM for

- a straightforward description of a pre-existing cracks

- a straightforward description of damage propagation

In order to

monitor where damage occurs. Special focus on tectonics faults geometry.

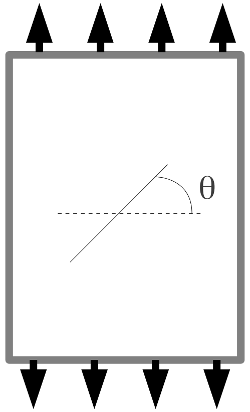


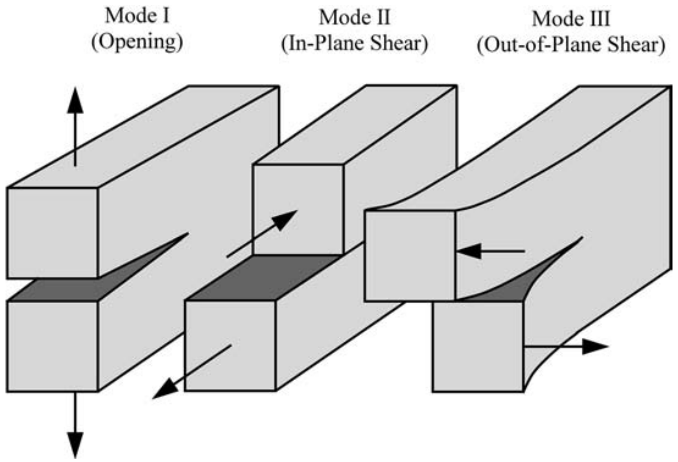
Fault formation scheme (Mandl, 1988)

II. Crack propagation analysis

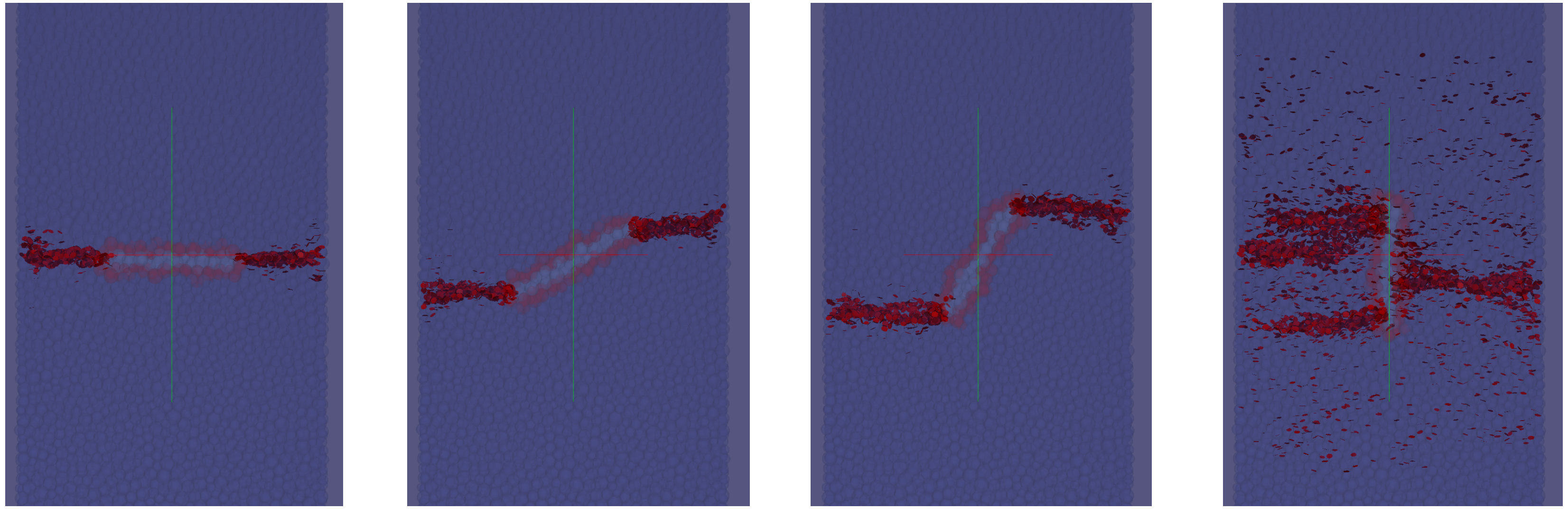
Justification of the model

Mode I+II analysis





The 3 fundamental crack loadings (Anderson, 2005)



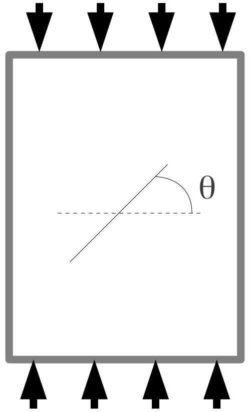
Crack patterns for different flaw orientations

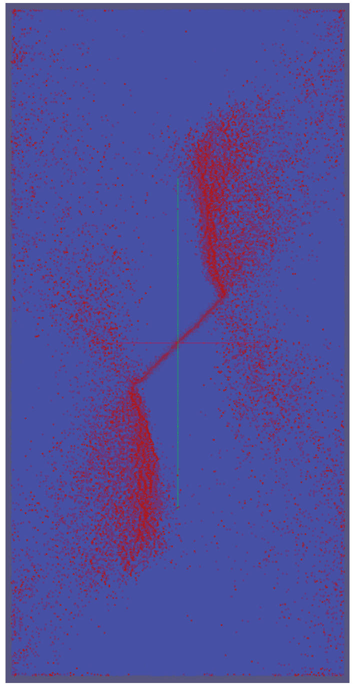
Propagation of DE cracks (according to σ_I direction) consistent to previous models !

II. Crack propagation analysis

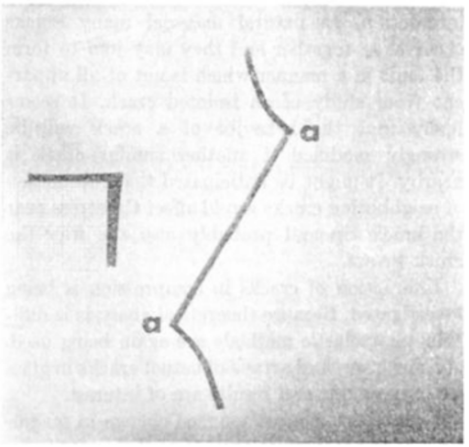
Justification of the model

Mode I+II analysis

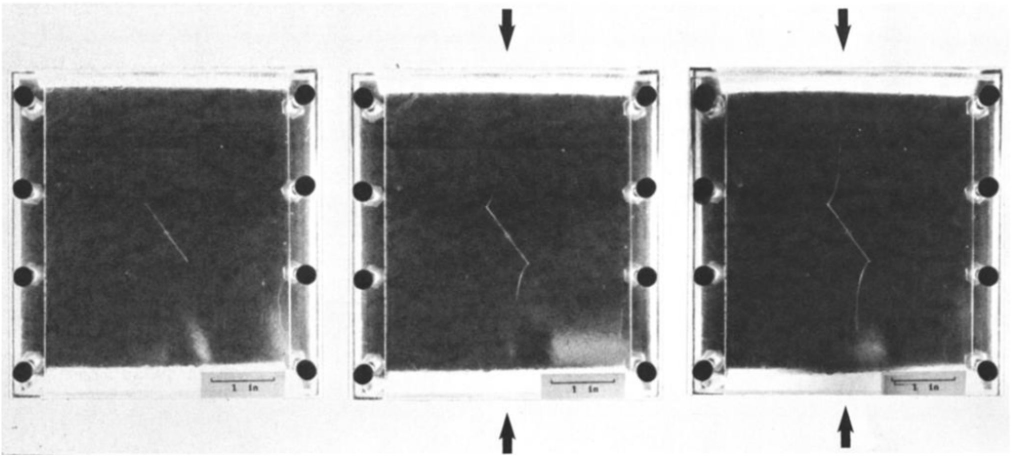




DEM



Crack growth in photo-elastic material (Brace, 1963)



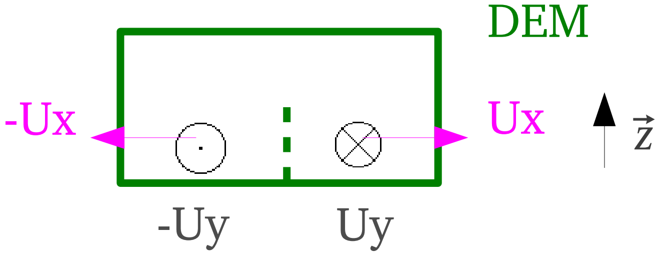
Experiments on a brittle model material (Nemat-Nasser, 1982)

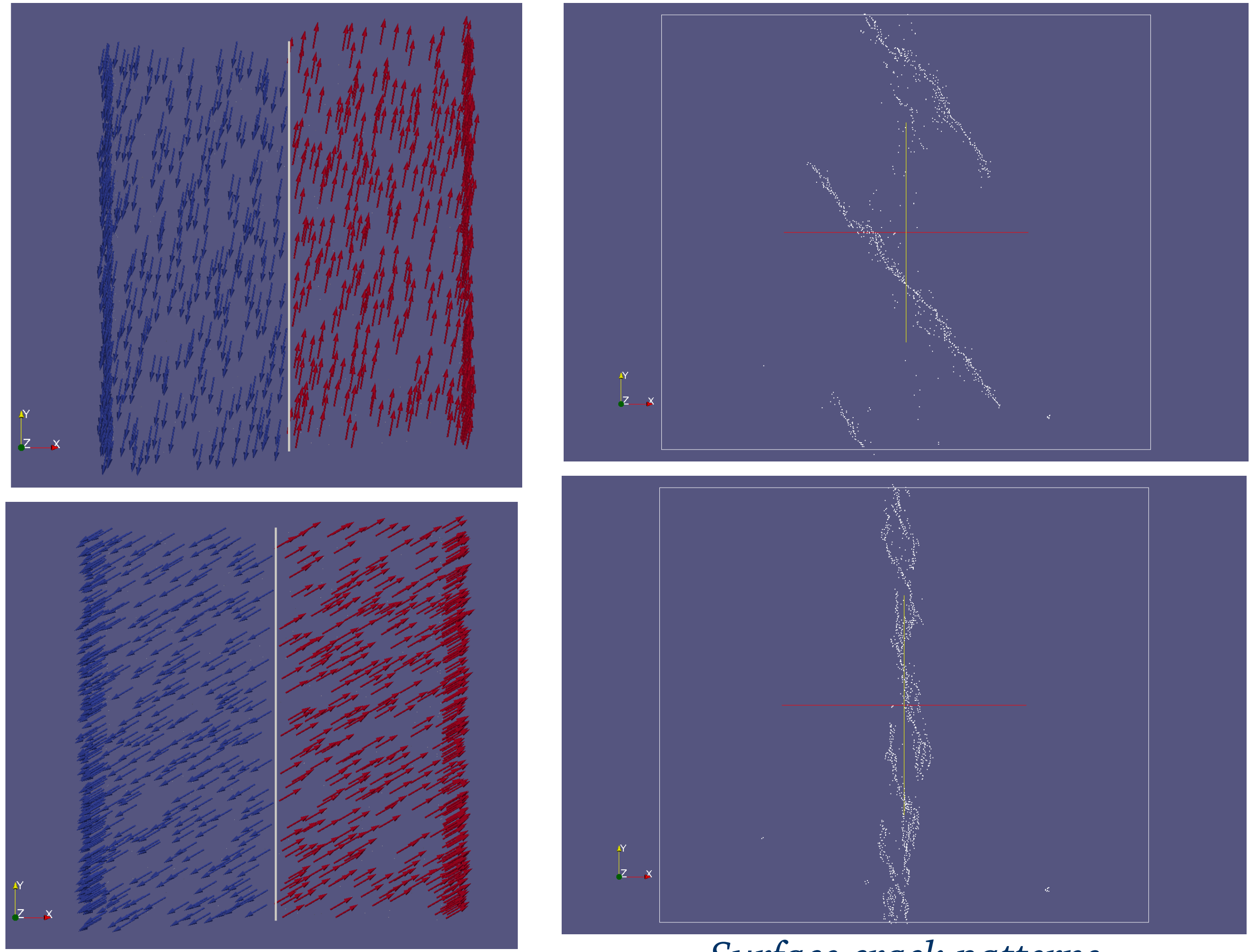
Wing cracks aligned with major principal stress after some kink at the tip

II. Crack propagation analysis

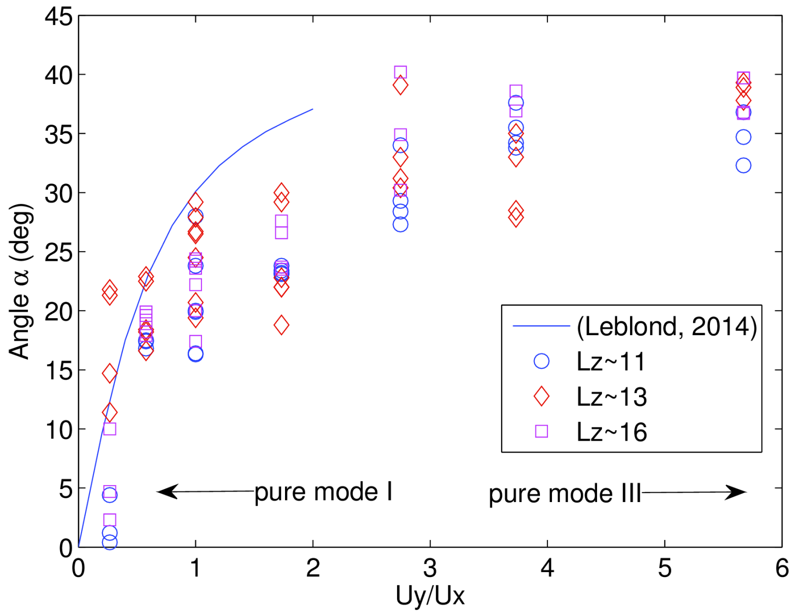
Justification of the model

Mode I+III

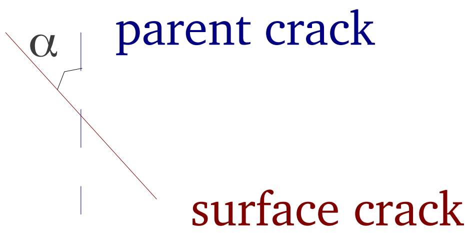




Surface crack patterns



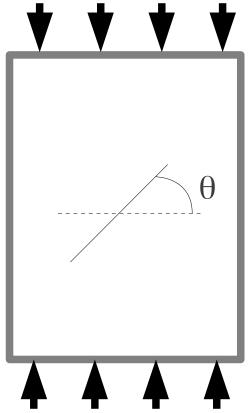
Orientation of surface cracks : DEM vs analytical LEFM

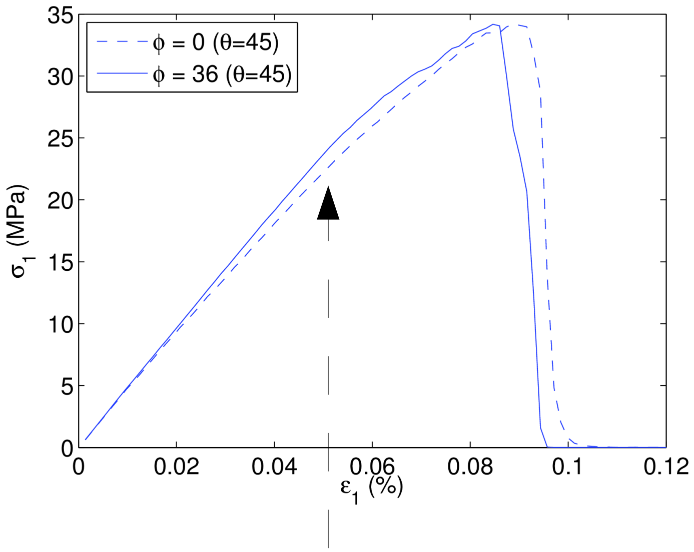


II. Crack propagation analysis

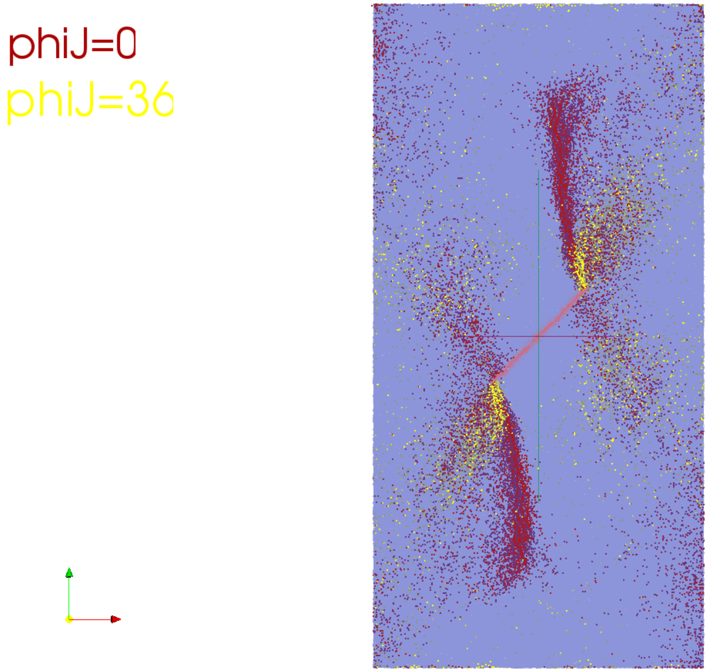
New insights

Mode II analysis





Damage propagation ruled also by the crack mechanical properties, not only by those of the matrix !



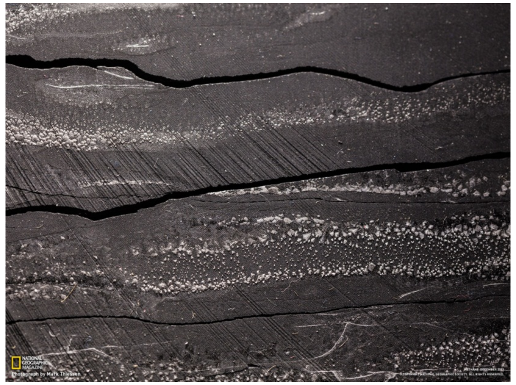
$\varepsilon_1 \sim 0{,}05$ %

III. Hydro-mechanics for fluid extraction/injection processes

Take advantages of the DEM for

- a straightforward description of a cohesive pre-cracked media, that can damage.

- various hydro-mechanical coupling methods

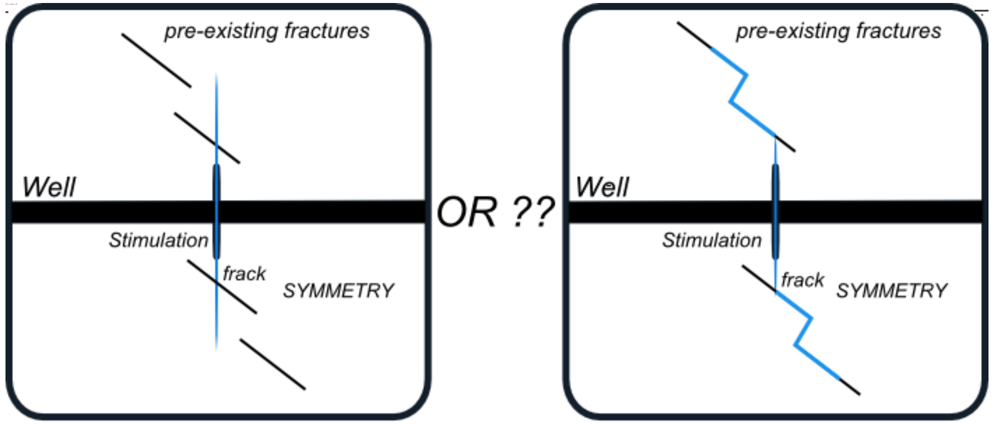


Shale

In order to

- monitor crack propagation, for a human-defined flow, to optimize the injection (hydraulic fracking)

- monitor fluid propagation, for nature-defined cracks, to optimize the extraction (production)



III. Hydro-mechanics for fluid extraction/injection processes

- set of polyedric blocks interacting along pre-defined surfaces. Rigid or deformable blocks, but without any damage

- compressible fluid flows along the surfaces (flow between parallel plates). With no flow through the blocks

- permeability depends on normal relative displacement between blocks

- use facilities, e.g. for stochastic fracture network generation

- set of bonded (or not) spherical particles. Deformation occurs until possible debonding.

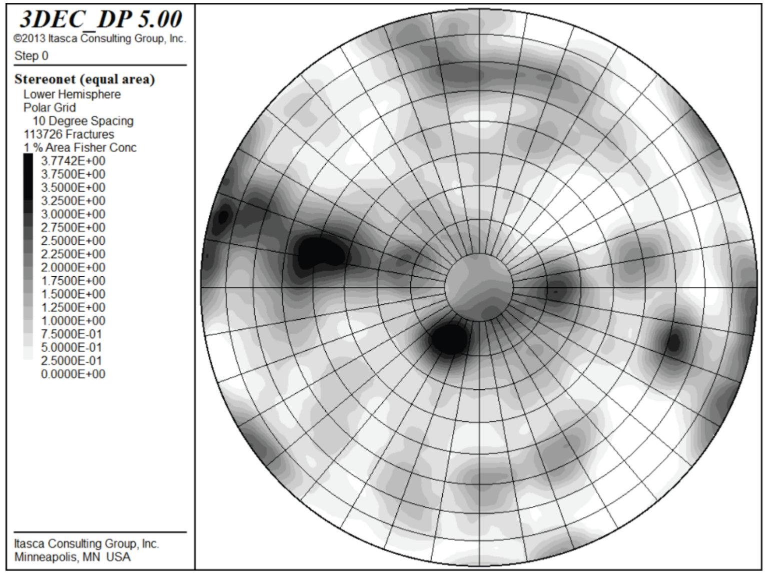
- compressible or incompressible fluid flows across any pore volume between particles.

- two different flow equations: one with permeability=f(pore size) for intact matrix, another 3DEC-like for (new or pre-existing) cracks

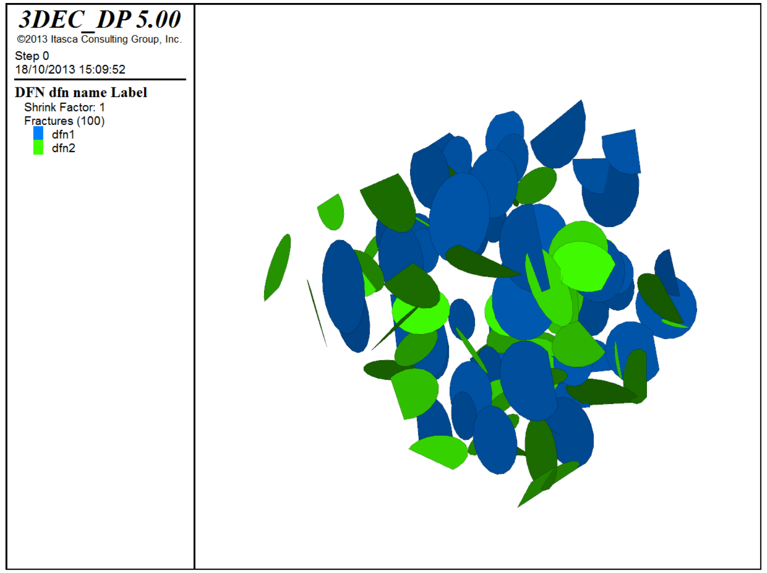
=> Timos' talk on wednesday

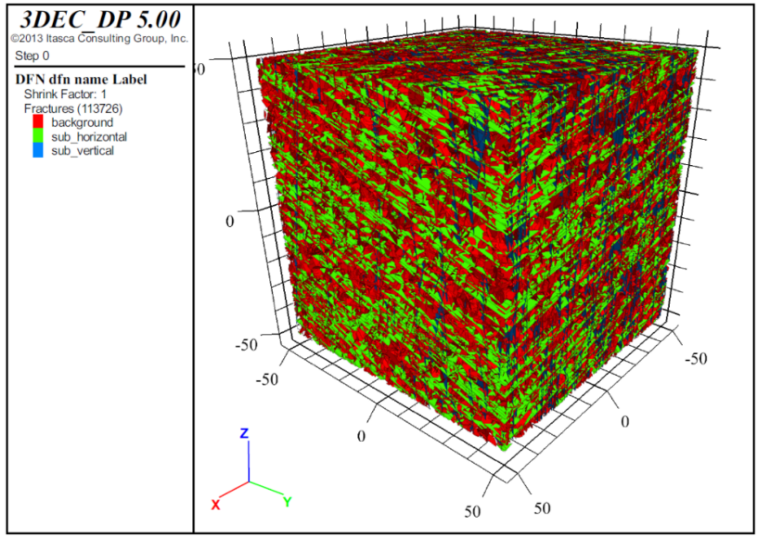
III. Hydro-mechanics for fluid extraction/injection processes

Stochastic fracture network generation



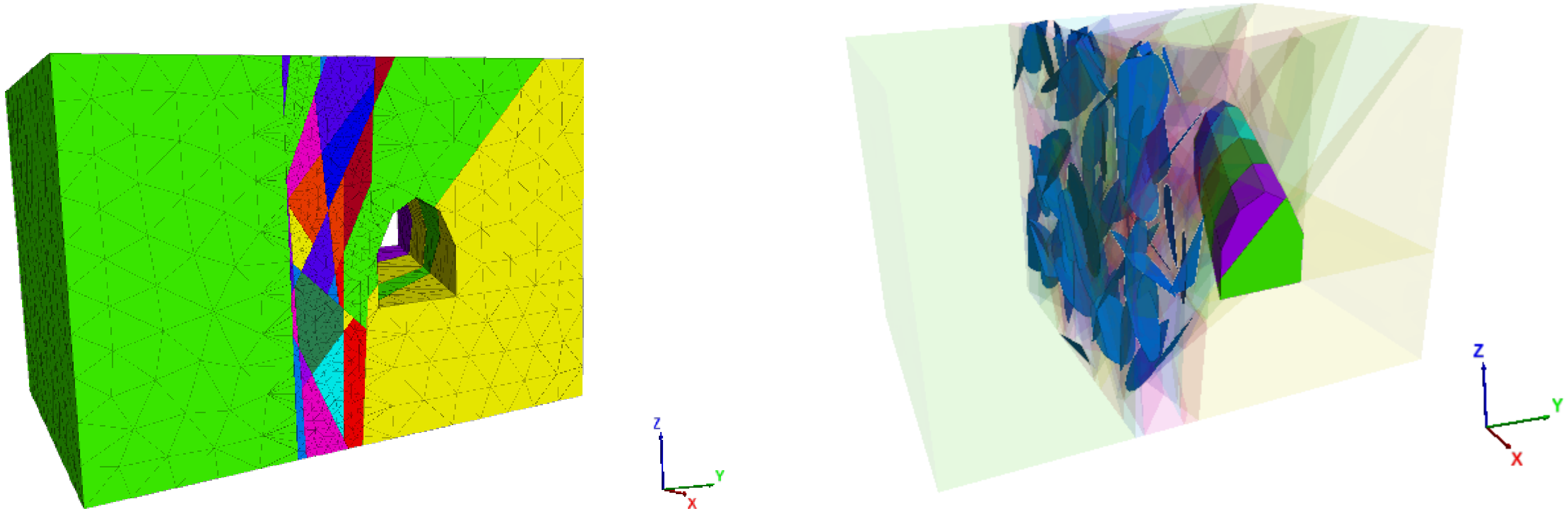
Stereographic representation of fracture surfaces according to their orientation





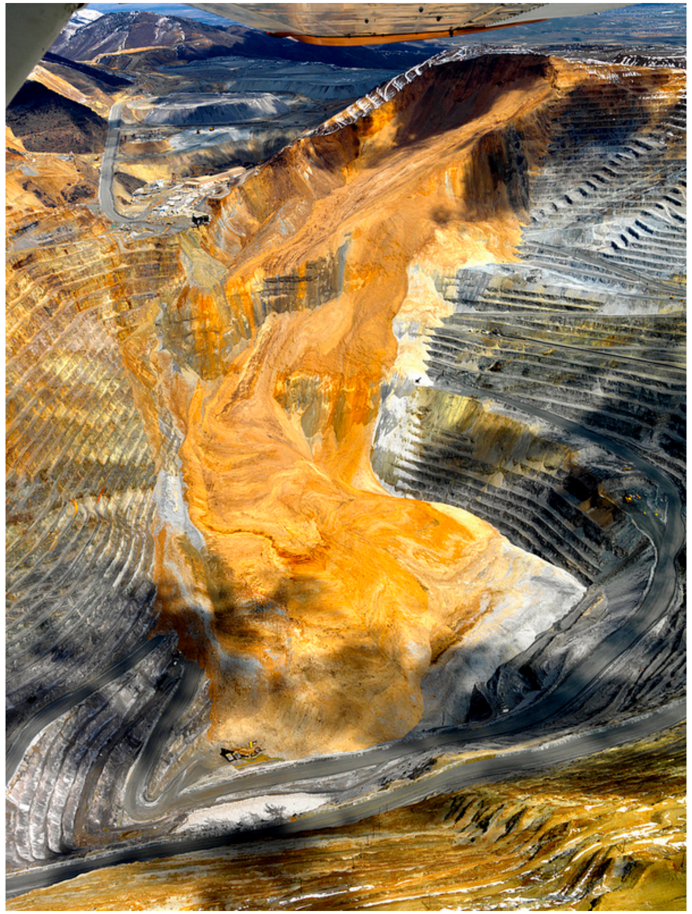
Various fracture networks in 3DEC

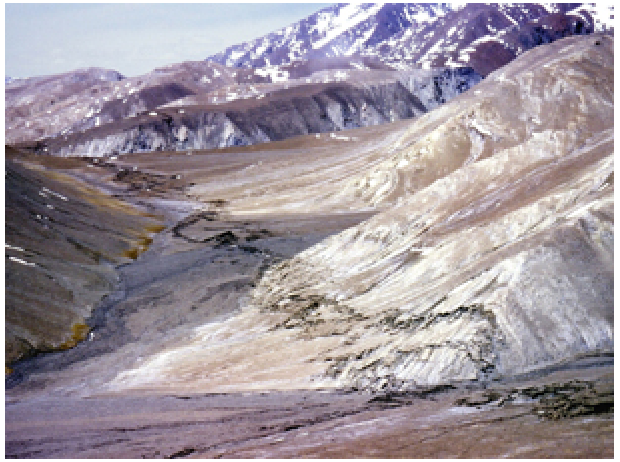
Flow in a fracture network

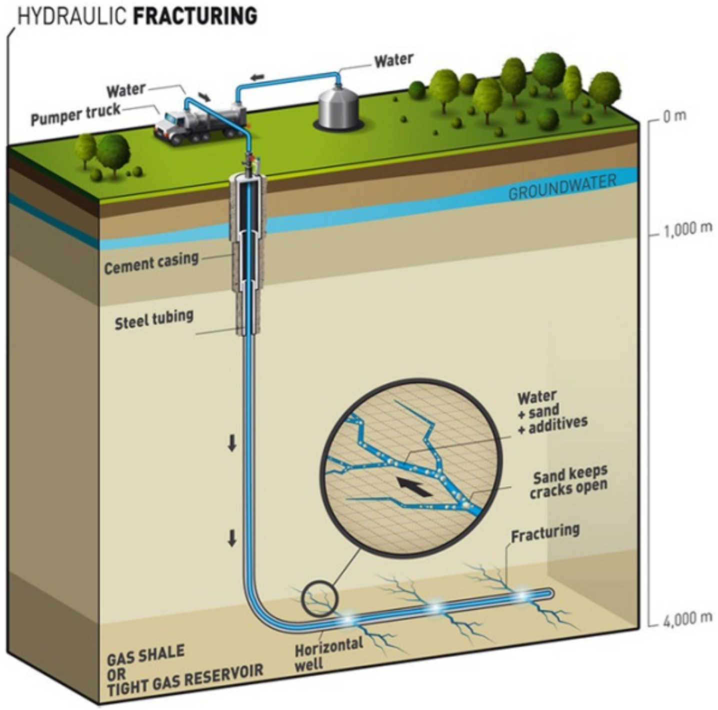


Tournemire tunnel model

Thank you for your attention !







Discrete Element modelling of fractured rock

Jérôme Duriez[1], Frédéric-Victor Donzé[1], Luc Scholtès[2]

jerome.duriez@3sr-grenoble.fr

[1] 3SR, Université Grenoble-Alpes, CNRS
[2] GeoRessources, Université de Lorraine, CNRS

Funded by ANR GeoSMEC

1st YADE workshop, Grenoble, July 8th 2014

Why studying rock ?

* Rock mass deformations may affect human societies

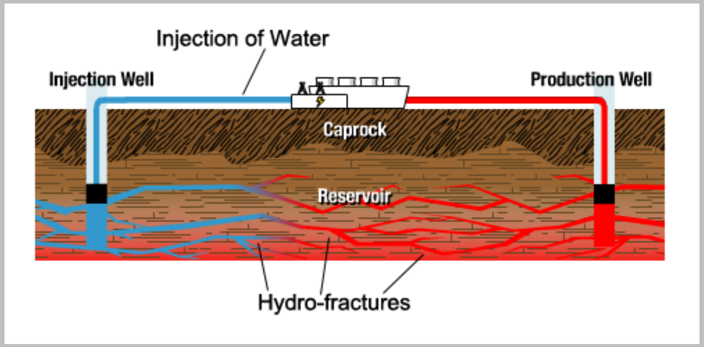
from above

Rockfall in Vercors, France (Géolithe)

from below

civildefence.govt.nz

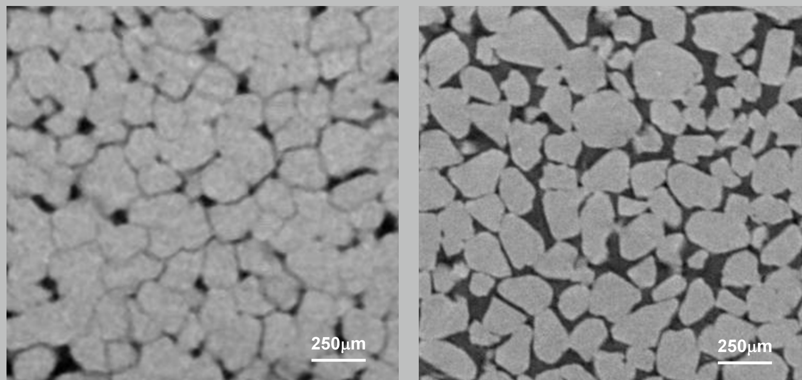
* Rock hides ressources considered as valuable



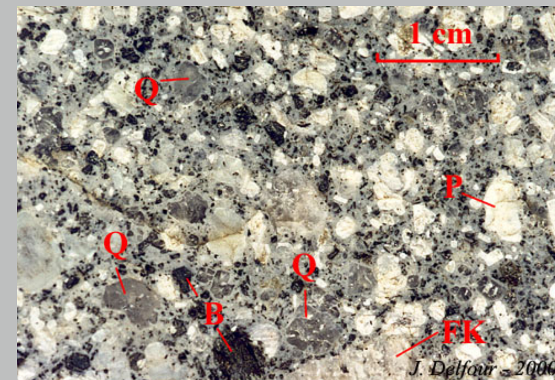
A geothermal system (http://www.renewablegreenenergypower.com)

What is rock ?

A cohesive assembly of mineral grains



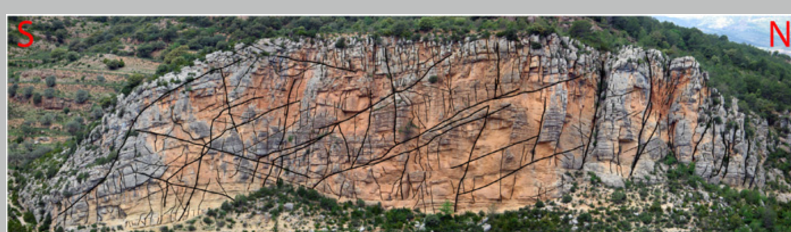
Tomograph scans of Fontainebleau sandstones
(Fonseca et al., 2013)

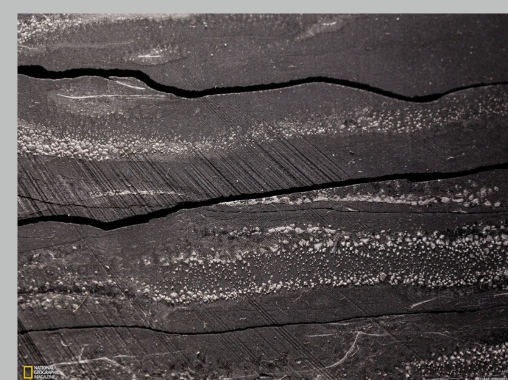


Granite (J. Delfour)

<u>DEM modelling of rock matrix ?</u>

With also macro-scale discontinuities, that may pre-exist or appear under loading





<u>DEM modelling of rock fractures ?</u>

I. DEM modelling of rock matrix

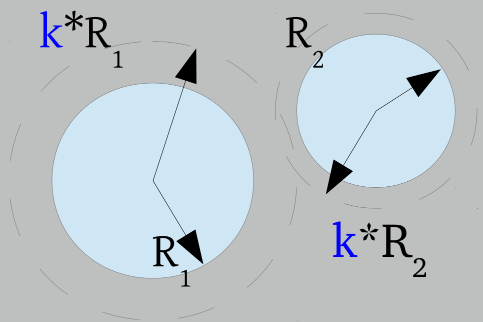
(Jointed) Cohesive Frictional Particle Model (JCFpm) in Yade

~ contact bond variant of Potyondy's model (Potyondy & Cundall, 2004)

* Definition of interacting spheres: ≠ Potyondy

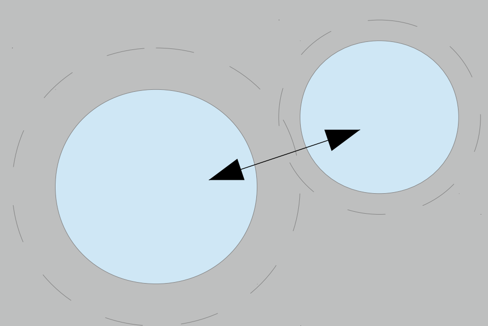
adequate (~ 10) σ_c/σ_t = UCS/UTS ratio
(Scholtès & Donzé, 2013)

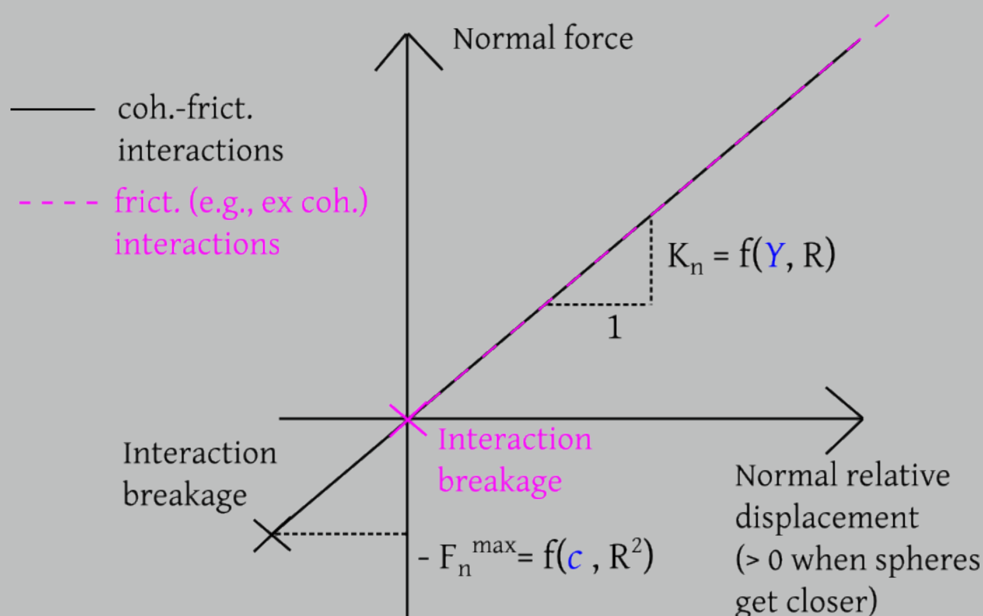
Aternative = use of clumps (Cho, 2007)



Spherical particles with controled range interactions
=> cohesive interactions
New frictional interactions depending on direct overlap

* Normal interactions

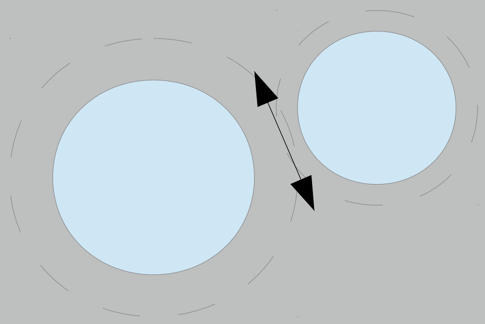


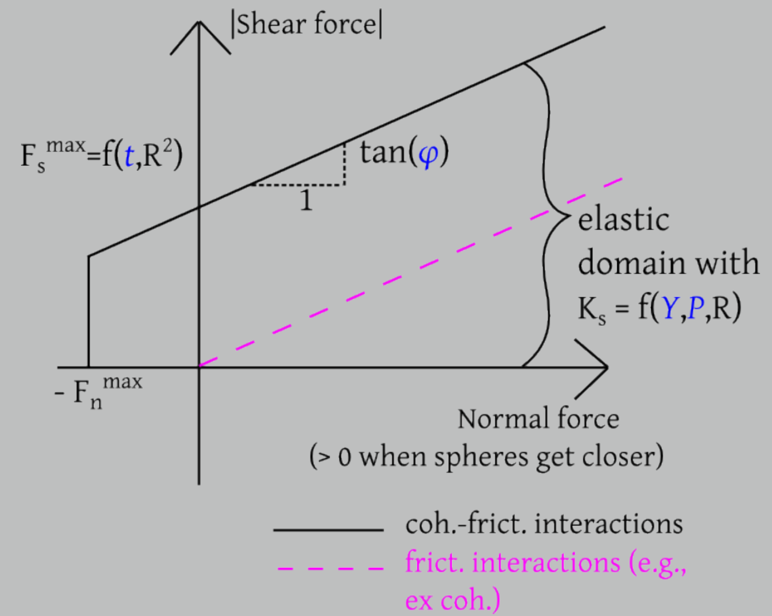


I. DEM modelling of rock matrix

(Jointed) Cohesive Frictional Particle Model (JCFpm) in Yade

* Tangential interactions

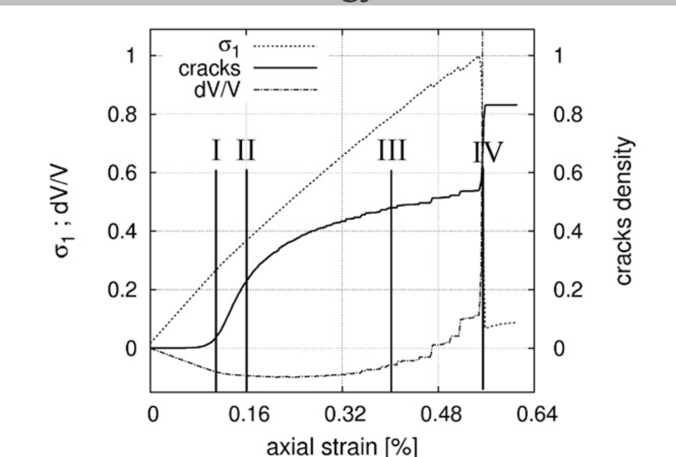




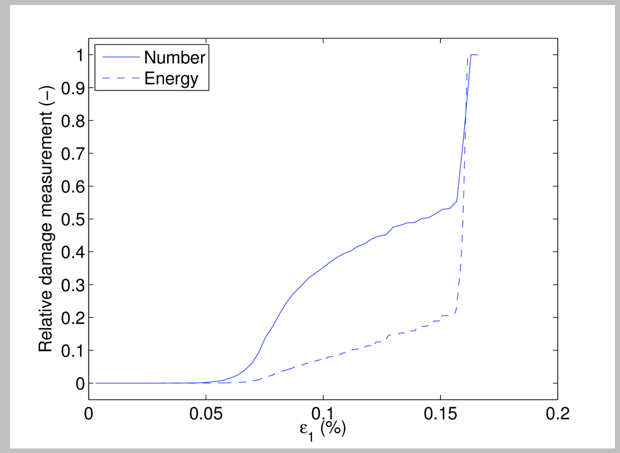
* No moment transfer law

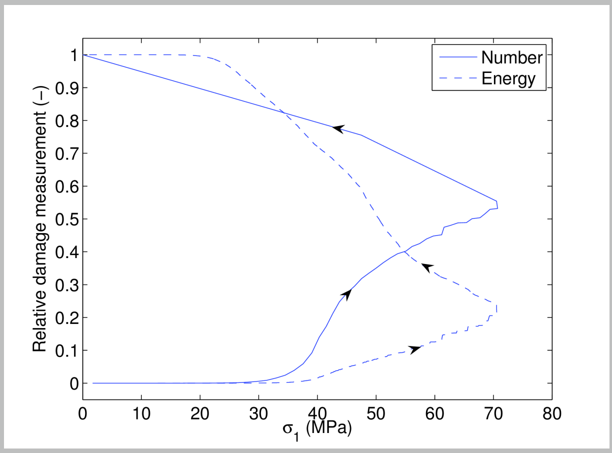
I. DEM modelling of rock matrix

Monitor of damage through cohesive interactions brittle rupture: number, or elastic energy loss ?



Density (relative number) of broken interactions during simple compression simulation (Scholtès, 2013)

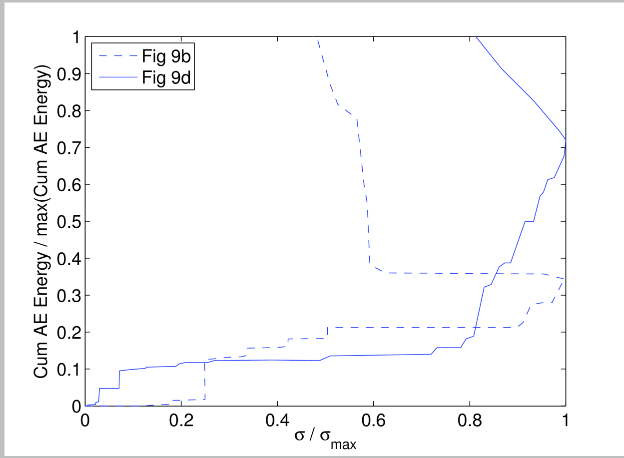




Number or energy data for the interactions broken during a simple compression simulation

=> different insights into damage mechanisms

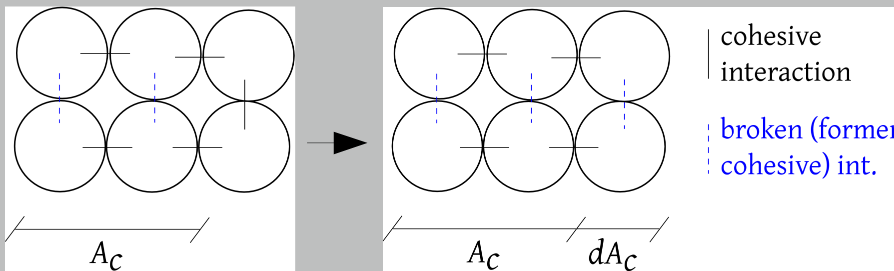
Experimental evidences ?



AE energy data, after data from (Wassermann, 2009)

I. DEM modelling of rock matrix

Size element influence ?

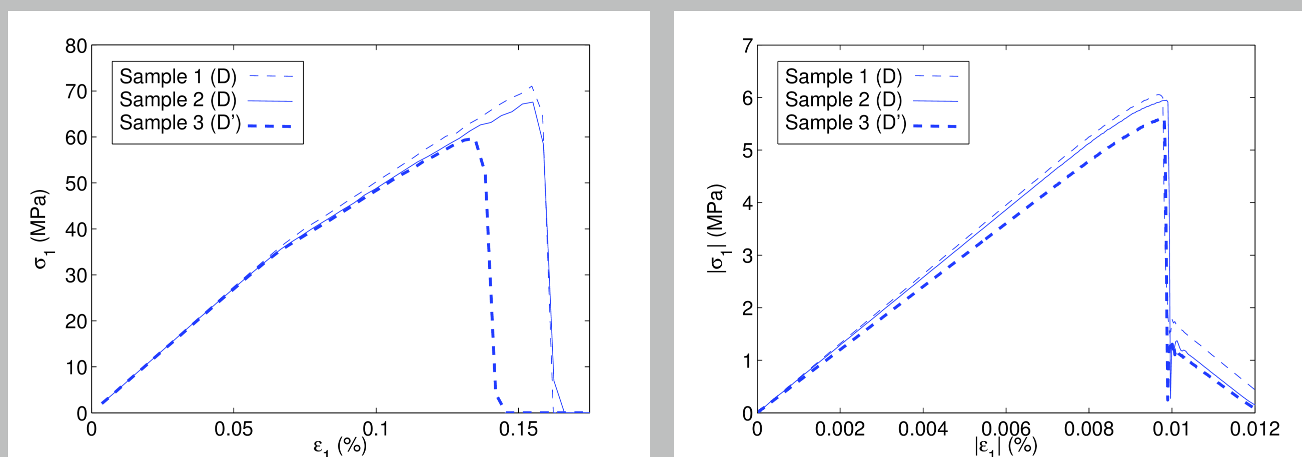


Crack propagation in a DEM model

$F^n_{max} = f(\boldsymbol{c}, R^2)$ $\quad$ $Kn = f(\boldsymbol{Y}, R)$

=> $G_c = dE / dA_c$ prop to : $\boldsymbol{c}^2 / \boldsymbol{Y} * D$

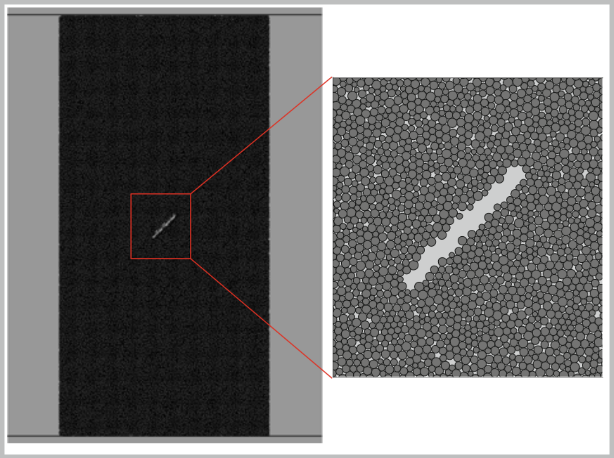
<=> K_I^c prop to sqrt(D) (Potyondy, 2004)



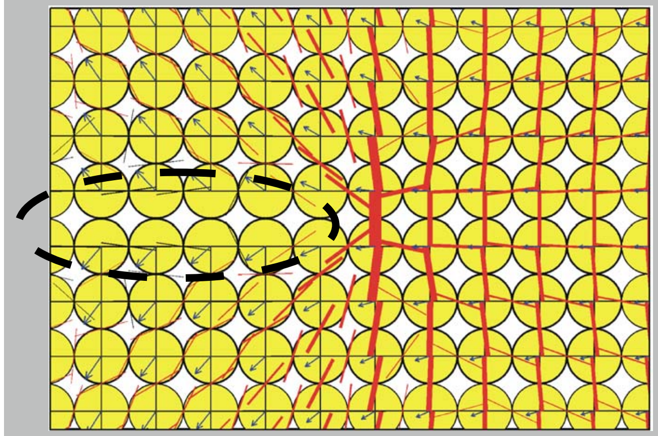
DE size influence for compressive (left) or tensile (right) tests. D' ~ D / 2
All samples have more than 10 000 elements

II. DEM modelling of rock fractures

Discrete modelling of pre-existing rock fractures ?



(*Zhang & Wong, 2012*)



(*Potyondy & Cundall, 2004*)

DE removed

=> open cracks

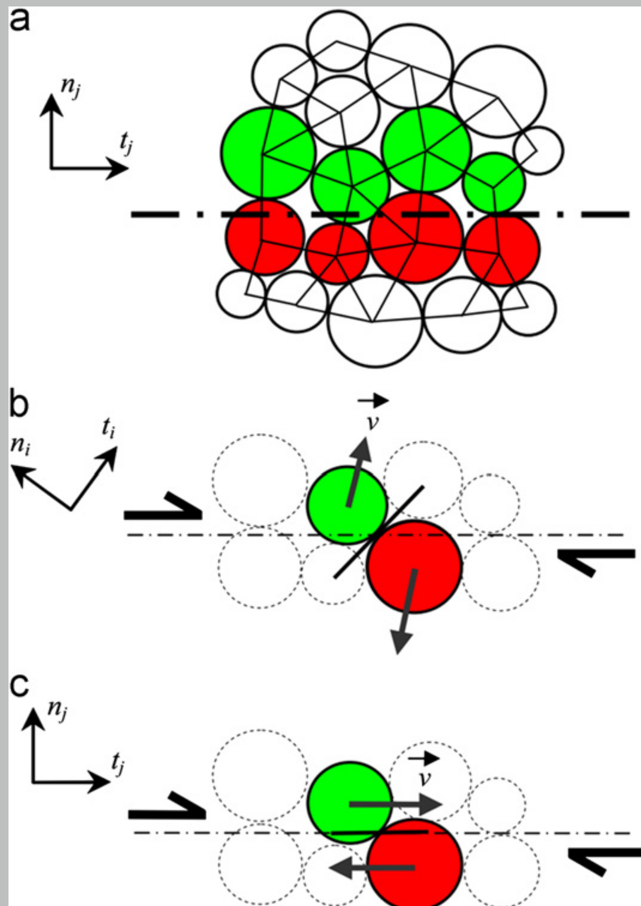
Cohesive links between DE removed

=> closed cracks
=> influence of the DE assembly (roughness) ?

closed cracks without influence of DE assembly ??

II. DEM modelling of rock fractures

The smooth joint model *(SJM, Ivars et al., 2008)*: Jointed (Cohesive Frictional Particle Model) in Yade



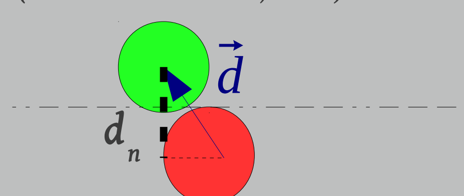
(Scholtès & Donzé, 2012)

DE assembly along a discontinuity surface (pre existing crack !)

classical DE interaction accross this surface : defined by (n_i,t_i)

SJM DE interaction accross this surface : (n_i,t_i) reoriented to (n_j,t_j)

d



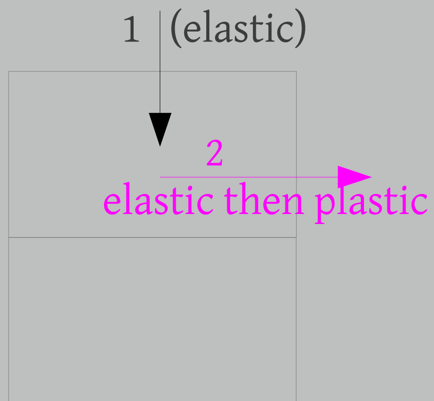
SJM DE interaction accross this surface : normal force computed according to $\boldsymbol{d}_n$, instead of $\|\vec{d}\|$

II. DEM modelling of rock fractures

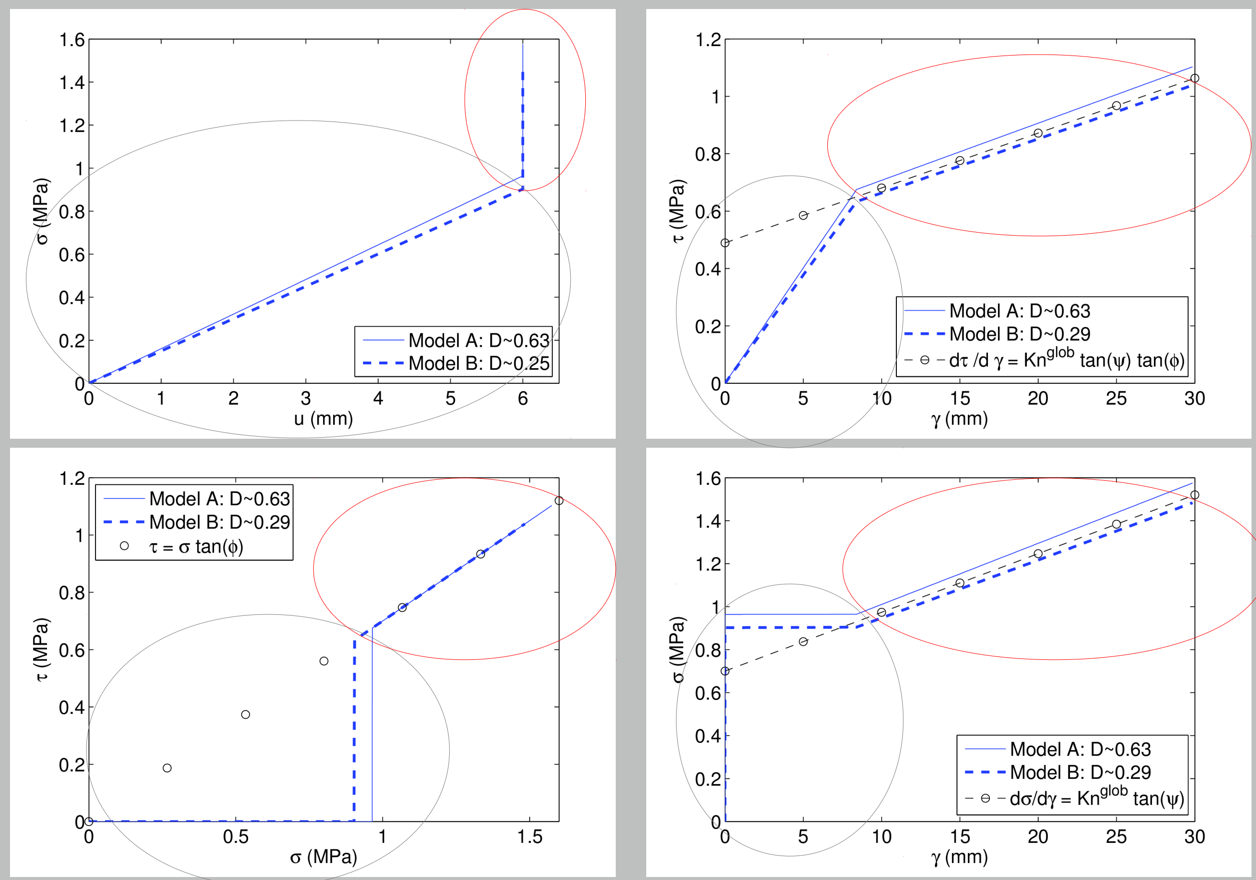
Contact laws for joint interactions

$K_n = f(\boldsymbol{K_n^J}, R^2)$

$K_s = f(\boldsymbol{K_s^J}, R^2)$



Sliding defined by friction angle $\boldsymbol{\phi^J}$, and dilatancy angle $\boldsymbol{\psi^J}$



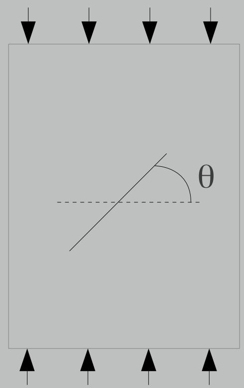
Behaviour of two rock joint models, with different DE sizes

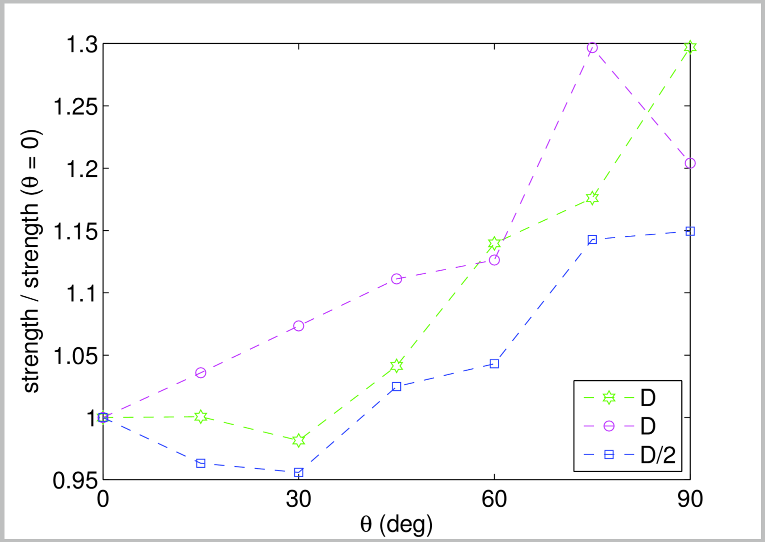
elastic macro-properties ↔ $\boldsymbol{K_n^J}, \boldsymbol{K_s^J}$, porosity of the packing, ~~element size~~

plastic macro-properties = $\boldsymbol{\phi^J}, \boldsymbol{\psi^J}$

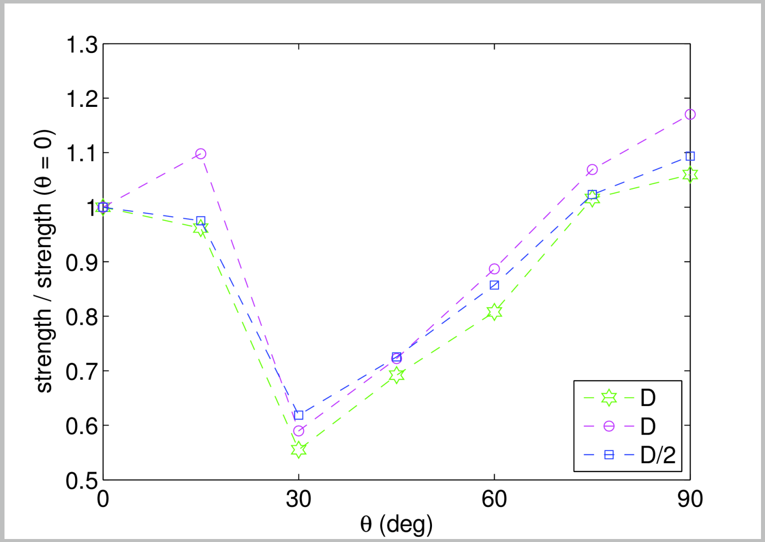
II. DEM modelling of rock fractures

SJM – JCFpm vs other approaches

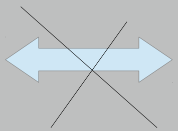


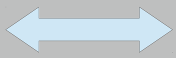


Discrete modelling removing cohesive links along the flaw



Discrete modelling removing cohesive links along the flaw ***and using the smooth joint model***

description of the flaw  properties of the flaw

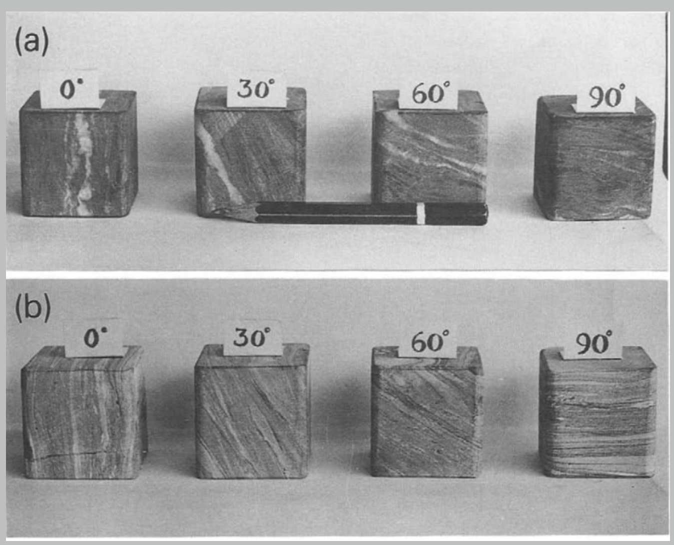
description of the flaw  properties of the flaw

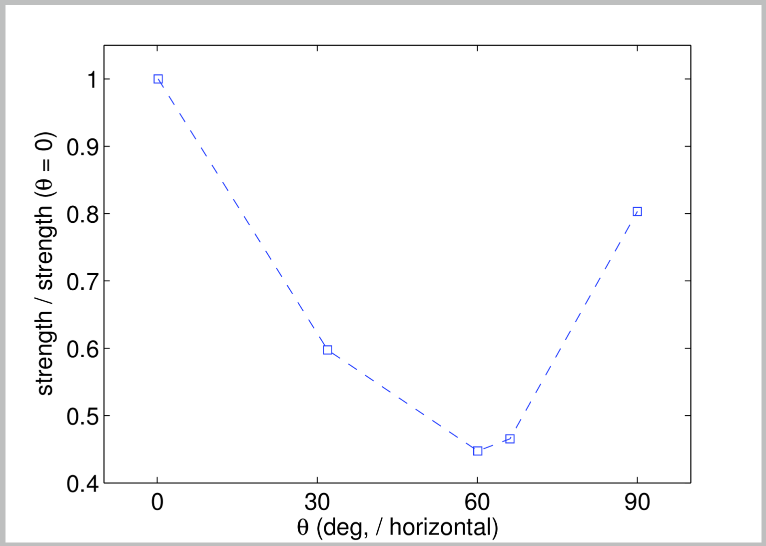
Conclusions & Perspectives

JCFpm model appears as an adequate discrete model for fractured rock, with limited numerical biais.

It allows to focus on crack mechanical properties to tackle different rock mechanics problems (cf yesterday afternoon).

It could be used to describe the anisotropy of foliated rocks ?





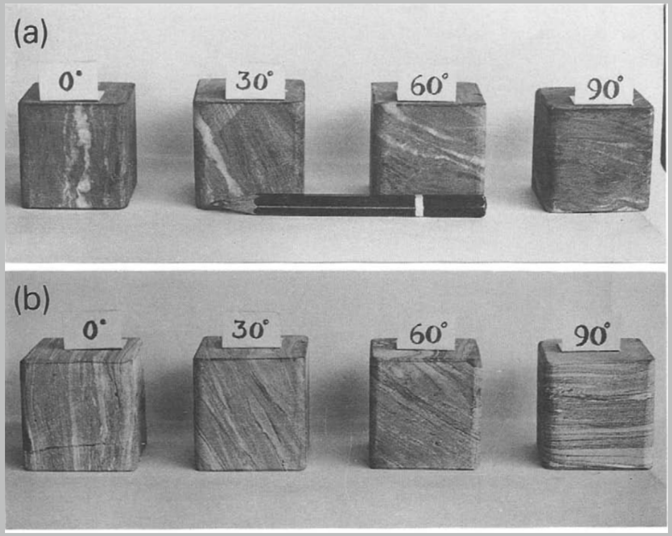
Foliated specimens of phyllite (Ramamurthy, 1993)

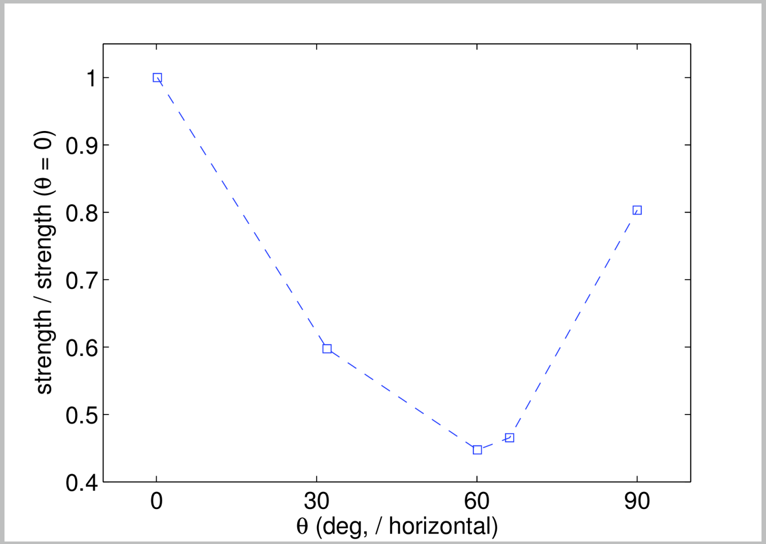
Thank you for your attention !

JCFpm model appears as an adequate discrete model for fractured rock, with limited numerical biais.

It allows to focus on crack mechanical properties to tackle different rock mechanics problems (cf yesterday afternoon).

It could be used to describe the anisotropy of foliated rocks ?





Foliated specimens of phyllite (Ramamurthy, 1993)

Particle and Nodes: logical separation

Václav Šmilauer

woodem.eu ib-keramik.de

July 2014

Abstract

WooDEM treats nodes (e.g. for motion integration) as separate entities from particles (for collision detection and contacts). This talk summarizes the conveniences and inconveniences of this approach.

What is Woo?

[dem]

Custom Discrete Element Solutions

- open-source fork of Yade, started in 2011 (Uni Innsbruck)
  - I was breaking Yade all the time
  - low-quality code around in Yade (IMHO)
  - non-DEM needed: meshfree – the finite pointset method, GPU (experimental)
- most code revised and rewritten from scratch
- tighter Python integration, easier scripting, portability
- funding model: open-source with contracts for customizations
- emphasis on industrial processes (segregation, sieving, ...; PSD factories, analyses)
- non-research stuff: reporting, UI, docs
- dev: generic particle shapes (potential particles), Xeon Phi support, cloud computing

Particles in Yade

Yade conflates particles as:

1. colliding (bboxes)
2. having contacts (shapes)
3. undergoing motion (mass, velocity, ...)

— Simple in code, fine for pure DEM...

Latent issues

- connectivity (meshes, cables)

Particles in Woo

Node concept

- borrowed from FEM
- assures C_n continuity accross elements sharing nodes
- important for deformable elements (rods, membranes, volumes)

Split shape and motion

- shape positioned by one *or more* associated nodes
- nodes are shared (mesh vertices)
- possibile generalization: split shape from collision (e.g. multiple bboxes per shape)
- nodes carry different kinds of data (`DemData`, `GlData`, ...)

Disadvantages

Consistency

- particles reference their nodes
- nodes need to know about their particles

— periodic internal consistency checks

Verbosity

- typing: `particle->shape->nodes[0]->getData<DemData>().vel`

— convenience accessors in Python for uninodal particles: `p.vel`

- merely adding particle will not make it movable

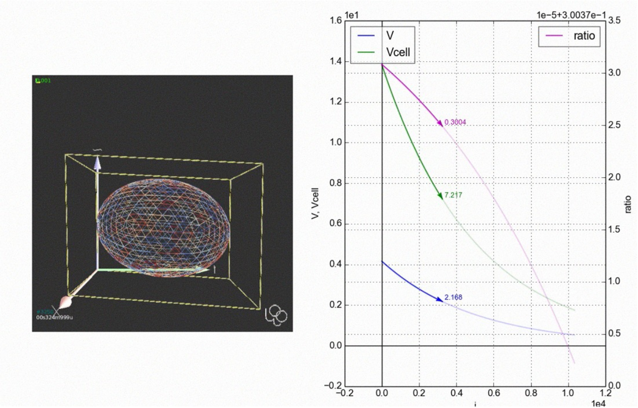
— convenience `S.dem.par.append(p); S.dem.collectNodes()`

Advantages

- connectivity (C_n continuity, as in FEM):
  - deforming membrane (separate talk)
  - static mesh moving with periodic space
  - computing closed mesh volume
- anything having a node can move – e.g. engines
- clumps are not special particles
- `Node` defines local coordinates, e.g. contacts

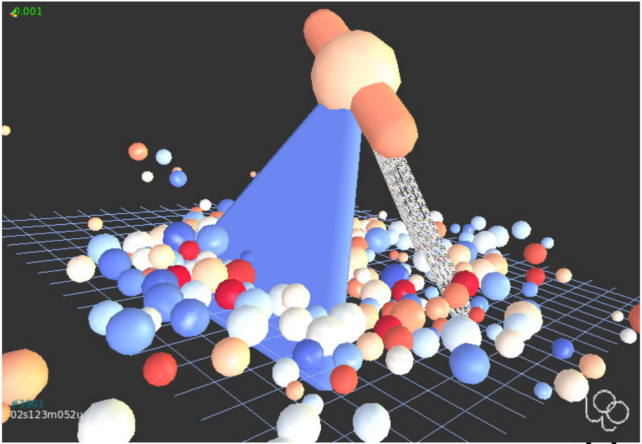
Example: computing mesh volume (mesh-volume.py)

The computation is fast since connectivity is known via nodes shared between neghboring facets.



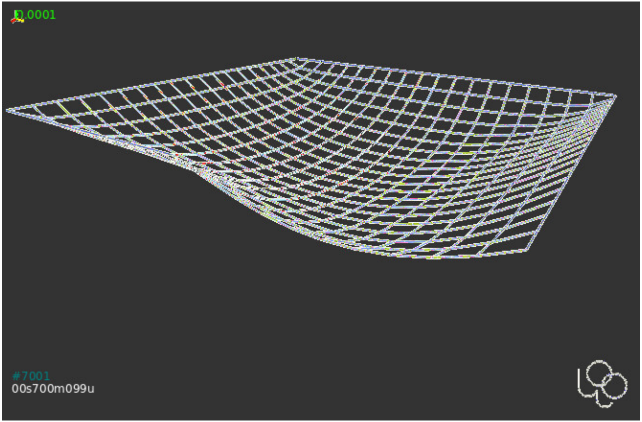
Example: node shared between many particles (truss-facet.py))

The moving node is shared between facets, sphere, capsule, trusses, and was prescribed linear and angular velocity.



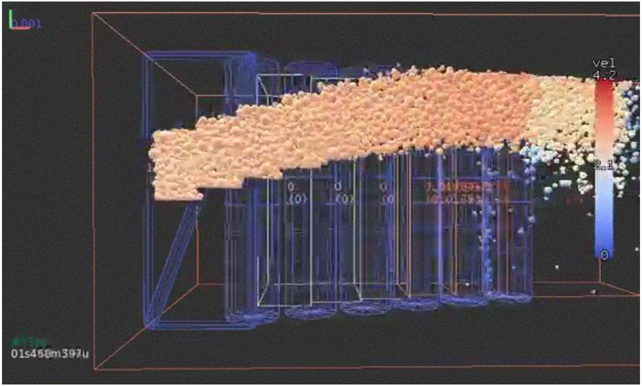
Example: cloth simulated with trusses (truss-cloth.py)

This shows trusses as 2-node particles; they have their own internal force due to strain of each single truss (the simplest finite element with only 1 degree of freedom); more on this concept in the membrane talk.



Example: feed moving

The feed (ConveyorFactory) is moving because its position is defined via a node. The node has harmonic motion prescribed.



Extended use of periodic boundary conditions

Jan Stránský, Martin Doškář, Jan Novák

Czech Technical University in Prague
Faculty of Civil Engineering
Department of Mechanics

Outline

Outline

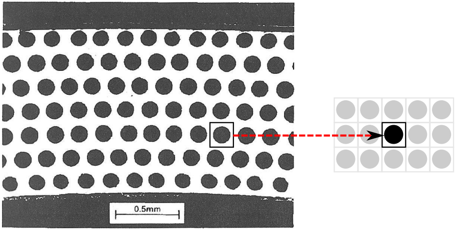
Outline

Outline

1. Introduction
2. Theory
3. Example
4. Conclusion

Motivation

- Modelling of heterogeneous materials in efficient way



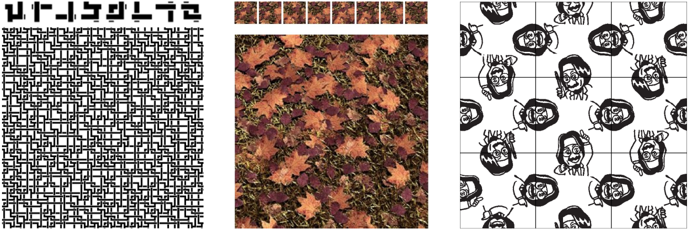
[M. ŠEJNOHA, Ph.D. thesis, 1996]

PUC → SEPUC → extension?

Possible advantages

- Reducing periodicity in reconstructed representation
- Modelling of materials without clearly separated scales

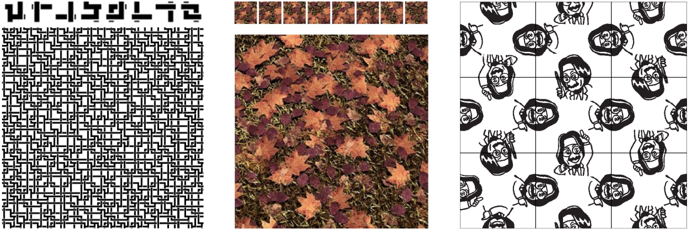
Wang tilings



[COHEN A KOL., ACM Trans. Graph., 2003], [DEMAINE A KOL., Graph. Combinator., 2007]

- Entscheidungproblem (D. Hilbert, 1928) → Halting problem (A. Turing, 1936) → Domino problem (H. Wang, 1961)
- The smallest aperiodic sets (104 by R. Berger → 13 by K. Čulík)
- Another applications- modelling of quasi-crystals and DNA structures
- Computer graphics- *naturally looking textures*

Wang tilings

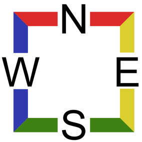


[COHEN A KOL., ACM Trans. Graph., 2003], [DEMAINE A KOL., Graph. Combinator., 2007]

- Entscheidungproblem (D. Hilbert, 1928) $\rightarrow$ Halting problem (A. Turing, 1936) $\rightarrow$ Domino problem (H. Wang, 1961)
- The smallest aperiodic sets (104 by R. Berger $\rightarrow$ 13 by K. Čulík)
- Another applications- modelling of quasi-crystals and DNA structures
- Computer graphics- *naturally looking textures*

Wang tiles and sets

- Wang tile $\approx$ square dominoe like piece with coded edges



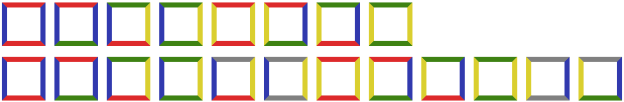
Tiles can not be neither rotated nor reflected during tiling procedure
mapping $\tau : \mathbb{A} \times \mathbb{B} \rightarrow \mathcal{T}$, where $\mathbb{A}, \mathbb{B} \subset \mathbb{N}$

- Wang tile set $\mathcal{T}$

$$Wn^t / n_1^c - n_2^c$$

W8/2-2

W12/2-3

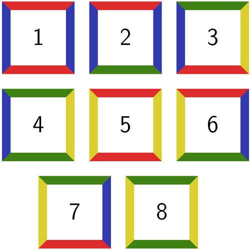


Tiling algorithms

- Aperiodic tiling

$$\tau(i+\alpha, j+\beta) = \tau(i,j) \quad \forall i \in \mathbb{A}, j \in \mathbb{B} \quad \Leftrightarrow \quad \alpha = \beta = 0$$

- Stochastic tiling [Cohen a kol., ACM, 2003]



Stochasticity ensured with:

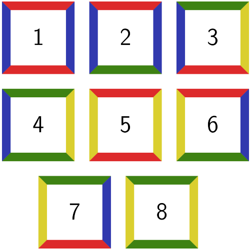
- Random number generator
- Existence of at least 2 tiles for each admissible NW code combination

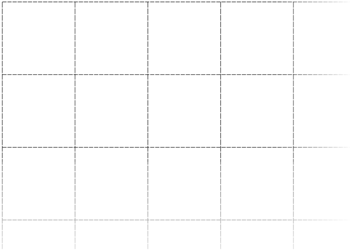
Tiling algorithms

- Aperiodic tiling

$$\tau(i+\alpha, j+\beta) = \tau(i,j) \quad \forall i \in \mathbb{A}, j \in \mathbb{B} \quad \Leftrightarrow \quad \alpha = \beta = 0$$

- Stochastic tiling [Cohen a kol., ACM, 2003]





Stochasticity ensured with:

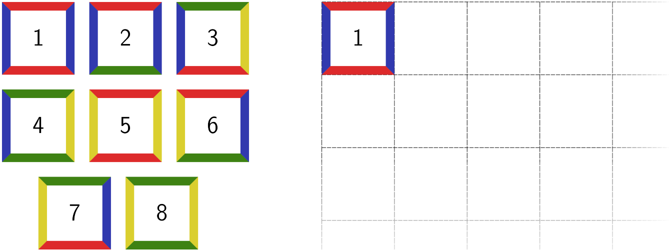
- Random number generator
- Existence of at least 2 tiles for each admissible NW code combination

Tiling algorithms

- Aperiodic tiling

$$\tau(i+\alpha, j+\beta) = \tau(i,j) \quad \forall i \in \mathbb{A}, j \in \mathbb{B} \quad \Leftrightarrow \quad \alpha = \beta = 0$$

- Stochastic tiling [Cohen a kol., ACM, 2003]



Stochasticity ensured with:

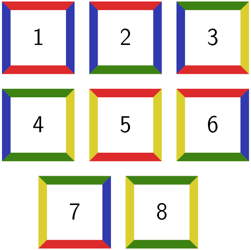
- Random number generator
- Existence of at least 2 tiles for each admissible NW code combination

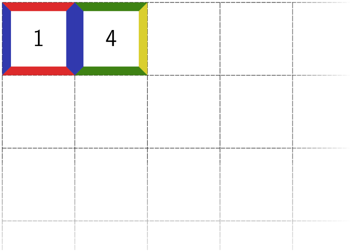
Tiling algorithms

- Aperiodic tiling

$$\tau(i+\alpha, j+\beta) = \tau(i,j) \quad \forall i \in \mathbb{A}, j \in \mathbb{B} \quad \Leftrightarrow \quad \alpha = \beta = 0$$

- Stochastic tiling [Cohen a kol., ACM, 2003]





Stochasticity ensured with:

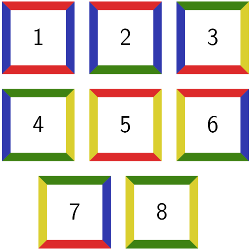
- Random number generator
- Existence of at least 2 tiles for each admissible NW code combination

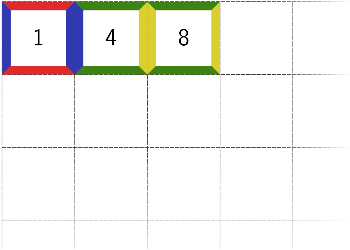
Tiling algorithms

- Aperiodic tiling

$$\tau(i+\alpha, j+\beta) = \tau(i,j) \quad \forall i \in \mathbb{A}, j \in \mathbb{B} \quad \Leftrightarrow \quad \alpha = \beta = 0$$

- Stochastic tiling [Cohen a kol., ACM, 2003]





Stochasticity ensured with:

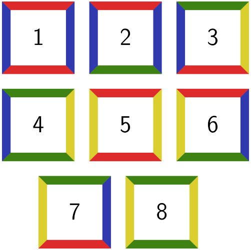
- Random number generator
- Existence of at least 2 tiles for each admissible NW code combination

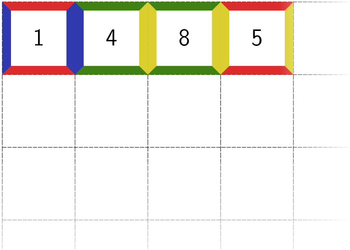
Tiling algorithms

- Aperiodic tiling

$$\tau(i+\alpha, j+\beta) = \tau(i,j) \quad \forall i \in \mathbb{A}, j \in \mathbb{B} \quad \Leftrightarrow \quad \alpha = \beta = 0$$

- Stochastic tiling [Cohen a kol., ACM, 2003]





Stochasticity ensured with:

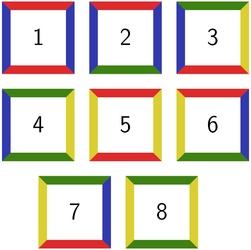
- Random number generator
- Existence of at least 2 tiles for each admissible NW code combination

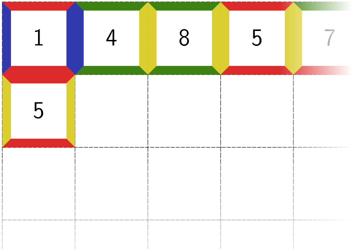
Tiling algorithms

- Aperiodic tiling

$$\tau(i+\alpha, j+\beta) = \tau(i,j) \quad \forall i \in \mathbb{A}, j \in \mathbb{B} \quad \Leftrightarrow \quad \alpha = \beta = 0$$

- Stochastic tiling [Cohen a kol., ACM, 2003]





Stochasticity ensured with:

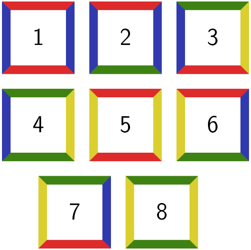
- Random number generator
- Existence of at least 2 tiles for each admissible NW code combination

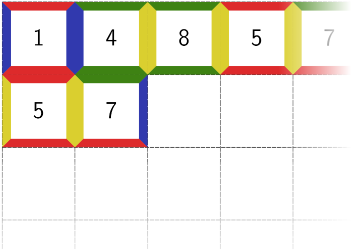
Tiling algorithms

- Aperiodic tiling

$$\tau(i+\alpha, j+\beta) = \tau(i,j) \quad \forall i \in \mathbb{A}, j \in \mathbb{B} \quad \Leftrightarrow \quad \alpha = \beta = 0$$

- Stochastic tiling [Cohen a kol., ACM, 2003]





Stochasticity ensured with:

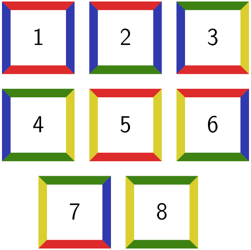
- Random number generator
- Existence of at least 2 tiles for each admissible NW code combination

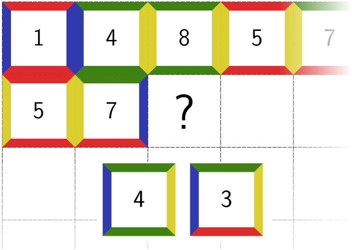
Tiling algorithms

- Aperiodic tiling

$$\tau(i+\alpha, j+\beta) = \tau(i,j) \quad \forall i \in \mathbb{A}, j \in \mathbb{B} \quad \Leftrightarrow \quad \alpha = \beta = 0$$

- Stochastic tiling [Cohen a kol., ACM, 2003]



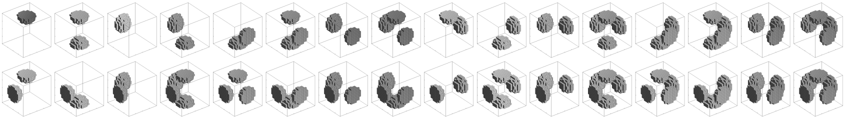


Stochasticity ensured with:

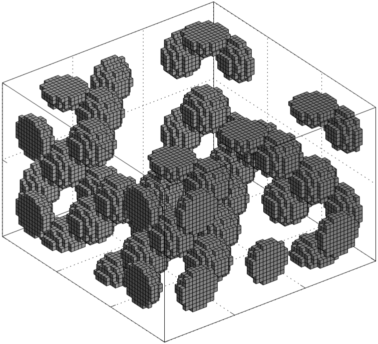
- Random number generator
- Existence of at least 2 tiles for each admissible NW code combination

Stochastic Wang Cubes

- Wang cubes set W32/2-2-2

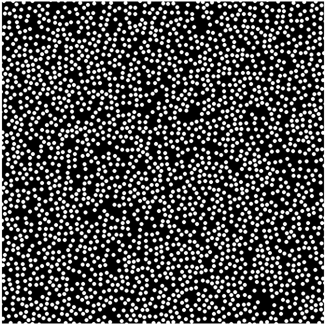


- example of 3 × 3 × 2 tiling

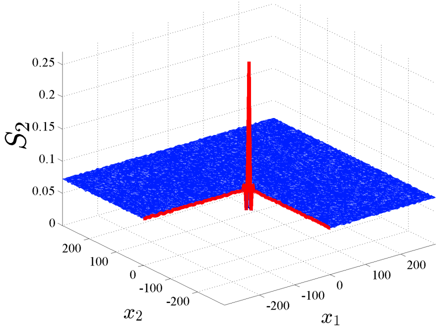


Parasitic long range orientation order artefacts

- Quantified by means of two-point probability function $S_2(\boldsymbol{x})$

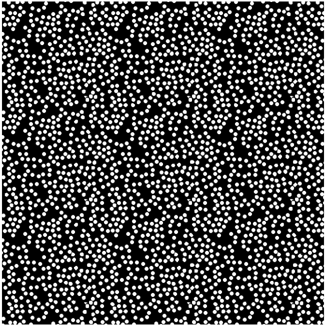


Reference microstructure

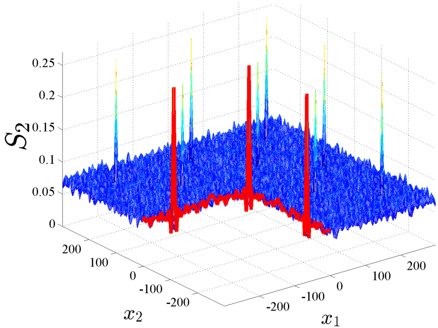


Parasitic long range orientation order artefacts

- Quantified by means of two-point probability function $S_2(\boldsymbol{x})$

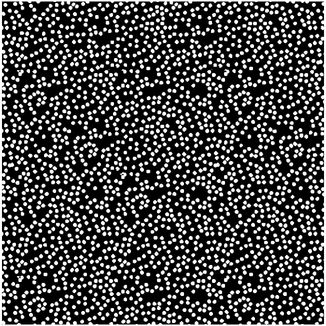


W1/1-1 (PUC)

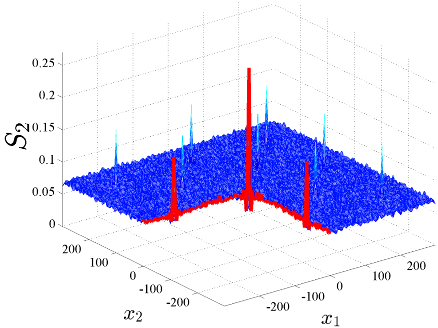


Parasitic long range orientation order artefacts

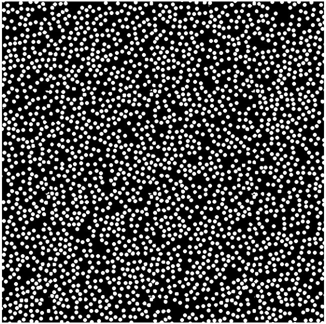
- Quantified by means of two-point probability function $S_2(\boldsymbol{x})$



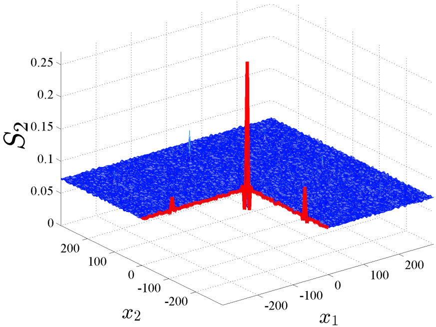
W8/2-2



- Quantified by means of two-point probability function $S_2(\boldsymbol{x})$



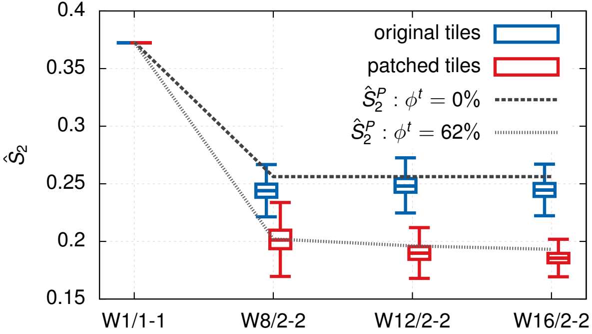
W8/2-2 + patches



Parasitic long range orientation order artefacts

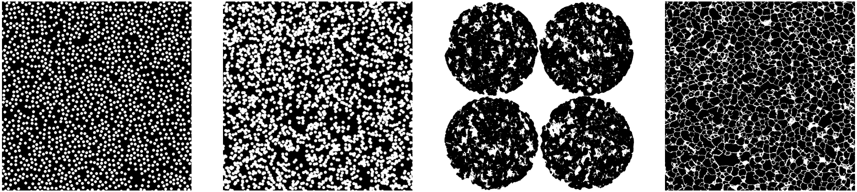
- Prediction [Novák et al. , PRE, 2012]

$$\hat{S}_2^P \approx \frac{\Phi^t}{n^t}\left[\Phi + (n^t - 1)\Phi^2\right] + \max_i \left\{ \frac{\Phi^e}{n_i^c}\left[\Phi + (n_i^c - 1)\Phi^2\right] \right\}$$

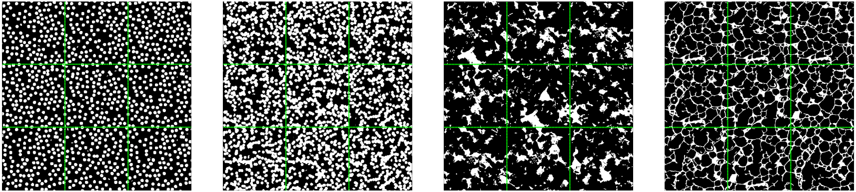


Comparison of reference and reconstructed media

- Target systems:



- Reconstructed systems (3 $\times$ 3 tiles):



Periodic packing

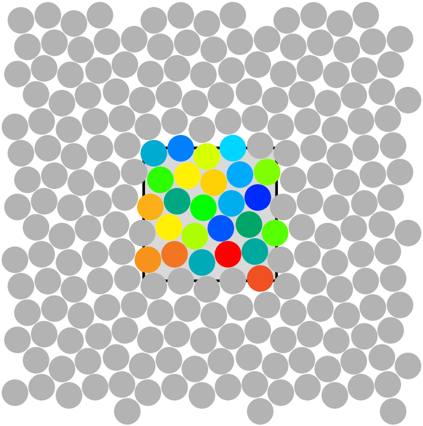
Periodic contact detection

- interactions found also with periodic images of other particles
- information about periodicity stored in interactions

Periodic packing

Periodic contact detection

- interactions found also with periodic images of other particles
- information about periodicity stored in interactions



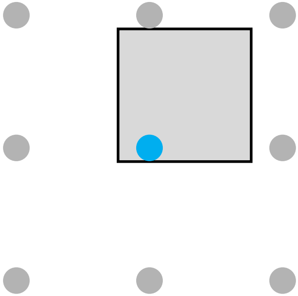
Periodic packing

Periodic contact detection

- interactions found also with periodic images of other particles
- information about periodicity stored in interactions

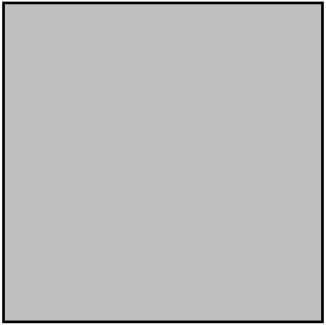
Packing algorithm

while (no overlaps) {
    save state;
    insert randomly one particle;
    rleax to static equlibrium;
}
// now overlaps exist
load last saved state;
// where no overlap exists



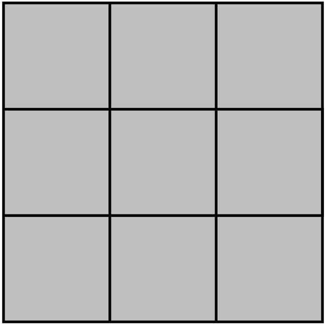
“Semi-periodic” packing

- particles put into groups (marked with group bitmask)
- only particles with compatible masks interact (mask1 & mask2)



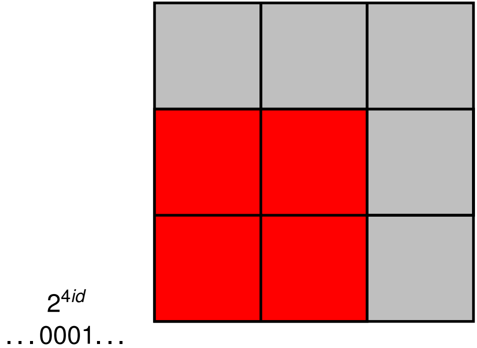
"Semi-periodic" packing

- particles put into groups (marked with group bitmask)
- only particles with compatible masks interact (mask1 & mask2)



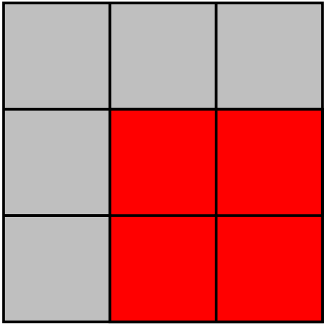
“Semi-periodic” packing

- particles put into groups (marked with group bitmask)
- only particles with compatible masks interact (mask1 & mask2)



"Semi-periodic" packing

- particles put into groups (marked with group bitmask)
- only particles with compatible masks interact (mask1 & mask2)



2^{4id+1}

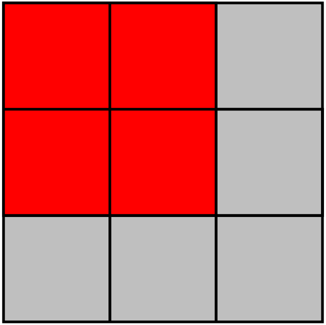
$\ldots 0010 \ldots$

“Semi-periodic” packing

- particles put into groups (marked with group bitmask)
- only particles with compatible masks interact (mask1 & mask2)

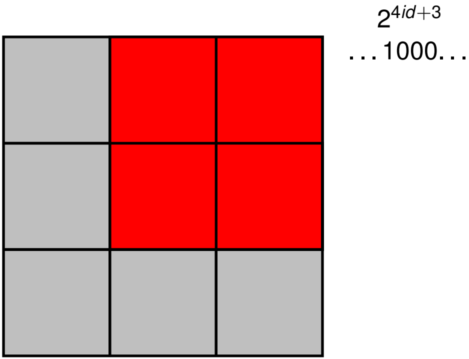
2^{4id+2}

$\ldots 0100 \ldots$



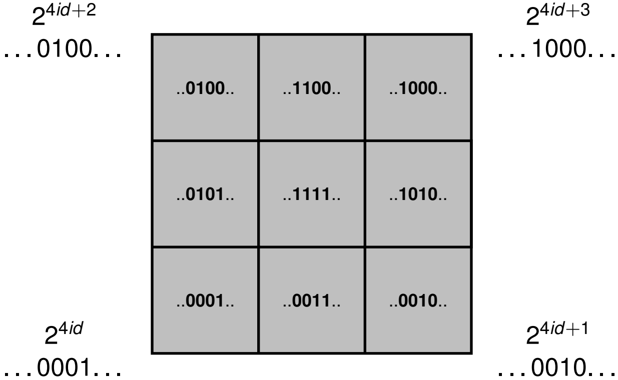
"Semi-periodic" packing

- particles put into groups (marked with group bitmask)
- only particles with compatible masks interact (mask1 & mask2)



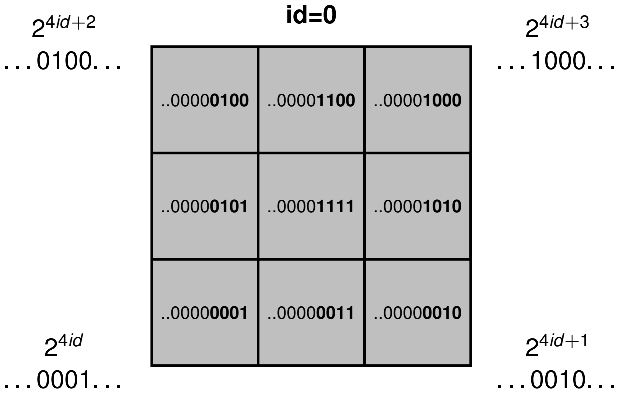
“Semi-periodic” packing

- particles put into groups (marked with group bitmask)
- only particles with compatible masks interact (mask1 & mask2)
- $\rightarrow$ periodic contact detection diabled



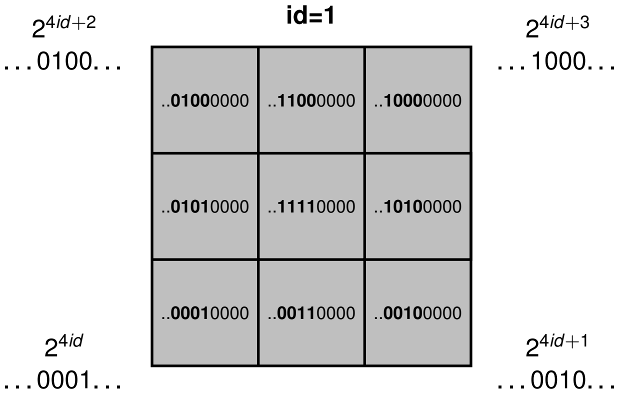
"Semi-periodic" packing

- particles put into groups (marked with group bitmask)
- only particles with compatible masks interact (mask1 & mask2)
- $\rightarrow$ periodic contact detection diabled



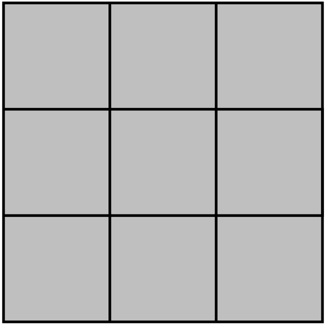
“Semi-periodic” packing

- particles put into groups (marked with group bitmask)
- only particles with compatible masks interact (mask1 & mask2)
- $\rightarrow$ periodic contact detection diabled



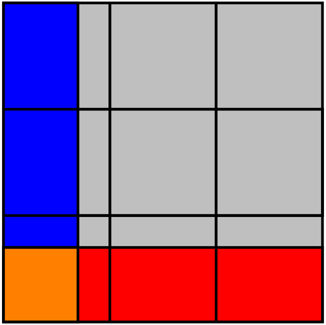
“Semi-periodic” packing

- particles put into groups (marked with group bitmask)
- only particles with compatible masks interact (mask1 & mask2)
- $\rightarrow$ periodic contact detection diabled
- boundary particles are treated in a special way



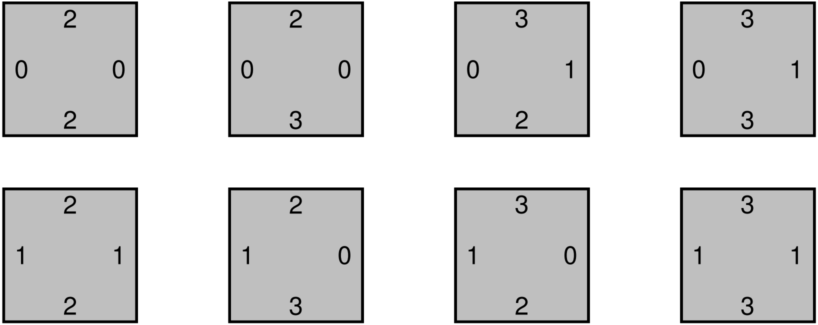
“Semi-periodic” packing

- particles put into groups (marked with group bitmask)
- only particles with compatible masks interact (mask1 & mask2)
- $\rightarrow$ periodic contact detection diabled
- boundary particles are treated in a special way



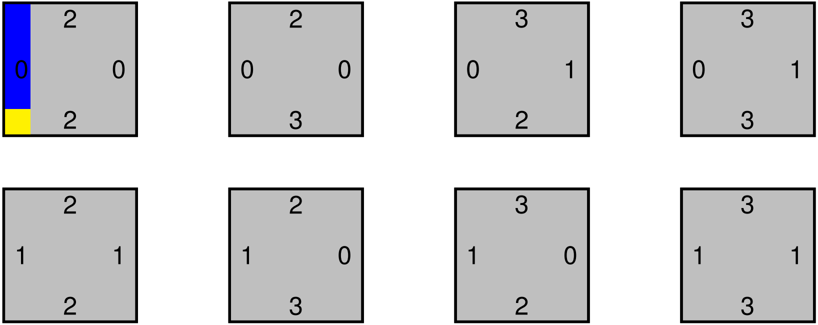
“Wangization” of semi-periodic cell

- each edge code has its master boundary particles
- mask of these particles is extended to allow them interact with all other particles with same edge number
- non-master boundary particles are deleted
- in corners, *x* direction takes precedence



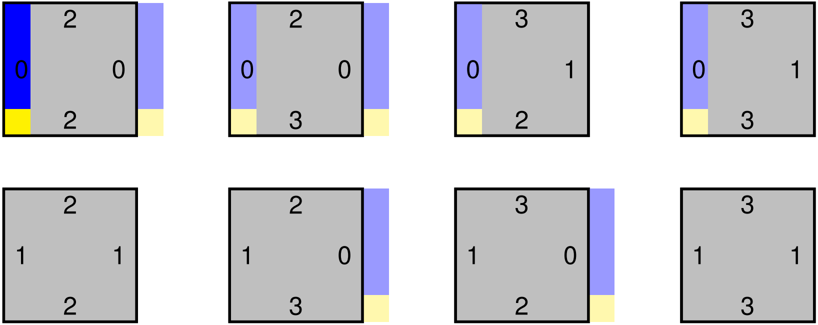
“Wangization” of semi-periodic cell

- each edge code has its master boundary particles
- mask of these particles is extended to allow them interact with all other particles with same edge number
- non-master boundary particles are deleted
- in corners, *x* direction takes precedence



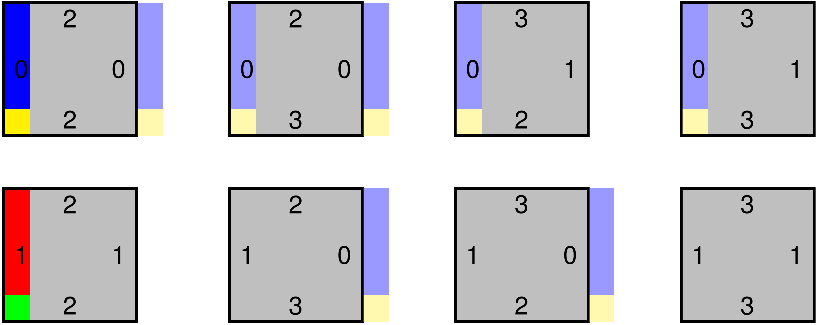
"Wangization" of semi-periodic cell

- each edge code has its master boundary particles
- mask of these particles is extended to allow them interact with all other particles with same edge number
- non-master boundary particles are deleted
- in corners, *x* direction takes precedence



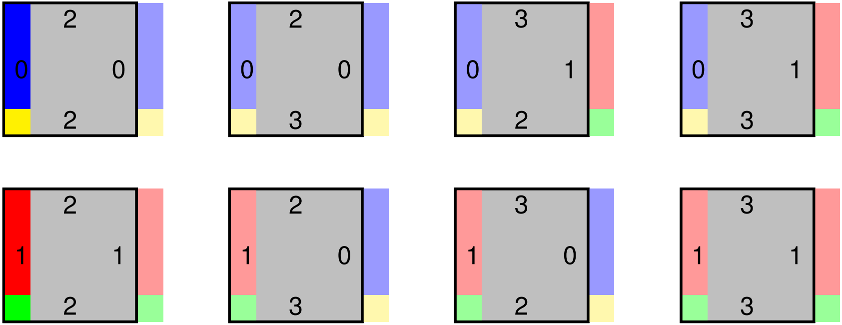
“Wangization” of semi-periodic cell

- each edge code has its master boundary particles
- mask of these particles is extended to allow them interact with all other particles with same edge number
- non-master boundary particles are deleted
- in corners, *x* direction takes precedence



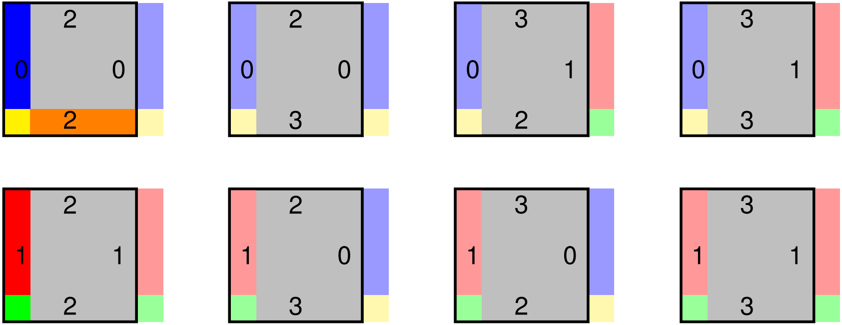
"Wangization" of semi-periodic cell

- each edge code has its master boundary particles
- mask of these particles is extended to allow them interact with all other particles with same edge number
- non-master boundary particles are deleted
- in corners, *x* direction takes precedence



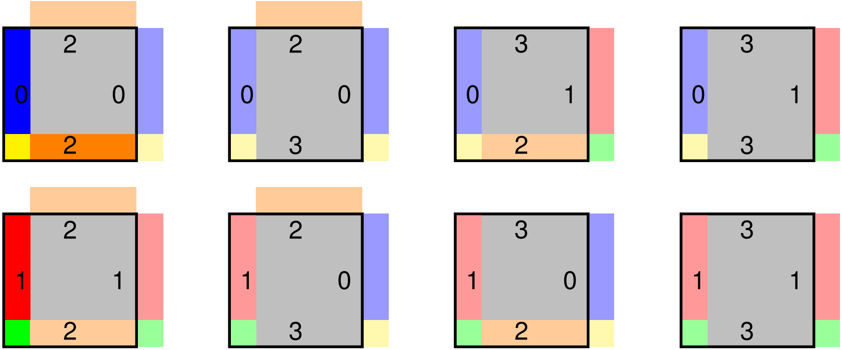
“Wangization” of semi-periodic cell

- each edge code has its master boundary particles
- mask of these particles is extended to allow them interact with all other particles with same edge number
- non-master boundary particles are deleted
- in corners, *x* direction takes precedence



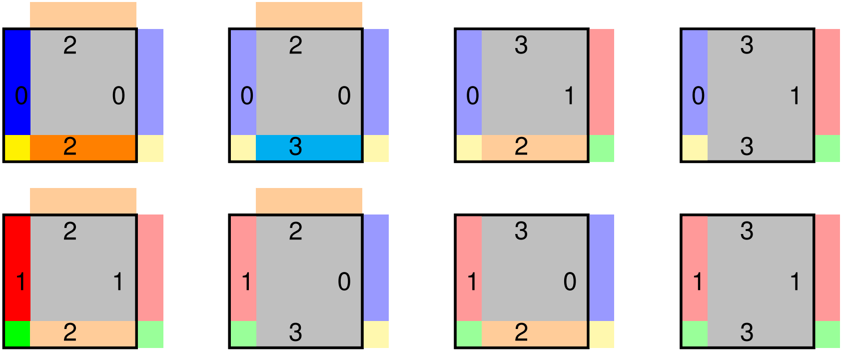
“Wangization” of semi-periodic cell

- each edge code has its master boundary particles
- mask of these particles is extended to allow them interact with all other particles with same edge number
- non-master boundary particles are deleted
- in corners, *x* direction takes precedence



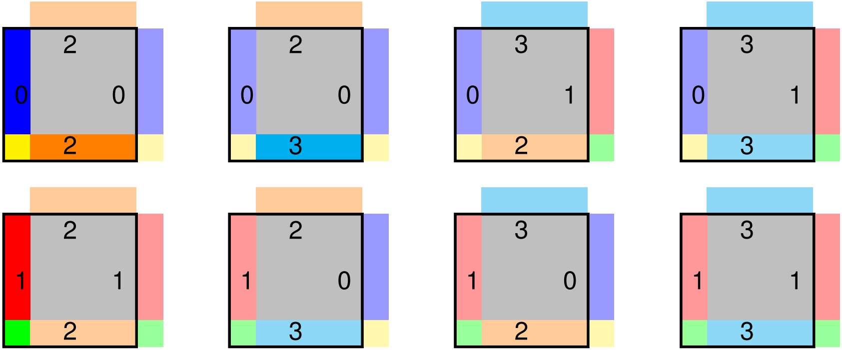
“Wangization” of semi-periodic cell

- each edge code has its master boundary particles
- mask of these particles is extended to allow them interact with all other particles with same edge number
- non-master boundary particles are deleted
- in corners, *x* direction takes precedence



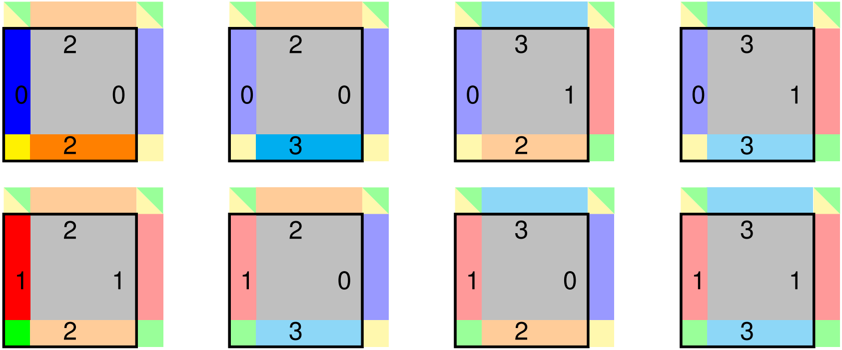
"Wangization" of semi-periodic cell

- each edge code has its master boundary particles
- mask of these particles is extended to allow them interact with all other particles with same edge number
- non-master boundary particles are deleted
- in corners, *x* direction takes precedence

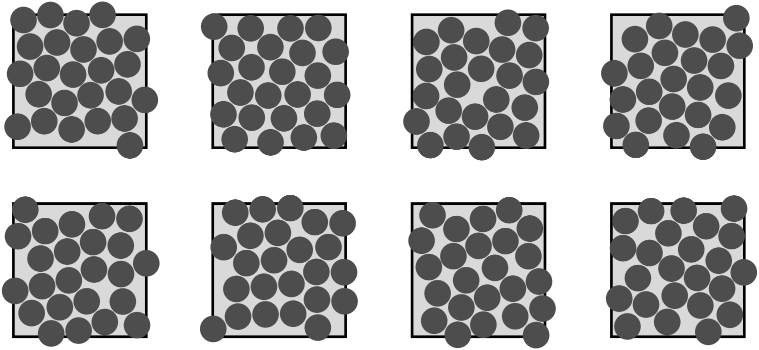


“Wangization” of semi-periodic cell

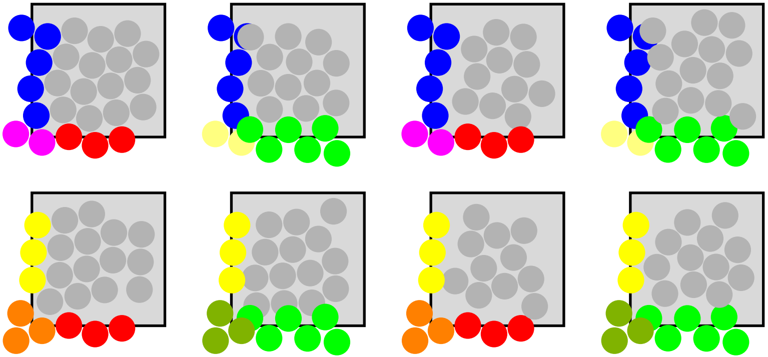
- each edge code has its master boundary particles
- mask of these particles is extended to allow them interact with all other particles with same edge number
- non-master boundary particles are deleted
- in corners, *x* direction takes precedence



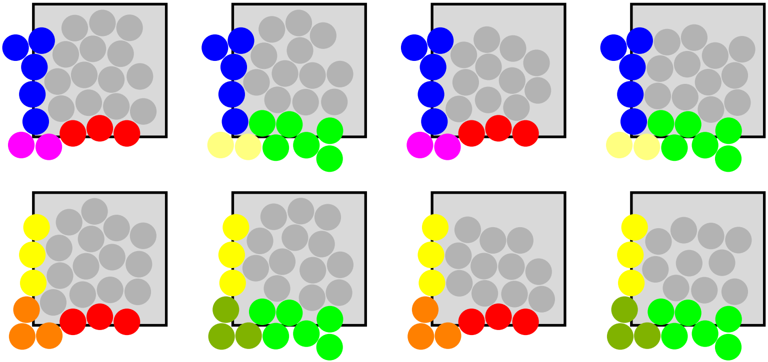
Example - 8 independent periodic cells



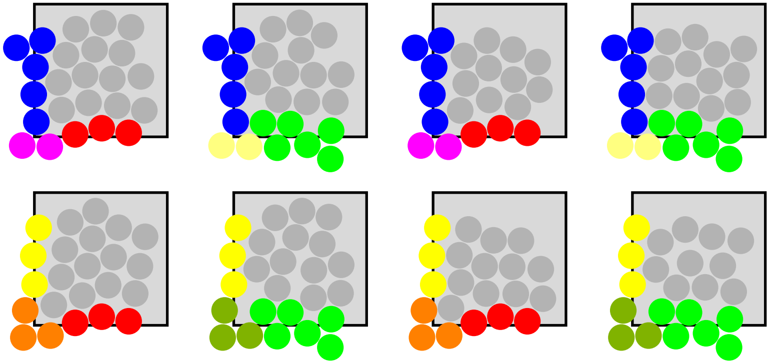
Example - “Wangization”



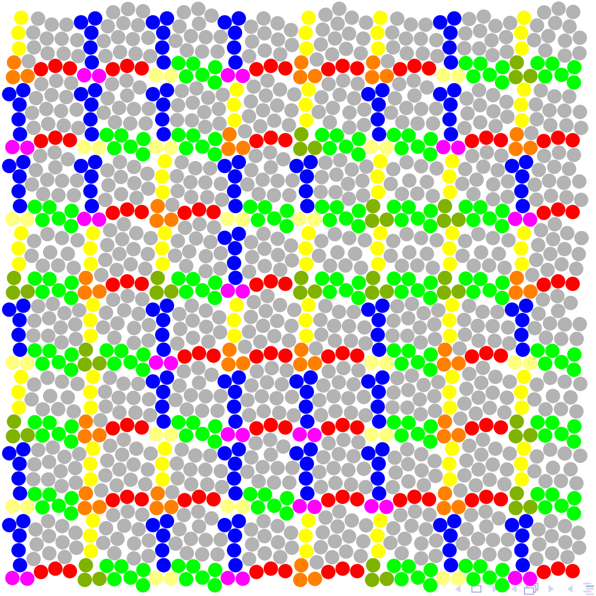
Example - relaxation



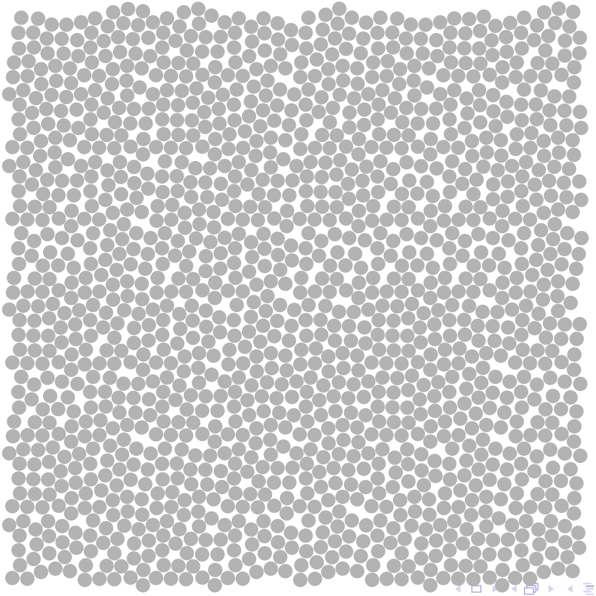
Example - a posteriori improvements



Example - tiling



Example - tiling



Conclusion

Summary

- "Wang boundary conditions" works
  - using **one** periodic cell
  - proper assignment of particles into interacting groups
- extended bitmask type (more than 64 bits) needed
- algorithm easily extensible to 3D

Future work

- extension to 3D
- polydisperse packings, PSD
- different shapes of particles
- documentation, public release of codes
- ...

Conclusion

Summary

- ”Wang boundary conditions” works
  - using **one** periodic cell
  - proper assignment of particles into interacting groups
- extended bitmask type (more than 64 bits) needed
- algorithm easily extensible to 3D

Future work

- extension to 3D
- polydisperse packings, PSD
- different shapes of particles
- documentation, public release of codes
- ...

Conclusion

Summary

- ”Wang boundary conditions” works
    - using **one** periodic cell
    - proper assignment of particles into interacting groups
- extended bitmask type (more than 64 bits) needed
- algorithm easily extensible to 3D

Future work

- extension to 3D
- polydisperse packings, PSD
- different shapes of particles
- documentation, public release of codes
- . . .

Acknowledgement

The following projects and financial endowment are gratefully acknowledged

- 13-24027S (JN, MD)
- SGS14/029/OHK1/1T/11
- YADE

Inconsistencies of contact force models and impossible elasticity of 3D granular materials

Bruno Chareyre[1]

[1]Grenoble INP, UJF, CNRS UMR 5521, 3SR lab, France

October 1st 2013

1st Yade Workshop 2014, Grenoble

Laboratoire
Sols Solides Structures Risques

Grenoble INP

Layout

1. Introduction
2. The 2D problem
    - Ratcheting in 2D
    - Corrected equations of McNamara et al.
    - Discussion
    - Numerical results
3. The 3D problem
4. Conclusions

Layout

Contact kinematics

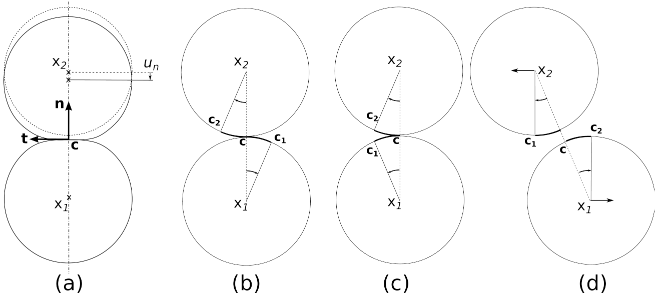


Figure: Relative movement of two particles in contact in 2D. (a) normal displacement, (b) pure shear, (c) pure rolling, (d) simple shear.

General form of contact laws

The contact law $\mathcal{L}$ defines the contact force.

$$\mathbf{f} = \mathcal{L}(\mathbf{x}_1, \mathbf{x}_2, \dot{\mathbf{x}}_1, \dot{\mathbf{x}}_2, \mathcal{H}) \qquad (1)$$

or as a rate equation:

$$\dot{\mathbf{f}} = \mathcal{L}^*(\mathbf{f}, \mathbf{x}_1, \mathbf{x}_2, \dot{\mathbf{x}}_1, \dot{\mathbf{x}}_2, \mathcal{H}) \qquad (2)$$

Normal/tangential decomposition:

$$\mathbf{f} = f_n \mathbf{n} + f_t \mathbf{t} \qquad (3)$$

$$\dot{\mathbf{f}}_t = \dot{f}_t \mathbf{t} + f_t \dot{\mathbf{t}} \qquad (4)$$

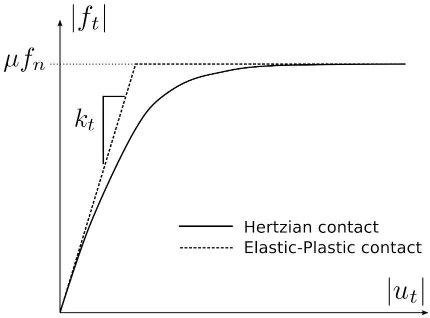
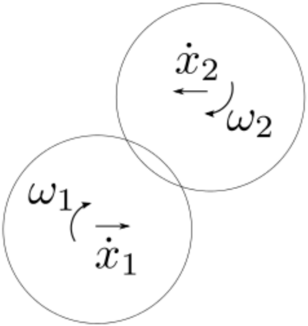


Figure: The shear force-displacement relation in Hertzian models and the elastic-plastic idealization.

Cundall's model (Géotechnique, 1979)

The procedure to define the contact forces is generally decomposed in three steps, which do not involve an explicit definition of $\mathcal{L}$. The most cited set of equation is due to P. Cundall and is as follows.
Relative velocity:

$$\mathbf{v_t} = (\dot{\mathbf{x}}_j - \dot{\mathbf{x}}_i) \cdot \mathbf{t} + r_j\omega_j + r_i\omega_i$$



Linear elasticity:

$$\dot{f}_t = -k_t(\mathbf{v_t} \cdot \mathbf{t})$$

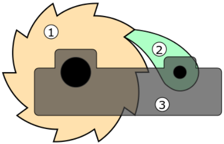
2nd law of Newton:

$$\begin{cases} m\ddot{\mathbf{x}}_\mathbf{i} \leftarrow \mathbf{f_t} + f_n\mathbf{n} \\ I\dot{\omega}_\mathbf{i} \leftarrow r_i\mathbf{f_t} \end{cases}$$

Ratcheting effect

A defect of Cundall 1979 equation has been pointed out by McNamara et al.

Quote: "The model that has been described above has been in use for almost thirty years. It has been used in many different studies, and considered to be well understood. Nevertheless, we show that this model contains an approximation that generates granular ratcheting."
(McNamara, Garcia-Rojo, Herrmann, PRE 2008)



Outline

- McNamara et al. rightly pointed out an inconsistency in Cundall 1979 and suggested a fix,
- To this date, it is not reflected in the computer codes nor in the litterature. Why?
- The suggested fix was for 2D problems, what happens in 3D?

Present study:

- Literature review and examination of publicly available DEM codes (PFC2D/3D, ESYS, LIGGGHTS, SDEC)
- Examination of objectivity, 3rd Newton's law, path dependance and energy conservation.
- Implementation of the various schemes in the same code (Yade-DEM.org) for benchmark tests

Layout

Ratcheting cycle

$$\mathbf{v_t} = (Id - \mathbf{n} \otimes \mathbf{n})(\dot{\mathbf{x}}_j - \dot{\mathbf{x}}_i) + r_j\omega_j \times \mathbf{n} + r_i\omega_i \times \mathbf{n}$$

$$\dot{f}_t = -k_t(\mathbf{v_t} \cdot \mathbf{t})$$

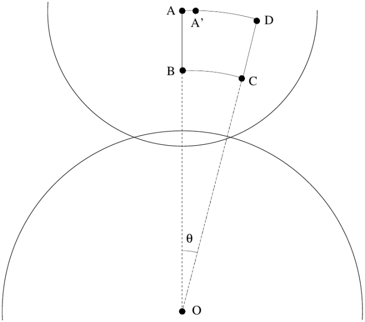


Figure: The ratcheting cycle of McNamara et al.

Corrected equations of McNamara et al.

Key points:

- First, define u_t as a function of positions/orientations
- Second, postulate an expression of the elastic potential as a function of (u_n, u_t), and deduce f_t from it:
- calculate the time derivative of f_t

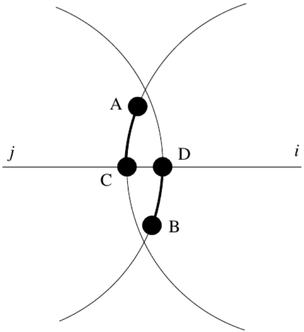


Figure: u_t after McNamara et al.

Corrected equations of McNamara et al.

The relative displacement is a function of the rotations wrt. contact normal, as in Iwashita and Oda, (Mech. Eng. 1998)

$$\mathbf{v_t} = (r_j\omega_j^* + r_i\omega_i^*) = \alpha(\dot{\mathbf{x}}_j - \dot{\mathbf{x}}_i)\cdot\mathbf{t} + (r_j\omega_j + r_i\omega_i)$$

$$\text{with } \alpha = \frac{r_j + r_j}{\|\mathbf{x}_i - \mathbf{x}_j\|} = \frac{r_j + r_j}{r_j + r_i - u_n}$$

With the elastic potential $E = \frac{1}{2}(k_n u_n^2 + k_t u_t^2)$, one has to modify the 2nd law of Newton (!) in order to conserve the Lagrangian

$$\begin{cases} m\ddot{\mathbf{x}}_\mathbf{i} \leftarrow \alpha k_t \mathbf{u_t} + f_n \mathbf{n} \\ I\dot{\omega}_\mathbf{i} \leftarrow r_i k_t \mathbf{u_t} \end{cases} \tag{5}$$

Discussion

- The additional computational cost is negligible
- The two sets of equations are identical if $u_n = 0$ ($\Rightarrow \alpha = 1$),
- The new equations seem to introduce a modification of Newton's law of motion,
- Various sets of equations found in the literature differ from Cundall 1979, those have not been commented by McNamara and co-authors

Cundall 1979:

$$\mathbf{v_t} = (\dot{\mathbf{x}}_j - \dot{\mathbf{x}}_i) \cdot \mathbf{t} + r_j\omega_j + r_i\omega_i$$

$$\dot{f}_t = -k_t(\mathbf{v_t} \cdot \mathbf{t})$$

$$\begin{cases} m\ddot{\mathbf{x}}_\mathbf{i} \leftarrow \mathbf{f_t} + f_n\mathbf{n} \\ I\dot{\omega}_\mathbf{i} \leftarrow r_i\mathbf{f_t} \end{cases}$$

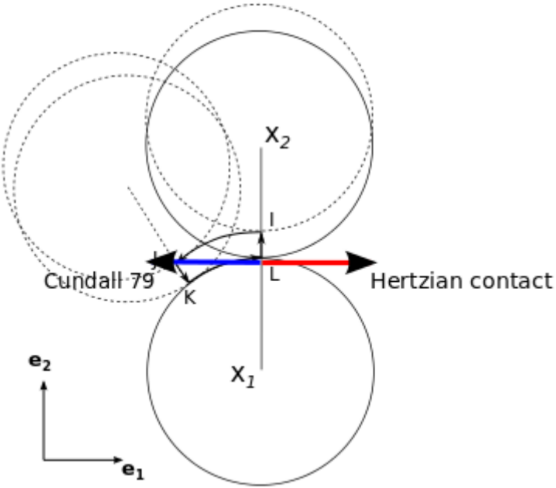
McNamara 2008:

$$\mathbf{v_t} = \alpha(\dot{\mathbf{x}}_j - \dot{\mathbf{x}}_i) \cdot \mathbf{t} + (r_j\omega_j + r_i\omega_i)$$

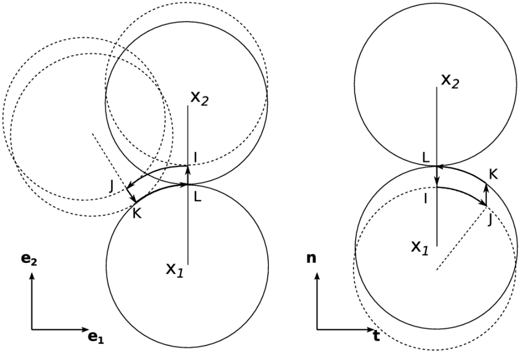
$$E = \frac{1}{2}(k_n u_n^2 + k_t u_t^2)$$

$$\begin{cases} m\ddot{\mathbf{x}}_\mathbf{i} \leftarrow \alpha k_t\mathbf{u_t} + f_n\mathbf{n} \\ I\dot{\omega}_\mathbf{i} \leftarrow r_i k_t\mathbf{u_t} \end{cases}$$

Why is path dependence bad?

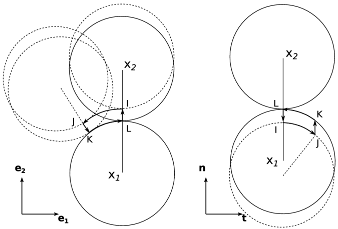


Objectivity



$$\mathbf{v_t} = (\dot{\mathbf{x}}_j - \dot{\mathbf{x}}_i) \cdot \mathbf{t} + r_j\omega_j + r_i\omega_i$$

Objectivity



Introducing the contact spin ω_n we can rewrite v_t in Cundall 1979 as

$$\mathbf{v_t} = \|x_i - x_j\|\omega_n - r_j\omega_j - r_i\omega_i$$

After rearranging, the expression as a function of rotations in the local frame is obtained:

$$\mathbf{v_t} = u_n\ \omega_n - r_j\omega_j^* - r_i\omega_i^*$$

In a rigid body rotation:

$$\dot{f}_t = k_t u_n \omega_n = \frac{k_t}{k_n} f_n \omega_n$$

Yet another set of equations

Introducing the contact point leads to another set of equations. This one was first introduced by Cundall 1974 (US Army Tech. Rep. MRD-2-74[1]). It is widely used, especially in 3D (PFC2D/3D, ESYS, SDEC, LIGGGHTS, YADE, ...).

$$\mathbf{v_t} = (\dot{\mathbf{x}}_j - \dot{\mathbf{x}}_i)\cdot\mathbf{t} + \omega_j \times (\mathbf{c} - \mathbf{x_j}) - \omega_i \times (\mathbf{c} - \mathbf{x_i})$$

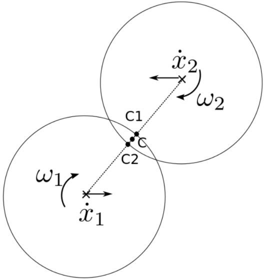
or $\mathbf{v_t} = (\dot{\mathbf{x}}_j - \dot{\mathbf{x}}_i)\cdot\mathbf{t} + r_j^*\omega_j + r_i^*\omega_i$

Like before:

$$f_t = -k_t u_t \text{ (or equivalently } \dot{f}_t = -k_t v_t)$$

The moments are calculated using the contact point

$$\begin{cases} m\ddot{\mathbf{x}}_\mathbf{i} \leftarrow \mathbf{f_t} + f_n\mathbf{n} \\ I\dot{\omega}_\mathbf{i} \leftarrow r_i^*\mathbf{n}\times\mathbf{f_t} \end{cases}$$



$C = pC_i + qC_j$

classical choice:

$p = q = 0.5$

[1] see also Hart, Cundall, Lemos, Rock Mech. Mining Sci., 1988

C. Action reaction (3rd Newton's law)

$$\begin{cases} f_{1\to2} = -f_{2\to1} \\ M_{1\to2/O} = -M_{2\to1/O} \end{cases} \tag{23}$$

$$M_{1\to2/O_2} - \mathbf{x_1} \times \mathbf{f}_{1\to2} + M_{2\to1/O_1} - \mathbf{x_2} \times \mathbf{f}_{2\to1} = \mathbf{0} \tag{24}$$

$$M_{1\to2/O_2} + M_{2\to1/O_1} = (r_1^* + r_2^*)\mathbf{n} \times \mathbf{f}_{1\to2} \tag{25}$$

We can obtain more general constraints on the contact force expression by considering the virtual power of f_n, M_1 and M_2 in an arbitrary deformation rate $\dot{u}_n, \dot{\Omega}_1, \dot{\Omega}_2$

$$W = f_n \dot{u}_n + M_1 \cdot \dot{\Omega}_1 + M_2 \cdot \dot{\Omega}_2, \tag{39}$$

Which has a potential only if the curl of the $\mathbb{R}^3$ vector f_n, M_1, M_2 is zero:

$$\frac{\partial f_n}{\partial \Omega_1} - \frac{\partial (r_1^* f_t)}{\partial u_n} = 0 \tag{40}$$

$$\frac{\partial f_n}{\partial \Omega_2} - \frac{\partial (r_2^* f_t)}{\partial u_n} = 0 \tag{41}$$

$$\frac{\partial (r_1^* f_t)}{\partial \Omega_2} - \frac{\partial (r_2^* f_t)}{\partial \Omega_1} = 0 \tag{42}$$

Constraint 1: f_n independant of rotations
The system then gives

$$p f_t - r_1^* \frac{\partial(f_t)}{\partial u_n} = 0$$
$$q f_t - r_2^* \frac{\partial(f_t)}{\partial u_n} = 0$$
$$\frac{\partial(r_1^* f_t)}{\partial \Omega_2} - \frac{\partial(r_2^* f_t)}{\partial \Omega_1} = 0.$$

Combining the two first equations leads to

$$p(r_2 - q u_n) - q(r_1 - p u_n) = 0$$

Since $p + q = 1$, constraints 1 leads to $p = r_1/(r_1 + r_2)$, $q = r_2/(r_1 + r_2)$ and we can rewrite the first equation

$$r_1 \frac{\partial(f_t/\alpha)}{\partial u_n} = 0 \quad (47)$$

so that

$$f_t = \alpha f_t^0(\Omega_1, \Omega_2) \quad (48)$$

Constraint 2: f_t independant of u_n

In that case eqs. 40-42 give

$$\frac{\partial(f_n)}{\partial\Omega_1} + pf_t = 0 \quad (49)$$

$$\frac{\partial(f_n)}{\partial\Omega_2} + qf_t = 0 \quad (50)$$

$$r_1^*\frac{\partial f_t}{\partial\Omega_2} - r_2^*\frac{\partial f_t}{\partial\Omega_1} = 0. \quad (51)$$

If f_t is independant of u_n, the last equation can be true for all u_n only if r_1^*/r_2^* is a constant, which again leads us to $p = r_1/(r_1 + r_2)$, $q = r_2/(r_1 + r_2)$. At this condition, any function of $r_1\Omega_1 + r_2\Omega_2$ is a solution of last equation. Choosing a linear function of rotations leads to

$$f_t = k_t(\Omega_1 r_1 + \Omega_2 r_2) = k_t u_t \quad (52)$$

Which is independant of α. In turns, the other equations can be integrated and lead to

$$f_n = f_n(u_n, u_t) = f_n(u_n, 0) + \frac{k_t u_t^2}{2(r_1 + r_2)} \quad (53)$$

If f_n is defined as linear in u_n and if it is must be zero when the contact is created (i.e. when u_n and u_t are both zero), then

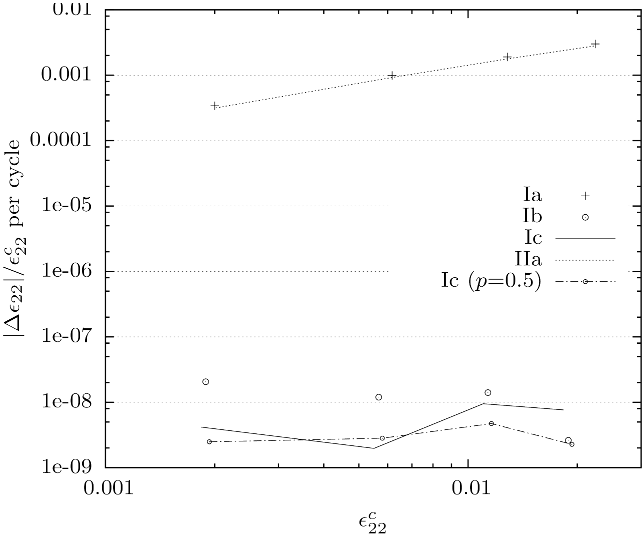
$$f_n = k_n u_n + \frac{k_t u_t^2}{2(r_1 + r_2)} \quad (54)$$

and the potential energy of the contact is

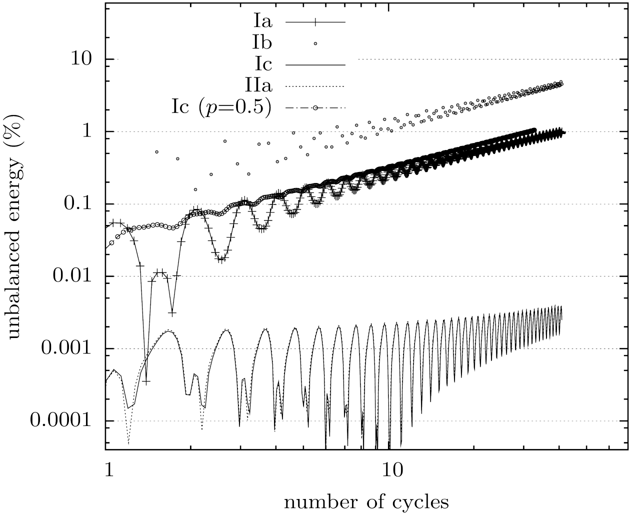
$$V = \frac{1}{2}(k_n u_n^2 + \frac{1}{\alpha}k_t u_t^2) \quad (55)$$

May be interpreted as an inclination of the shear force when the contact is squeezed

Numerical results



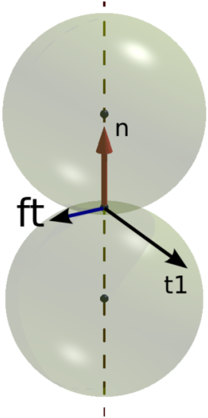
Numerical results



Layout

Contact geometry in 3D

Recall: $\dot{\mathbf{f}}_\mathbf{t} = \dot{f}_t \mathbf{t} + f_t \dot{\mathbf{t}}$
Stiff modes: traction/compression + shear
Floppy modes: rolling + twisting



Contact geometry in 3D

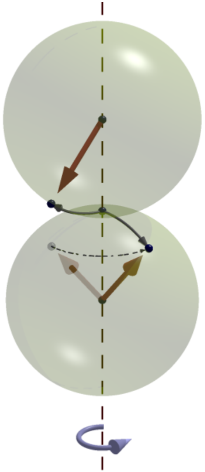
Recall: $\dot{\mathbf{f}}_\mathbf{t} = \dot{f}_t \mathbf{t} + f_t \dot{\mathbf{t}}$
Stiff modes: traction/compression + shear
Floppy modes: rolling + twisting
The path independent relations (in 2D) can be generalized to 3D easily. In the "contact point" method, the integrand would be divided by α. We know already that it leads to path dependence.

$$\mathbf{u}(t) = \int_{t_0}^{t} (r_1\omega_1^* + r_2\omega_2^*) \times \mathbf{n} dt \tag{6}$$

Unfortunately, it can be shown that $\mathbf{u}_k(t)$ is not a path independent function of the rotation, using the identity
$\dot{R} = (\omega^* \times)R$

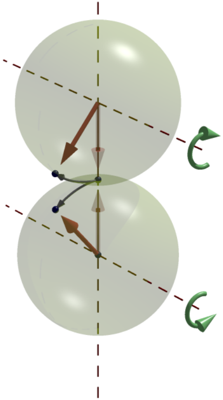
Path dependance in 3D

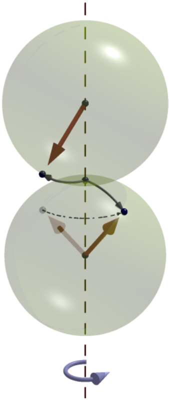
Wang (Acta Geotech. 2009) proposed an approach similar to McNamara et al., i.e. define the force as function of positions and orientations. It has been implemented in the ESYS code.



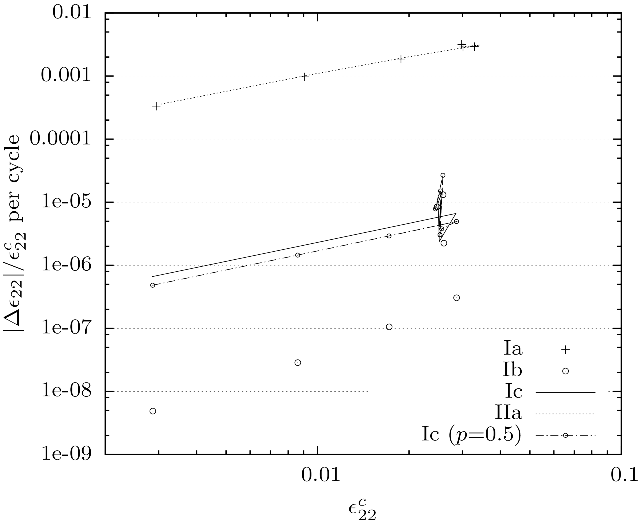
Path dependance in 3D

An interesting point is that it gives a different rotation of the shear force in twisting motion.
However, it is energetically inconsistent...

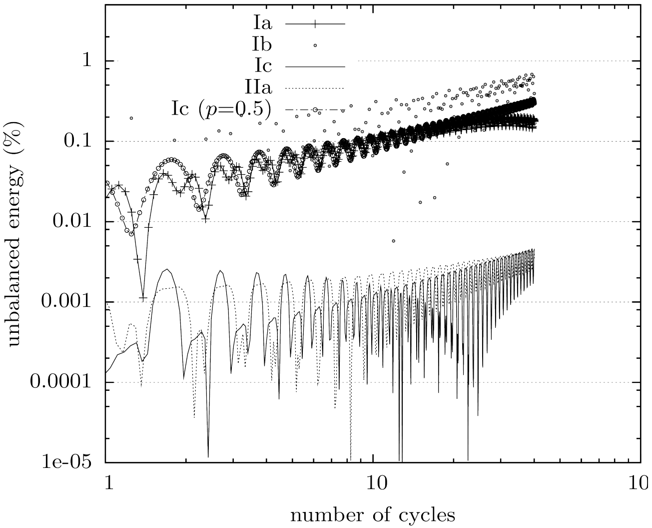




Numerical results



Numerical results



Layout

Conclusions

- Programming a DEM code is easy. Programming an inconsistent DEM code is even easier.
- We cannot have cross independency of the normal and tangential force/displacement relations
- In 2D we proposed the general form for consistent elastic relations, McNamara et al. was found as a special case
- In 3D, it is simply impossible to define a path independant shear force
- A 3D packing has always internal mechanisms
- The bigger the packing, the less these mechanisms are noticeable at the macroscale

YADE 2014

“Short overview of the development process”

CONTENT

GIT-Statistic

General statistic (state 2014/06/18)

1. First commit: Thu **Jan 6 13:52:30 2005** from Olivier Galizzi
2. Age: 3451 days, 1483 active days (42.97%)
3. Total Files: 824
4. Total Lines of Code: **132 884** (2 019 236 +, 1 886 352 -)
5. Total Commits: **4385**
6. Average 3.0 commits per active day, 1.3 per all days
7. Authors: 35 (average 125.3 commits per author)
8. Average 125.3 commits per author
9. Total releases: 33
10. Average commits per release: 132.88

ACTIVITY (GIT)

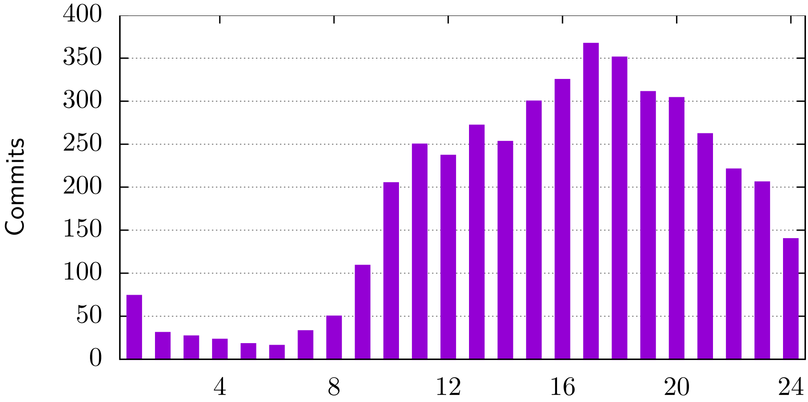


Fig. 1: Hour of day

ACTIVITY (GIT)

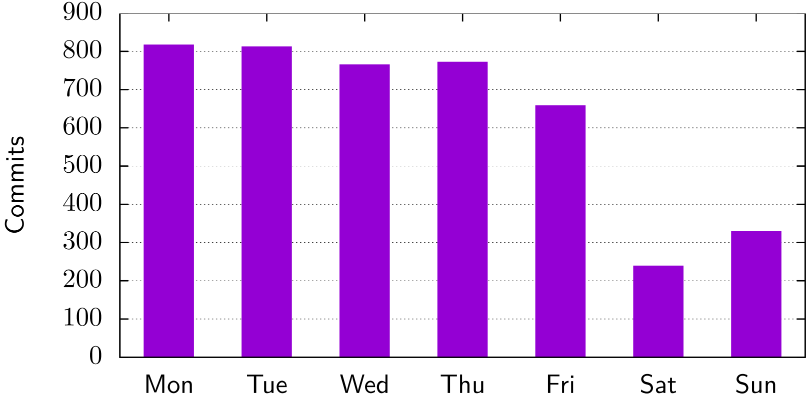


Fig. 2: Day of week

ACTIVITY (GIT)

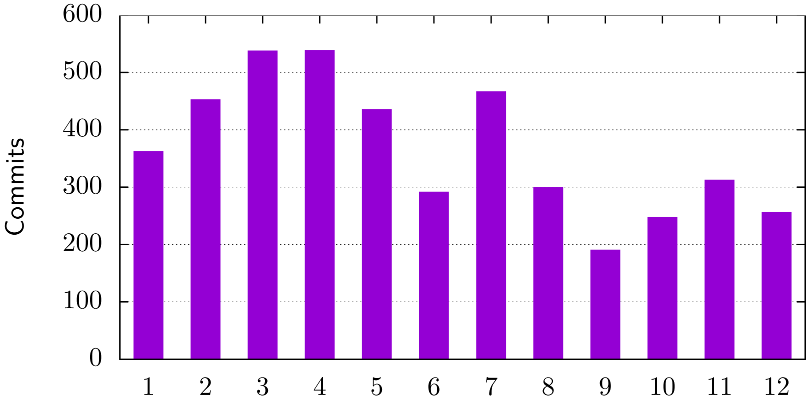


Fig. 3: Month of year

ACTIVITY (GIT)

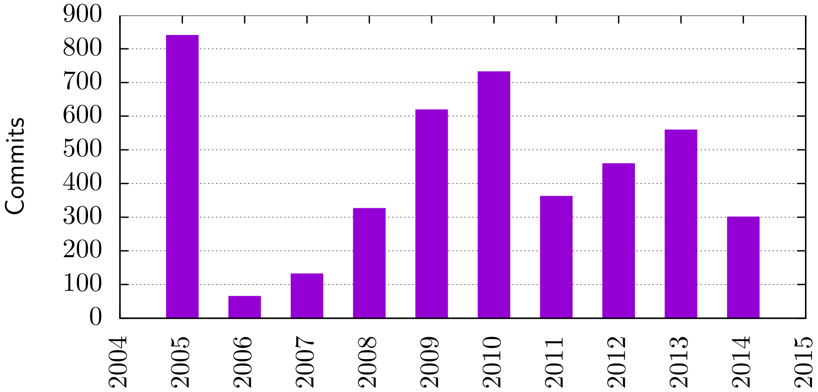


Fig. 4: Commits by year

ACTIVITY (GIT)

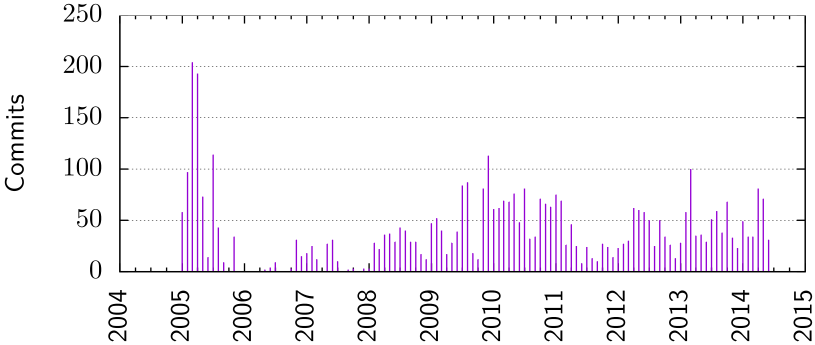


Fig. 5: Commits by year/month

ACTIVITY (GIT)

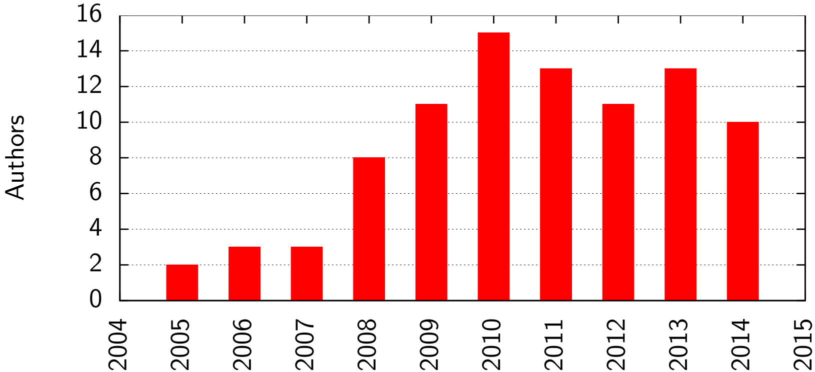


Fig. 6: Author by year

ACTIVITY (GIT)

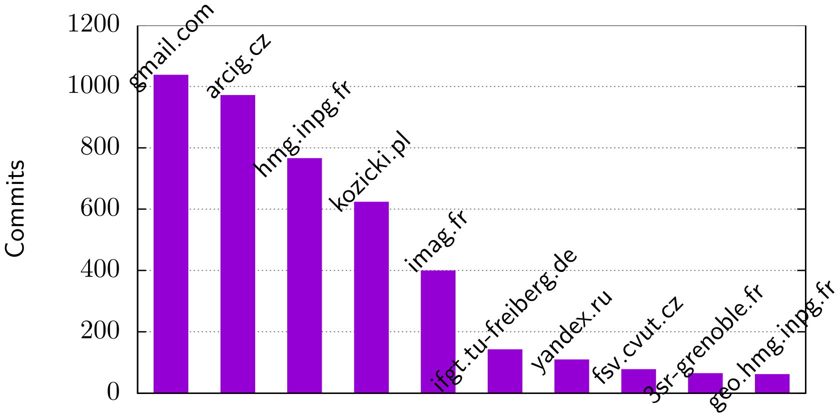


Fig. 7: Commits by domain

ACTIVITY (GIT)

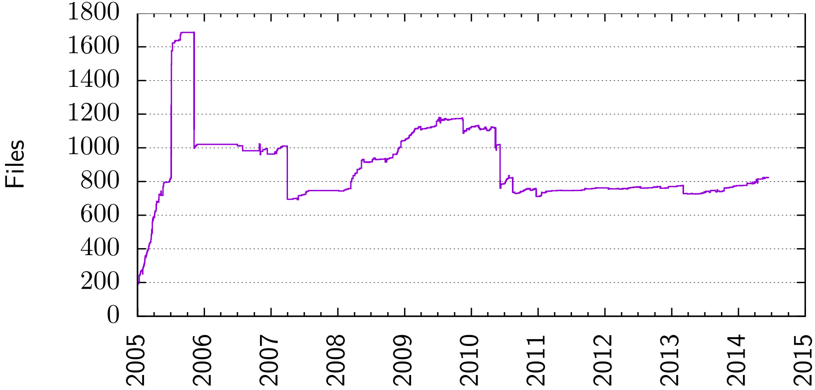


Fig. 8: Files by date

ACTIVITY (GIT)

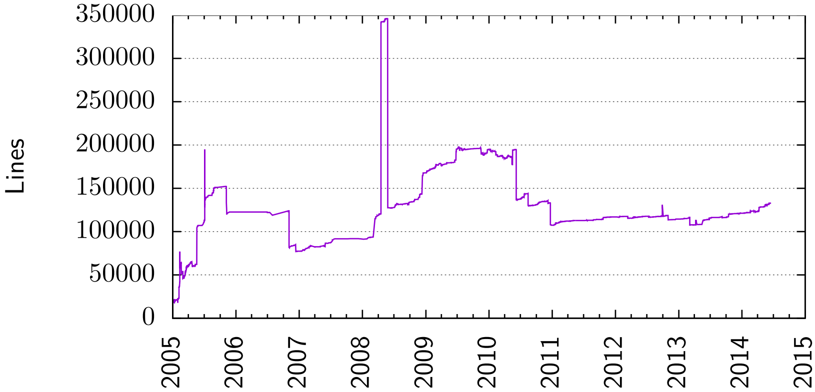


Fig. 9: Lines of code by date

YADE-DEM.ORG (2012/01/01 – 2014/06/18)

General statistic

1. All sessions: 83 400
2. Users: 38 201
3. Pageviews: 308 722
4. Page/sessions: 3.7
5. Average session duration: 00:05:10

YADE-DEM.ORG (2012/01/01 – 2014/06/18)

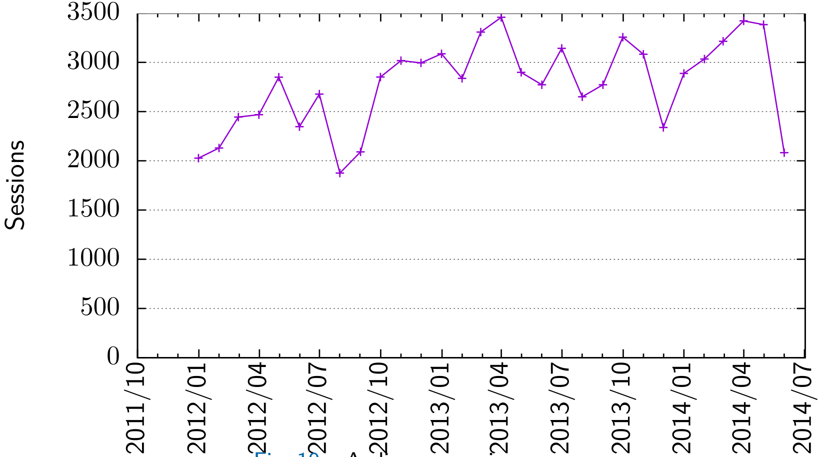


Fig. 10: Audience overview

YADE-DEM.ORG (2012/01/01 – 2014/06/18)

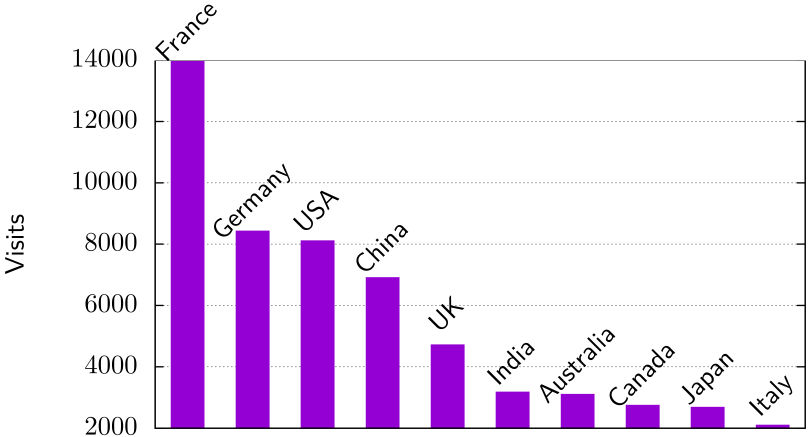


Fig. 11: Location of visitors

YADE-DEM.ORG (2012/01/01 – 2014/06/18)

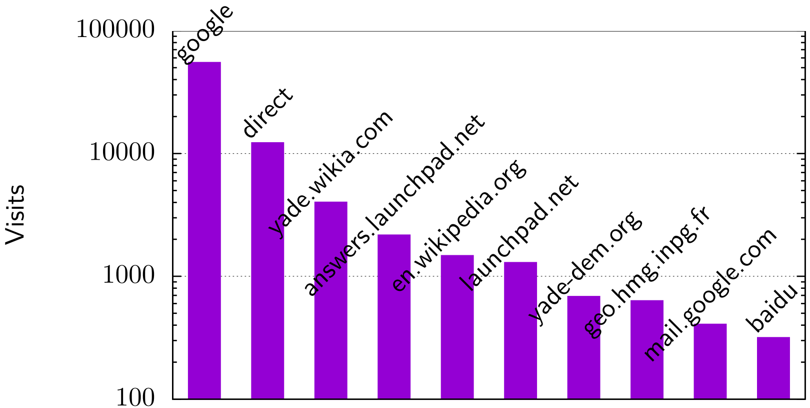


Fig. 12: Source of traffic

YADE-DEM.ORG (2012/01/01 – 2014/06/18)

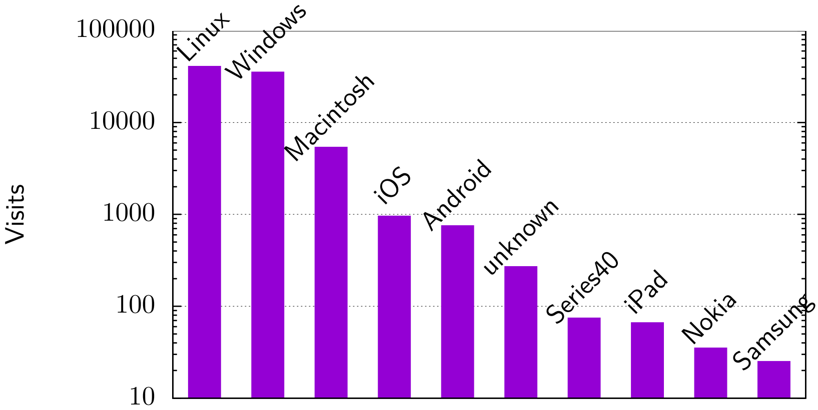


Fig. 13: Operating system

DEVELOPMENT POINTS SINCE 2011

Done

1. Migrate build-system to CMake
2. Move version control system to Git => GitHub
3. Adding Yade to Debian main archive
4. Move daily-packages from Launchpad to yade-dem.org

Plans

1. Provide Qt5-support
2. Provide Python3-support
3. Slowly migrate to C++11 and C++14
4. Provide binaries for Windows-users (?)

QUALITY ASSURANCE

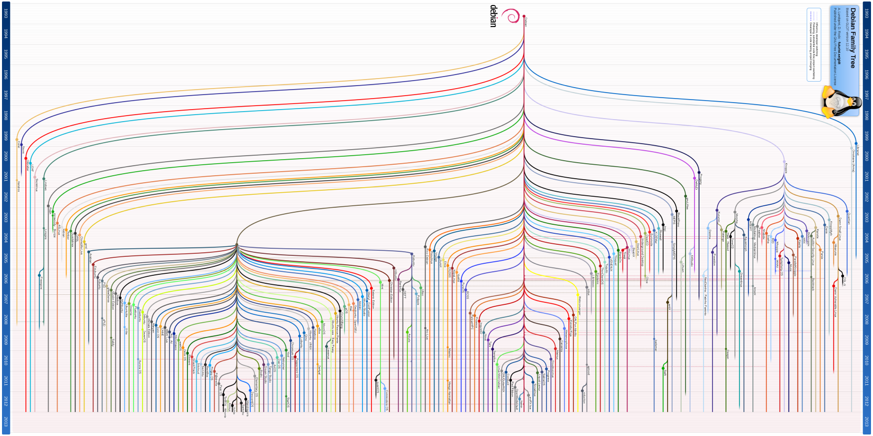
Buildbot

1. Compiles the whole Yade-code after each commit
2. Nightly compilation with gcc and clang
3. Execute –test and –check autotests during package building

Current development

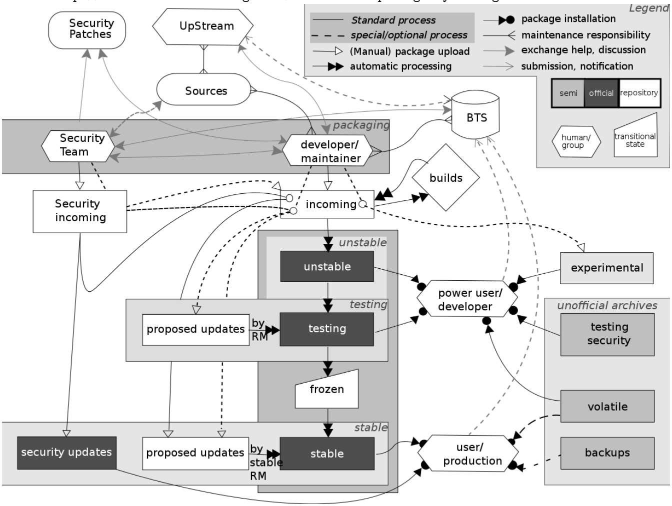
WHY DEBIAN?

Source: `http://commons.wikimedia.org/wiki/File:Debian_family_tree_11-06.png`



DEBIAN PACKAGE CYCLE

Source: `http://commons.wikimedia.org/wiki/File:Debian-package-cycle.svg`



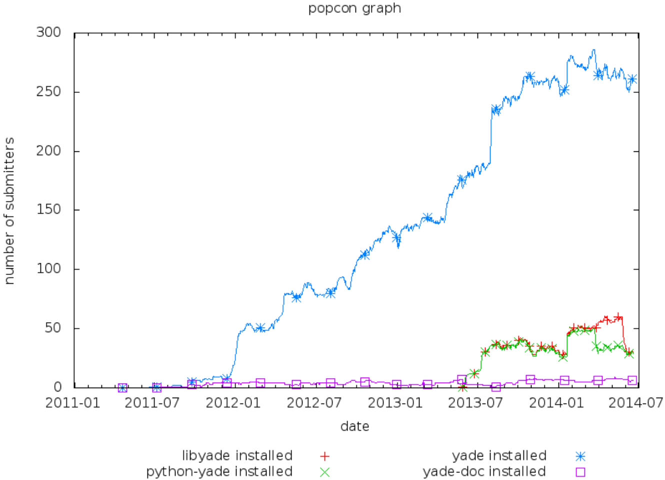
YADE IN DEBIAN AND UBUNTU

Steps

1. First upload into Debian 01/2011 (version 0.60-1)
2. Released in Debian Wheezy 05/2013 (version 0.80.1-2)
3. First sync in Ubuntu 05/2011
4. Released in Ubuntu Oneiric 07/2011 (version 0.60.3-2)
5. Total 38 uploads into Debian
6. "Daily" packages are provided for recent Debian/Ubuntu versions

YADE IN DEBIAN AND UBUNTU

Source: `http://qa.debian.org/popcon.php?package=yade`



YADE IN DEBIAN AND UBUNTU

Packages

1. Source package builds 4 binary-packages: yade, libyade, python-yade, yade-doc
2. Builds in a clean environment (pbuilder)
3. Test functionality during package building

YADE DEVELOPMENT WORKFLOW

GIT

1. Small, logical commits
2. Don't commit half-done work
3. Always review code before committing it
4. Don't change published history
5. Do keep up to date
6. Don't commit binary files
7. Write good commit messages

Source [1] http://www.lullabot.com/blog/article/git-best-practices-workflow-guidelines
Source [2] http://www.git-tower.com/learn/ebook/command-line/appendix/best-practices

YADE DEVELOPMENT WORKFLOW

Good commit message, example

Short (50 chars or less) summary of changes

More detailed explanatory text, if necessary. Wrap it to about 72 characters or so. In some contexts, the first line is treated as the subject of an email and the rest of the text as the body. The blank line separating the summary from the body is critical (unless you omit the body entirely); tools like rebase can get confused if you run the two together.

Further paragraphs come after blank lines.

Source [1] http://git-scm.com/book/en/Distributed-Git-Contributing-to-a-Project

YADE DEVELOPMENT WORKFLOW

General tips

1. Our license GPL-2+, do not choose another one!
2. Avoid adding 3-rd party files into the source
3. Do not use "using namespace ..." in a global scope
4. Do not change formatting in files (indenting, braces position etc.)
5. Try avoid code duplication
6. Not sure – ask!

Thank you for your attention!

Yade : continuous integration, doc and wiki

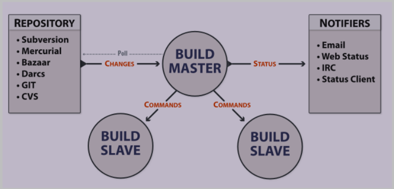
Rémi Cailletaud

Laboratoire 3SR

Yade Workshop 7-8-9 July 14

Automatic build, test and release

- Polls sources from repo, and compile/test
- Cry if you break something (yade-dev mailing list)
- Test packages building



https://yade-dem.org/buildbot/

Hardware

- Build master : a VM hosting doc, wiki, and buildbot (yade-dem.org).
- Build slaves : 3 nodes from 3SR Lab cluster, submitting high priority compile job.

Builders

		clang	full	package
When	Nightly	✓	✓	✓
	On each commit		✓	
What	Checkout, compile, install	✓	✓	✓
	Test, check [1]	✓	✓	✓
	Compile and upload doc		✓	✓
	Build packages			✓
	Cry if something goes bad	✓	✓	✓

[1] yade –test and yade –checks

Web view

Home - Waterfall Grid T-Grid Console Builders Recent Builds Buildslaves Changesources Users - JSON API - About

Builder yade-full Build #2588

Results:

Build successful

SourceStamp:

Project	yade
Repository	https://github.com/yade/trunk
Branch	master
Revision	df135d74a9fc67aeb1758f7a3a107767bea30ed5
Got Revision	df135d74a9fc67aeb1758f7a3a107767bea30ed5
Changes	see below

BuildSlave:

r0calcul5

Reason:

scheduler

Steps and Logfiles:

1. git update (17 secs)
   1. stdio
2. shell Cmake OK (1 mins, 7 secs)
   1. stdio
3. compile compile warnings (42 mins, 14 secs)
   1. stdio
   2. warnings (8)
4. shell_1 Install OK (4 secs)
   1. stdio
5. test testing build (4 secs)
   1. stdio
6. test_1 checking build (10 secs)
   1. stdio
7. shell_2 doc compiled (7 mins, 48 secs)
   1. stdio
8. MasterShellCommand Ran (0 secs)
   1. stdio
9. upload uploading (5 secs)
   1. - no logs -
10. upload_1 uploading (5 secs)
    1. - no logs -
11. upload_2 uploading Yade.pdf (1 secs)
    1. - no logs -
12. upload_3 uploading Yade.epub (2 secs)
    1. - no logs -
13. MasterShellCommand_1 Ran (0 secs)
    1. stdio
14. shell_3 cleaned (1 secs)
    1. stdio

Build Properties:

Name	Value	Source
branch	master	Build
buildername	yade-full	Builder
buildnumber	2588	Build
got_revision	df135d74a9fc67aeb1758f7a3a107767bea30ed5	Source
project	yade	Build
repository	https://github.com/yade/trunk	Build
revision	df135d74a9fc67aeb1758f7a3a107767bea30ed5	Build
scheduler	full	Scheduler
slavename	r0calcul5	BuildSlave
warnings-count	8	WarningCountingShellCommand
workdir	/home/buildbot/yade/yade-full	slave

Forced Build Properties:

Name	Label	Value

Blamelist:

1. Anton Gladky

Timing:

Start	Thu Jul 3 21:19:32 2014
End	Thu Jul 3 22:11:35 2014
Elapsed	52 mins, 3 secs

Resubmit Build:

This tree was built from a specific set of source files, and can be rebuilt exactly

To force a build, fill out the following fields and push the 'Force Build' button

Reason for re-running build:

Rebuild

All Changes:

1. **Change #1750**

Category	None
Changed by	Anton Gladky <gladky.anton@gmail.com>
Changed at	Thu 03 Jul 2014 21:08:46
Repository	https://github.com/yade/trunk
Project	yade
Branch	master
Revision	df135d74a9fc67aeb1758f7a3a107767bea30ed5

Comments

Add missing header.

Documentation and wiki

https://yade-dem.org/doc/

- Yade web site home page
- Updated each commit

https://yade-dem.org/wiki/

- Yade quick tour of installation and usage
- Account required to edit

Membrane finite element in Woo

Václav Šmilauer

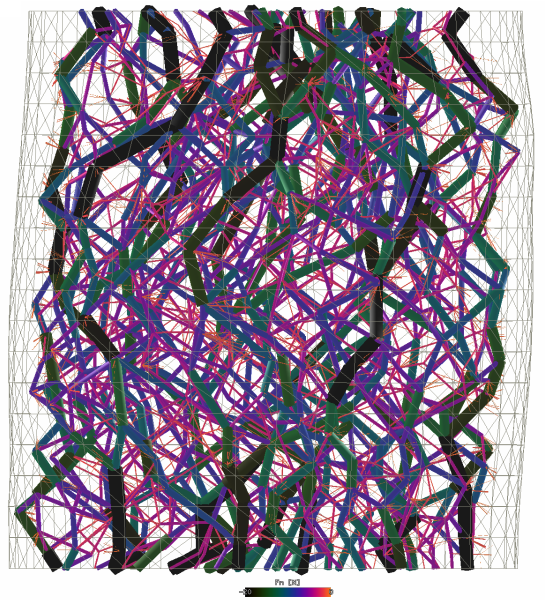
woodem.eu ib-keramik.de

July 2014

Abstract

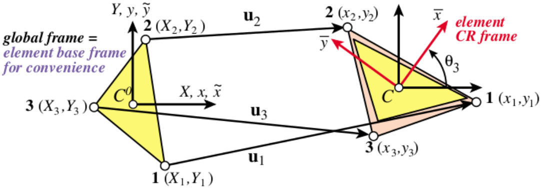
Woo implements 3-node triangle element in 3d – superposition of the CST and DKT elements known from FEM. This talks gives an overview of the elemtn theory and their integration into DEM.

Motivation



- please do a simulation of triaxial test with rubber membrane
- 3-node facets in Woo, good for rigid mesh, what is missing?
- nodes respond to motion (lumped mass)
- facets generate nodal forces from its own deformation
- nodal forces: superpose plate (CST) and bending (DKT) elements

Corotational formulation of FEM



- decompose motion of an element into rigid body + strains of the corotated configuration [1]
- rigid body motion is unconstrained
- strains must be *small*

Best-fit corotational frame (element-local coordinates)

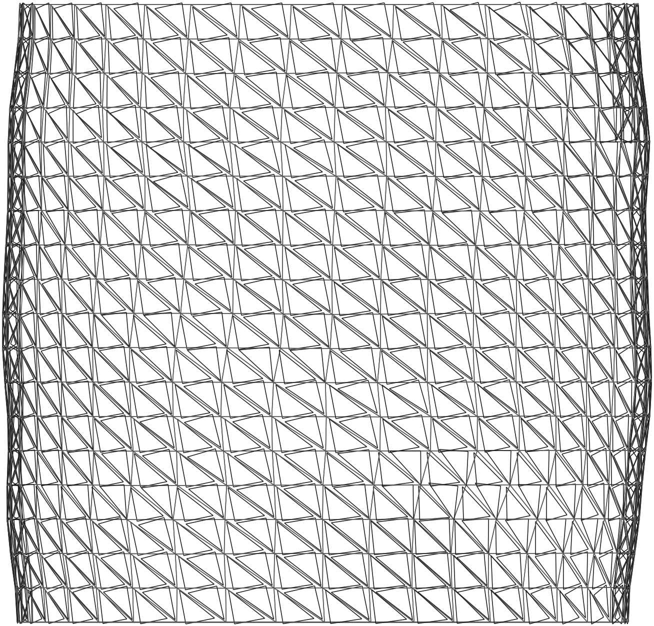
Problem

- Find local frame minimizing strain (nodal displacements)
- X_i, Y_i are reference nodal positions in element base frame
- x_i, y_i are current nodal positions in not-yet-z-rotated frame
- nodal displacements: the difference between X_i, Y_i and coordinates in CR

Position and orientation

- frame origin: always the centroid of the element
- frame xy plane: normal $((x_2, y_2) - (x_1, y_1)) \times ((x_3, y_3) - (x_2, y_2))$
- rotation around z:[1]

$$\tan \vartheta_3 = \frac{x_1 Y_1 + x_2 Y_2 + x_3 Y_3 - y_1 X1 - y_2 X_2 - y_3 X_3}{x_1 X_1 + x_2 X_2 + y_3 Y_3 + y_1 Y_1 + y_2 Y_2 + y_3 Y_3}$$



Rotations

- element rotation WRT initial element-node ($\mathbf{q}_e$) orientation
- store initial per-node rotation difference $\Delta\Phi_{i0} = \mathbf{q}^*_{i0}\mathbf{q}_{e0}$ (quaternions: read backwards, multiplication by conjugate is rotation subtraction; purists write $\mathbf{q}^*_i\mathbf{q}_e\mathbf{q}_i$)
- current nodal rotation is $(\Delta\Phi_i)^*(\mathbf{q}^*_i\mathbf{q}_e) = (\mathbf{q}^*_{i0}\mathbf{q}_{e0})^*(\mathbf{q}^*_i\mathbf{q}_e)$
- breaks for rotations $> \pi$, but that is hardly *small* strain

Plate element (constant strain triangle, CST)

- strain-displacement matrix

$$\mathbf{B} = \frac{1}{2A} \begin{pmatrix} y_{23} & 0 & y_{31} & 0 y_{12} & 0 & \\ 0 & x_{23} & 0 & x_{12} & 0 & x_{21} \\ x_{32} & y_{23} & x_{12} & y_{31} & x_{21} & y_{12} \end{pmatrix}.$$

- elastic stiffness matrix (plane stress):

$$\mathbf{E} = \frac{E}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{pmatrix}$$

- stiffness matrix (for constant element thickness h):

$$\mathbf{K}^{\mathrm{CST}} = Ah\mathbf{B}^T\mathbf{E}\mathbf{B}.$$

CST forces

$$\begin{pmatrix} F_{x1} \\ F_{y1} \\ F_{x2} \\ F_{y2} \\ F_{x3} \\ F_{y3} \end{pmatrix} = \mathbf{K}^{\mathsf{CST}} \begin{pmatrix} u_{x1} \\ u_{y1} \\ y_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{pmatrix}$$

$$\mathbf{F}^{\mathsf{CST}} = \mathbf{K}^{\mathsf{CST}} \mathbf{u}$$

Bending element (discrete Krichhoff triangle, DKT)

- bending elasticity matrix[3]

$$\mathbf{D}_b = \frac{Eh^3}{12(1-\nu^2)} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{pmatrix}$$

- strain-displacement matrix (see [3] for details)

$$\mathbf{B}_b(\xi,\eta) = \frac{1}{2A} \begin{pmatrix} y_{31}\mathbf{H}^T_{x,\xi} + y_{12}\mathbf{H}^T_{x,\eta} \\ -x_{31}\mathbf{H}^T_{y,\xi} - x_{12}\mathbf{H}^T_{y,\eta} \\ -x_{31}\mathbf{H}^T_{x,\xi} - x_{12}\mathbf{H}^T_{x,\eta} + y_{31}\mathbf{H}^T_{y,\xi} + y_{12}\mathbf{H}^T_{y,\eta} \end{pmatrix}.$$

- stiffness matrix (integrated numerically using Gauss quadrature[4])

$$\mathbf{K}^{DKT} = 2A \int_0^1 \int_0^{1-\eta} \mathbf{B}^T \mathbf{D}_b \mathbf{B} \, d\xi \, d\eta$$

DKT generalized forces

$$\mathbf{F}^{\mathsf{DKT}} = \begin{pmatrix} F_{z1} \\ T_{x1} \\ T_{y1} \\ F_{z2} \\ T_{x1} \\ T_{x2} \\ F_{z3} \\ T_{x3} \\ T_{y3} \end{pmatrix} = \mathbf{K}^{\mathsf{DKT}} \begin{pmatrix} u_{z1} \equiv 0 \\ \varphi_{x1} \\ \varphi_{y1} \\ u_{z2} \equiv 0 \\ \varphi_{x2} \\ \varphi_{y2} \\ u_{z3} \equiv 0 \\ \varphi_{x3} \\ \varphi_{y3} \end{pmatrix}$$

Total generalized nodal forces

Superposed contributions

1. CST forces
2. DKT forces
3. hydrostatic pressure p acting on the current element area A^*
4. contact forces $\mathbf{F}_c$ and torques $\mathbf{T}_c$ (contact point $\mathbf{c}_i$),

$$\Delta \mathbf{F}_i = \begin{pmatrix} F_{xi}^{\mathrm{CST}} \\ F_{yi}^{\mathrm{CST}} \\ F_{zi}^{\mathrm{DKT}} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ p\frac{A^*}{3} \end{pmatrix} + \mathbf{F}_c/3$$

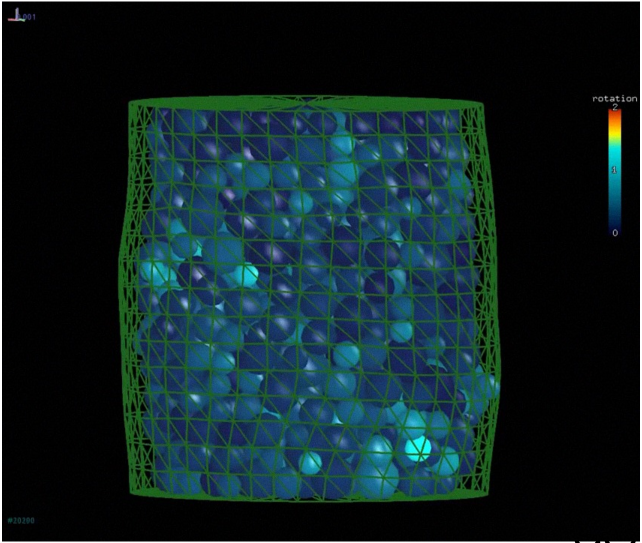
$$\Delta \mathbf{T}_i = \begin{pmatrix} F_{\varphi_{xi}}^{\mathrm{DKT}} \\ F_{\varphi_{yi}}^{\mathrm{DKT}} \\ 0 \end{pmatrix} + (\mathbf{x}_i - \mathbf{c}) \times \mathbf{F}_c + \mathbf{T}_c$$

Woo-specific considerations

- time-step: from stiffness matrix diagonal (`DynDt` engine), usually rather small
- no element-level damping, purely elastic behavior: contact damping will not damp the node motion

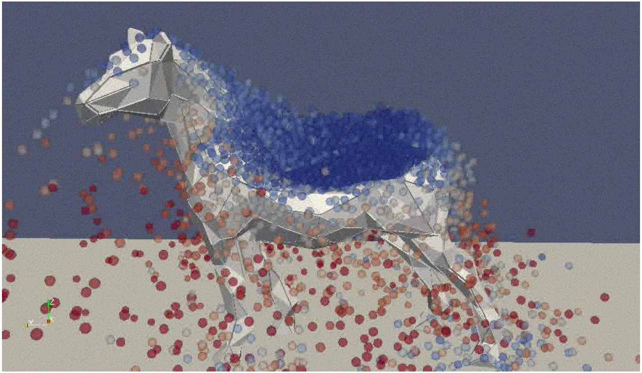
Example: triaxial test with elastic membrane

The membrane was modeled as membrane, using real stiffnesses. Hydrostatic pressure (from inside) was included in the simulation. Displacements are scaled, particles **do not** really get outside of the membrane. Uses the CylTriaxTest preprocessor.



Example: deforming horse

The horse is behaving as plate+shell (no volume), one can see buckling on its back, which recovers the original shape at the very end. Uses the FallingHorse preprocessor.

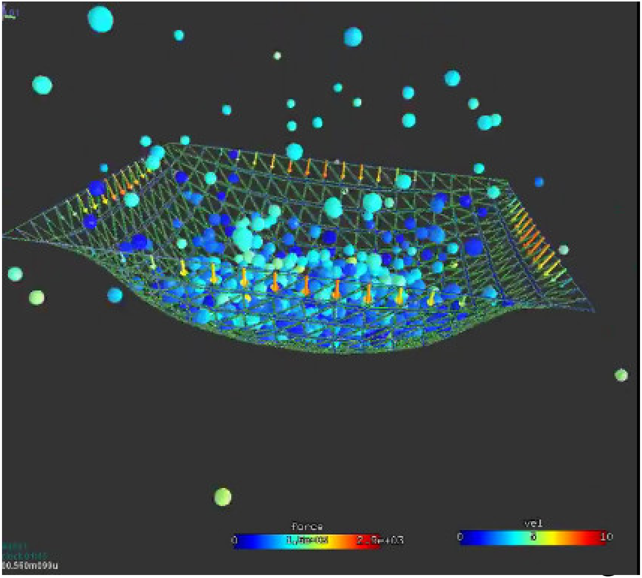


Example: soccer ball (bucky.py)

Model of a soccer ball, as membrane. Admittedly unrealistic. Air pressure inside the ball not included.

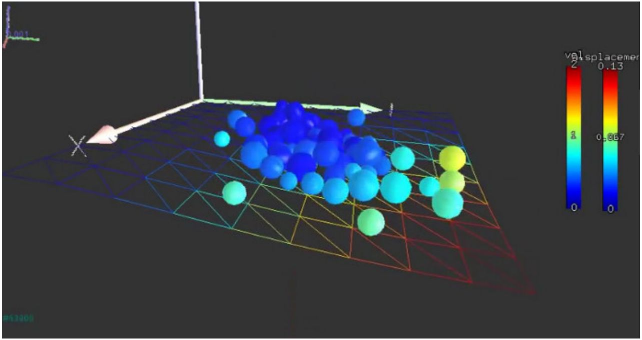
Example: plate elements

No bending in this example. Distributed under examples/membrane1.py.



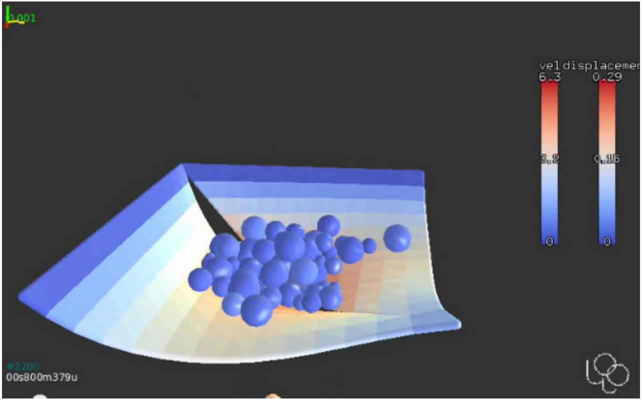
Example: plate+shell elements

Both stretching (plate) and bending (shell). Distributed under examples/membrane2.py.



Example: split membrane

The discontinuity was introduced by splitting nodes along the diagonal. Distributed as examples/membrane-split.py



Felippa, C. A., and B. Haugen. *A unified formulation of small-strain corotational finite elements: I. Theory.* Computer Methods in Applied Mechanics and Engineering 194.21 (2005): 2285-2335. http://www.colorado.edu/engineering/Aerospace/CAS/Felippa.d/FelippaHome.d/Publications.d/Report.CU-CAS-05-02.pdf

Carlos Felippa. *Introduction to Finite Element Methods (ASEN 5007)*. Online course at http://www.colorado.edu/engineering/cas/courses.d/IFEM.d/, 1998.

Batoz, JeanLouis, KlausJürgen Bathe, and LeeWing Ho. *A study of threenode triangular plate bending elements.* International Journal for Numerical Methods in Engineering 15.12 (1980): 1771-1812. http://web.mit.edu/kjb/www/Publications_Prior_to_1998/A_Study_of_Three-Node_Triangular_Plate_Bending_Elements.pdf

K. Kansara. *Development of membrane, plate and flat shell elements in java*. PhD thesis, Virginia Polytechnic, 2004.

http://scholar.lib.vt.edu/theses/available/etd-05142004-234133/unrestricted/Thesis.pdf

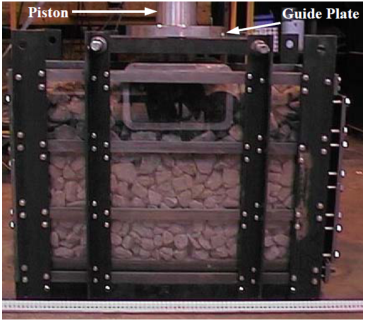
DEM simulation of ballast oedometric test

Jan Eliáš

Brno University of Technology, Faculty of Civil Engineering, Institute of Structural Mechanics, Czech Republic

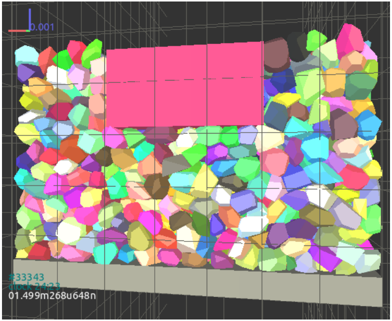
Motivation

- simulate behavior of railway ballast
- simulate ballast-sleeper interaction
- use realistically shaped elements
- include crushing of particles



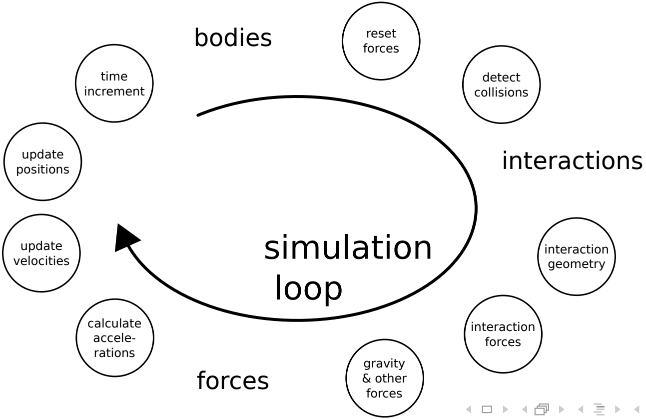
W. L. Lim, 2005, University of Nottingham

$\Leftrightarrow$



Discrete Element Method

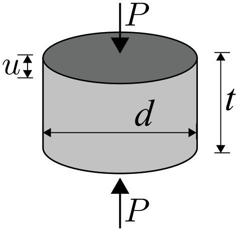
- originated in 1979 by Cundall & Strack
- bodies do not deform!
- approximation of interaction
- explicit dynamics



Experiment: large oedometric test

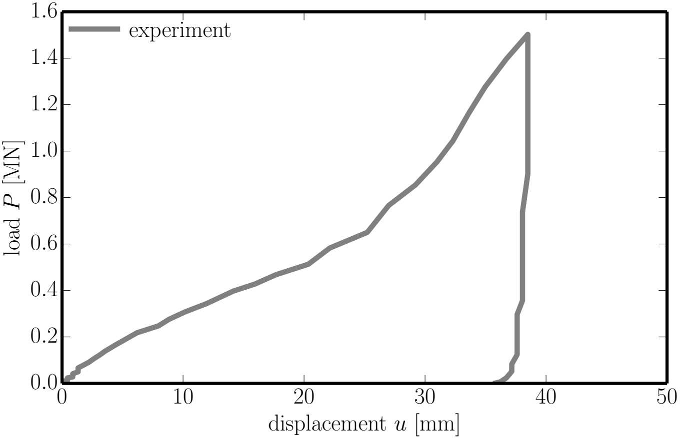
- performed by Lim & McDowell

  W. L. Lim & G. R. McDowell, 2005, Granular Matter.
- cylinder of diameter d=300 mm and depth t=150 mm
- compressed by force 1.5 MN (mean stress = 21.2 MPa) and then unloaded



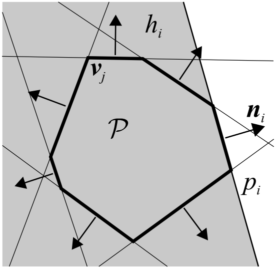
Experimentally recorded data

- initially vibrated on vibration table
- extensive crushing occured approx. from mean stress 1 MPa



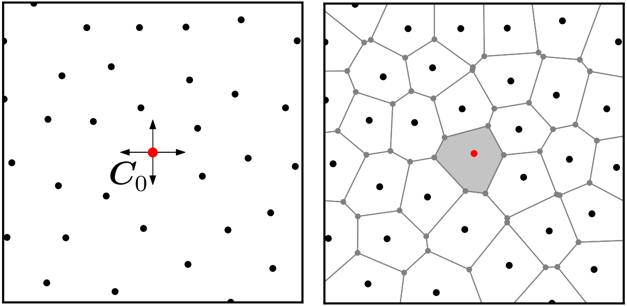
Polyhedron

- convex polyhedron is intersection of half-spaces: $\mathcal{P} = \cap_{i=1}^{n} h_i$
- half-space h_i is define by bounding plane p_i:
  $p_i \equiv a_i x + b_i y + c_i z + d_i = 0$
- h_i is set of all points at the negative side of bounding plane:
  $h_i = \{(x,\ y,\ z)\,,\mathrm{where}\, a_i x + b_i y + c_i z + d_i \leq 0\}$
- we use CGAL library to manipulate polyhedrons, compute convex-ghulls, etc.



Randomly-shaped polyhedral ballast grains

- initial central nucleus C_0 is placed at the origin
- nuclei are placed sequentially with random coordinates into the domain of size $5 \times 5 \times 5$ units
- minimal mutual distance $l_{\min}$ is restricted
- Voronoi tessellation is performed and cell associated with the central nucleus is taken



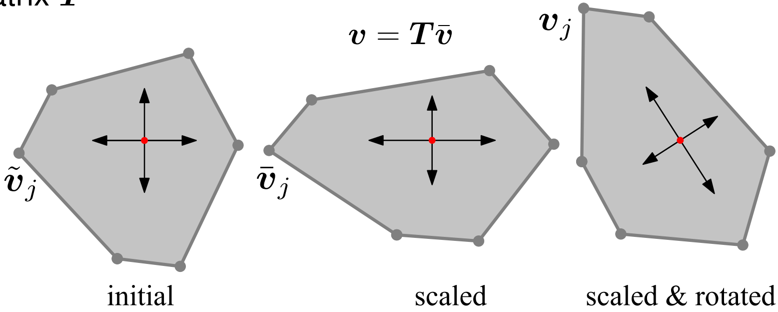
Randomly-shaped polyhedral ballast grains

- cell is scaled by scaling factor $\boldsymbol{s} = (s_x, s_y, s_z)$

$$\bar{\boldsymbol{v}} = \boldsymbol{s}\tilde{\boldsymbol{v}}$$

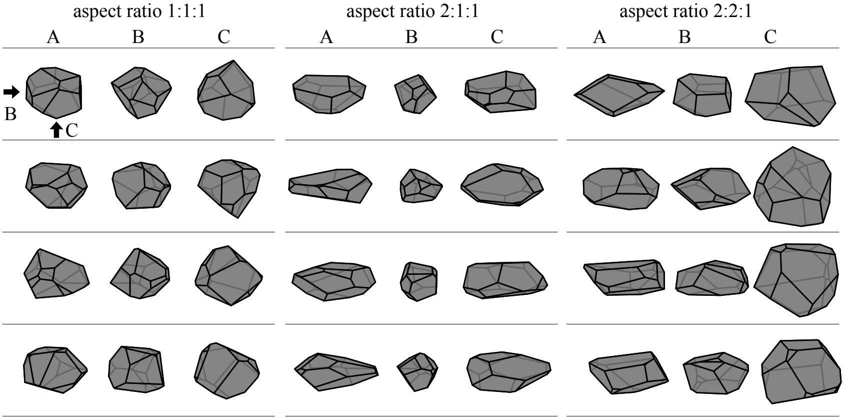
- finally, random orientation is applied using random rotation matrix $\boldsymbol{T}$

$$\boldsymbol{v} = \boldsymbol{T}\bar{\boldsymbol{v}}$$



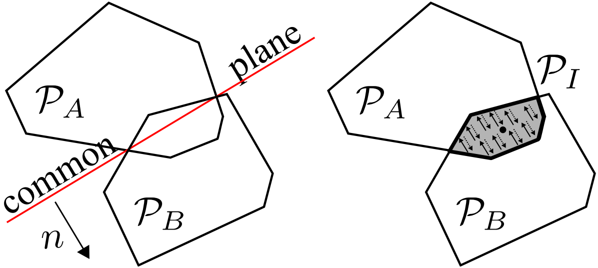
Controlling polyhedral shape

- ratio between scaling components $s_x : s_y : s_z$ controls grains shape
- absolute values of scaling components controls size of grains



Contact between polyhedrons

- only repulsive force which occurs when polyhedrons intersect
- magnitude of force related to intersecting volume
- widely used **Common Plane Method** reduces polyhedron–polyhedron contact to two polyhedron–plane contacts
- we attempted to use “exact” intersecting volume and repulsive force linearly dependent on it



Necessary algorithms

Contact detection

fast identification if two polyhedrons overlap: is $\mathcal{P}_A \cap \mathcal{P}_B$ empty?

Magnitude of normal force

linearly dependent of intersecting volume: $F_n = k_n V_I$, where V_I is volume of $\mathcal{P}_I = \mathcal{P}_A \cap \mathcal{P}_B$. How to compute $\mathcal{P}_I$?

Magnitude of shear force

standard incremental algorithm $+$ Coulomb friction

Normal direction and point of action

normal direction determined by least-square fitting of polyhedron shells intersection, point of action assumed in centroid of $\mathcal{P}_I$

Necessary algorithms

Contact detection

fast identification if two polyhedrons overlap: is $\mathcal{P}_A \cap \mathcal{P}_B$ empty?

Magnitude of normal force

linearly dependent of intersecting volume: $F_n = k_n V_I$, where V_I is volume of $\mathcal{P}_I = \mathcal{P}_A \cap \mathcal{P}_B$. How to compute $\mathcal{P}_I$?

Magnitude of shear force

standard incremental algorithm + Coulomb friction

Normal direction and point of action

normal direction determined by least-square fitting of polyhedron shells intersection, point of action assumed in centroid of $\mathcal{P}_I$

Necessary algorithms

Contact detection

fast identification if two polyhedrons overlap: is $\mathcal{P}_A \cap \mathcal{P}_B$ empty?

Magnitude of normal force

linearly dependent of intersecting volume: $F_n = k_n V_I$, where V_I is volume of $\mathcal{P}_I = \mathcal{P}_A \cap \mathcal{P}_B$. How to compute $\mathcal{P}_I$?

Magnitude of shear force

standard incremental algorithm $+$ Coulomb friction

Normal direction and point of action

normal direction determined by least-square fitting of polyhedron shells intersection, point of action assumed in centroid of $\mathcal{P}_I$

Necessary algorithms

Contact detection

fast identification if two polyhedrons overlap: is $\mathcal{P}_A \cap \mathcal{P}_B$ empty?

Magnitude of normal force

linearly dependent of intersecting volume: $F_n = k_n V_I$, where V_I is volume of $\mathcal{P}_I = \mathcal{P}_A \cap \mathcal{P}_B$. How to compute $\mathcal{P}_I$?

Magnitude of shear force

standard incremental algorithm $+$ Coulomb friction

Normal direction and point of action

normal direction determined by least-square fitting of polyhedron shells intersection, point of action assumed in centroid of $\mathcal{P}_I$

Test of overlapping

- search in set of possible candidates for separation plane; if none found, overlapping confirmed
  test of separation plane $s(a_s, b_s, c_s, d_s)$:

$$\begin{aligned} \forall\, \boldsymbol{x}_A \in \mathcal{P}_A \quad &: \quad a_s x_A + b_s y_A + c_s z_A \leq d_s \\ \forall\, \boldsymbol{x}_B \in \mathcal{P}_B \quad &: \quad a_s x_B + b_s y_B + c_s z_B \geq d_s \end{aligned}$$

- time can be saved by
  - saving and initial testing of separation plane from last step in case of no overlapping
  - saving of centroid $\boldsymbol{x}_c$ of intersecting polyhedron from last step and testing that it is inside both polyhedrons

$$\forall\, p_i(a_i,\, b_i,\, c_i,\, d_i) \in \mathcal{P}_A \text{ and } \mathcal{P}_B : a_i x_c + b_i y_c + c_i z_c \leq d_i$$

Calculation of intersecting polyhedron

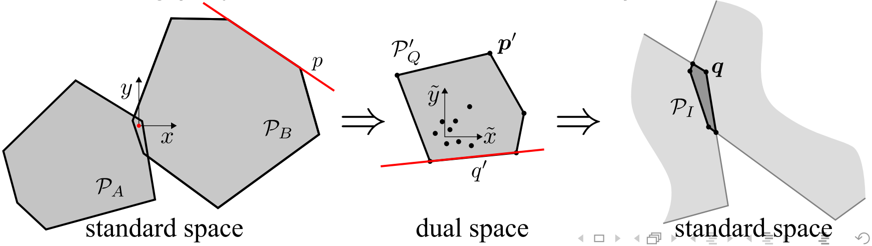
- through mapping between standard and dual space
- polyhedron planes p_i are projected to points $\boldsymbol{p}'$ in dual space

$$p \in \mathcal{P}_A \text{ and } \mathcal{P}_B \rightarrow \boldsymbol{p}' = (a/d,\, b/d,\, c/d)$$

- convex hull $\mathcal{P}'_Q$ of dualized planes $\boldsymbol{p}'$ is found
- bounding planes q' of $\mathcal{P}'_Q$ are again dualized to standard space

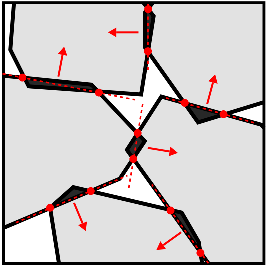
$$q' \in \mathcal{P}'_Q \rightarrow \boldsymbol{q} = (a'/d',\, b'/d',\, c'/d')$$

- intersecting polyhedron $\mathcal{P}_I$ is convex hull of points $\boldsymbol{q}$



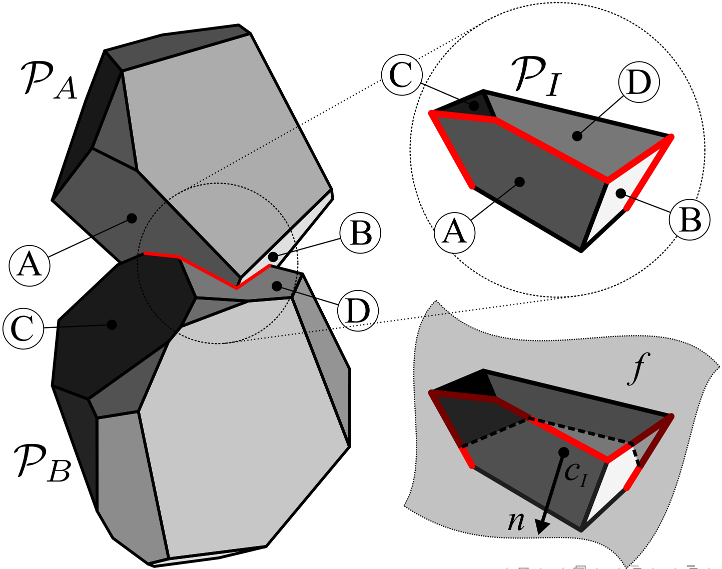
Estimation of normal direction

- analogy to 2D case where one can connect shell intersections by line and assume normal direction perpendicular



- shell intersection in 3D is a continuous enclosed non-planar piece-wise linear line
- we adopt least square linear fitting of it by plane
- normal direction is assumed perpendicularly to this fitted plane

3D sketch



Model parameters and compaction

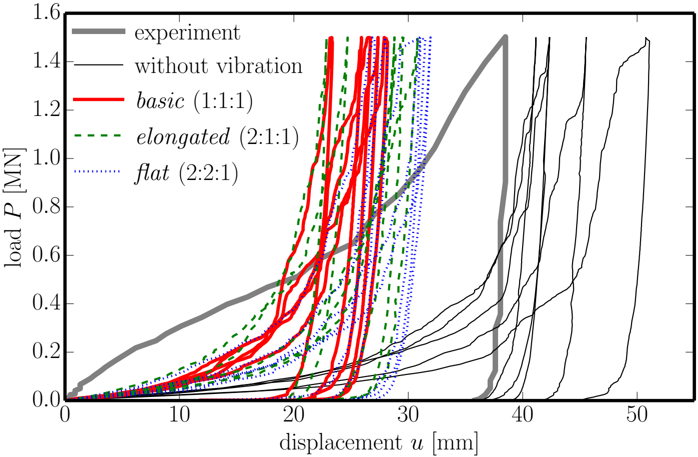
- chosen parameters of the materials

			ballast	steel
density	ρ	$\mathrm{kg/m^3}$	2600	7850
normal volumetric stiffness	k_n	$\mathrm{N/m^3}$	2×10^{13}	2×10^{14}
shear stiffness	k_s	$\mathrm{N/m}$	2×10^{8}	2×10^{9}
angle of internal friction	ϕ	-	0.6	0.4

- polyhedrons are generated sequentially with no overlapping in larger volume and left to freely fall into steel cylinder
- high level of compaction is hard to achieve in simulation
- following items should help
  - increased gravity $5\times$
  - decreased angle of internal friction to 0.1
  - vibrating by adding changing horizontal acceleration

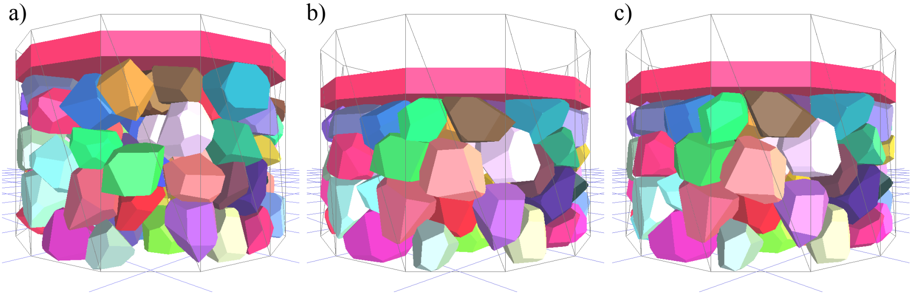
Comparison to experiment

- no crushing present in the model, loading curvature mostly caused by low compaction level



Views at different stages of simulation

- initial stage, after free fall and vibration (a)
- at the maximum load (b)
- after complete unloading (c)



Crushing in numerical model

- crushing of grains needs to be included
- incorporated via splitting of polyhedron pieces by plane running through centroid
- criterion based on principal stresses inside the grain:

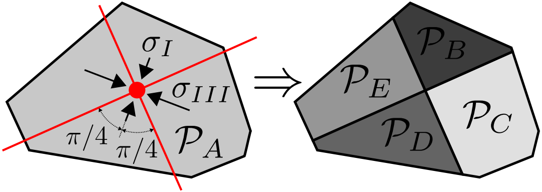
$$\sigma_{ij} = \sum_{k=1}^{N} F_i r_j \quad \Rightarrow \quad \sigma_I > \sigma_{II} > \sigma_{III}$$

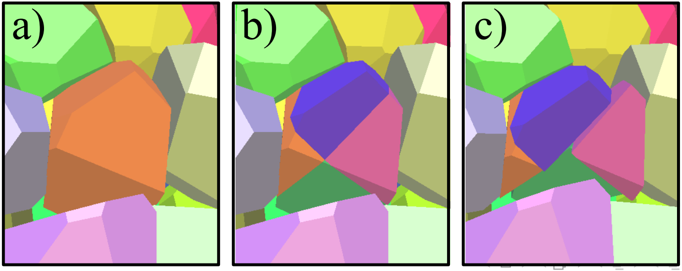
- splitting stress $\sigma_s = -\sigma_{III} + \sigma_I$
- size dependent strength $f_t = f_{t0}/r_{eq}$, where r_{eq} is equivalent radius (S. Lobo-Guerrero, L. E. Vallejo, 2006)

$$r_{eq} = \sqrt[3]{\frac{3V}{4\pi}}$$

Crushing in numerical model

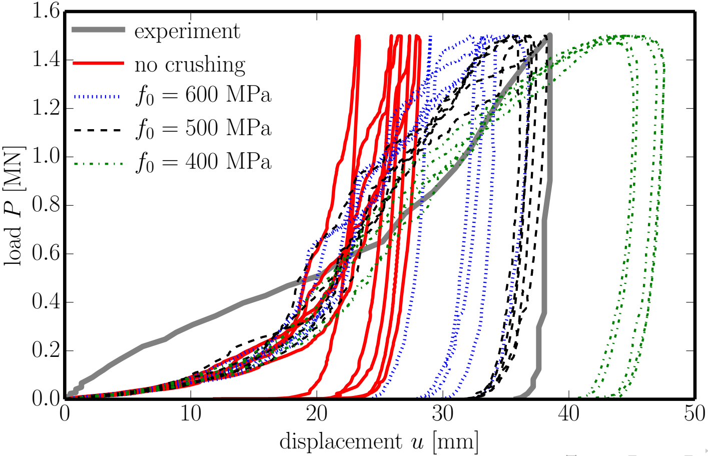
- splitting into four pieces by planes parallel to plane σ_{II} and under angle $\pi/4$ from σ_I and σ_{III} directions





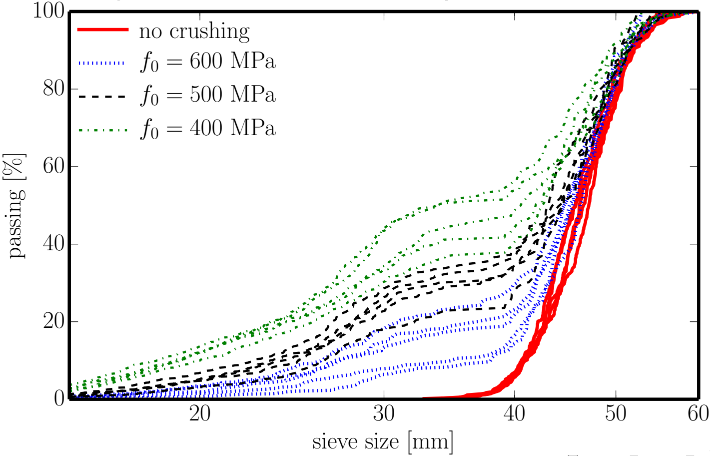
Comparison to experiment

- three material strengths tested: 400, 500 and 600 MPa
- pieces with volumes lower than 1 cm^3 were removed from the simulation



Sieve curve after crushing

- includes small particles with volume under 1 cm^3
- plateau between sieve sizes 30 and 40 mm is observed because the grains are crushed into pieces of similar sizes



Conclusions

- simple method to generate convex randomly shaped grains
- possibility to control aspect ratio of grains
- repulsive force estimated from volume of intersecting polyhedron
- normal direction estimated from least-square fitting of shells intersection by plane
- model used to simulate large oedometric test performed on railway ballast
- crushing of grains can be simply done in geometrical sense, problem is to develop correct criterion
- many thanks to YADE and CGAL developers

Grids elements in Yade-Dem as connected cylinders : from development to use.

F. Kneib

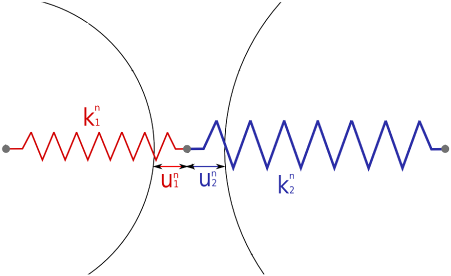
Irstea, 3SR

1st Yade Workshop

Sommaire

Sommaire

Main idea : already implemented through **remote cohesive contact**.



Kinematic by 6 DoF : **normal** displacement (1 DoF), **shear** displacement (2 DoF), **bending** rotation (2 DoF) and **twisting** rotation (1 DoF).

– *Illustration : 01-cohesion.py* –

Warning :

Internal behaviour of the whole beam as described highly depends on the inter-element distance.

Solution : need to ajust all stiffness according to the **Beam Theory**.
Analytic solution for each DoF :

Normal stiffness :
$$k_n = \frac{EA}{L}$$

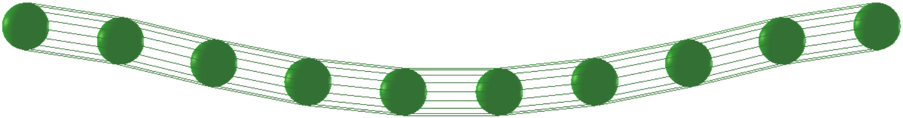
Bending stiffness :
$$k_b = \frac{EI}{L}$$

Twist stiffness :
$$k_t w = \frac{GI}{L}$$

Shear stiffness :
$$k_s = \frac{12EI}{L^3}$$

– Illustration : 02-impact.py –

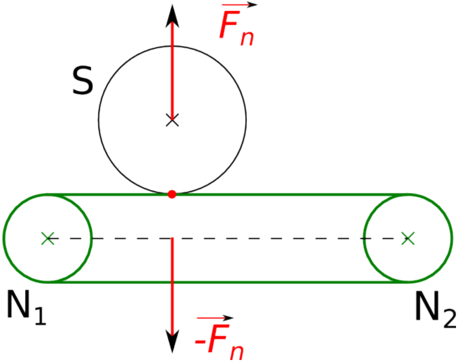
Need of a new specific set of classes to **handle external contacts** and avoid rugosity.
Bruno Chareyre made this many years ago, introducing the contact tracking.



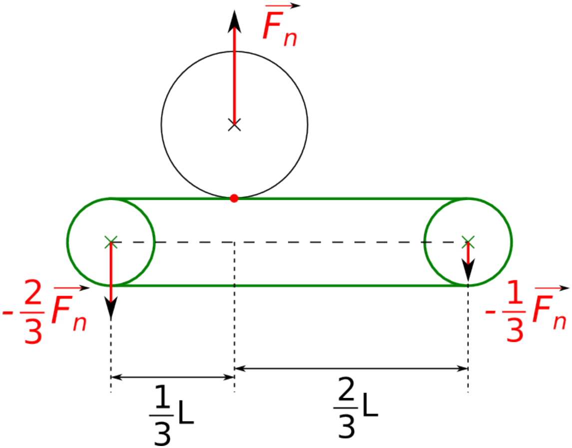
– *Illustration : 03-linear.py* –

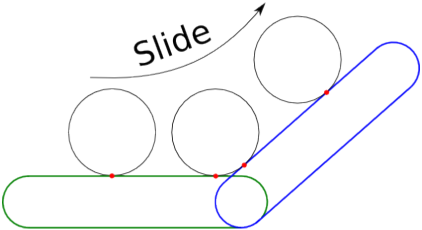
Force dispatching between the two nodes of a cylinder according to the contact position.

⇒ Need to write specific contact laws.

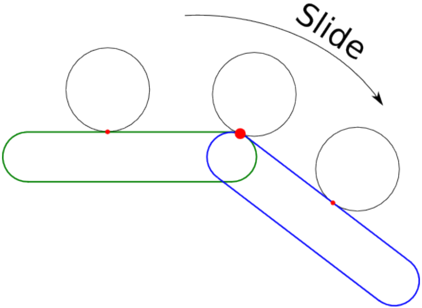


Force dispatching between the two nodes of a cylinder according to the contact position.
⇒ Need to write specific contact laws.





Slide over a **concave** grid connection : double contact considered as **rightful**.
⇒ No contact tracking.

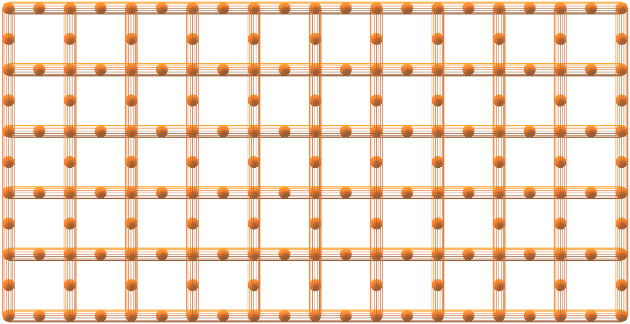


Slide over a **convex** grid connection : **avoid double contact** and **track** the tangential force (history).

Generalization of the chained cylinders : **grids**.

Each node can be connected to many other.

Allows arbitrary construction of **nodes** connected via **connections**.



Example of a regular grid.

– Illustrations : 04-grid.py and 05-pull.py –

Last feature : handle the contact between two grids.
Contact tracking not implemented, because it's difficult (but possible), and more ressource expansive.

– Illustration : 06-fish.py –

Sommaire

- **ChainedCylinder** classes are **deprecated**, don't use them. Use the **GridNode/GridConnection** instead.
- Internal grid behaviour : **handled by GridNode**

Shape	*GridNode*
Ig2	*Ig2_GridNode_GridNode_GridNodeGeom6D*
Material	*CohFrictMat*
Ip2	*Ip2_CohFrictMat_CohFrictMat_CohFrictPhys*

- For sphere - grid behaviour : **handled by GridConnection**

Shape	*GridConnection*
Bounding box	*Bo1_GridConnection_Aabb*
Ig2	*Ig2_Sphere_GridConnection_ScGridCoGeom*
Material	*(Coh)FrictMat*
Ip2	*Ip2_(Coh)FrictMat_(Coh)FrictMat_(Coh)FrictPhys*

- For grid - grid behaviour : **handled by GridConnection**

Shape	*GridConnection*
Bounding box	*Bo1_GridConnection_Aabb*
Ig2	*Ig2_GridConnection_GridConnection_GridCoGridCoGeom*
Material	*FrictMat*
Ip2	*Ip2_FrictMat_FrictMat_FrictPhys*

- Update *O.engines* by adding previous classes.
- Generate *GridNode* like spheres with the Python function : *utils.gridNodes(...)*.
- Generate *GridConnection* to bind two *GridNodes* : *utils.gridConnection(i, j, ...)*, where *i, j* are the ids of the nodes.

3SR :

- Bruno Chareyre
- Pascal Villard

Irstea

- Franck Bourrier
- Ignacio Olmedo-Manich
- David Toe

Related publication :

- Bourrier, F. ; Kneib, F. ; Chareyre, B. Fourcaud, T. (2013) "Discrete modeling of granular soils reinforcement by plant roots", *Ecological Engineering* , 61, Part C, 646 - 657.

DEM – LBM coupling … in YADE

LBM for: Lattice Boltzmann Method

Luc Sibille
3SR, Université Grenoble Alpes, CNRS

1st YADE Workshop
Grenoble – 7-9 July 2014

DEM-LBM coupling

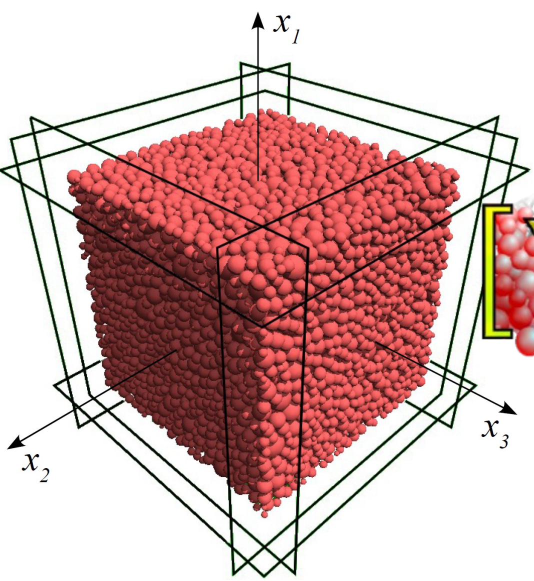
- **Description of the solid phase at the particle scale**
- **Description of the fluid dynamic in the inter-particle space**

Solid phase: Discrete Element Method
DEM, *Yade Software*

- Contact stiffnesses
- Contact friction angle
- Contact adhesion

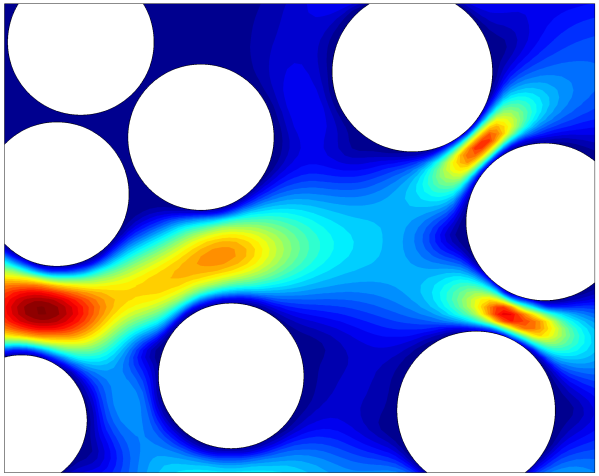
Fluid phase
Lattice Boltzmann Method (LBM)

- Fluid viscosity
- position of each solid particle explicitely described



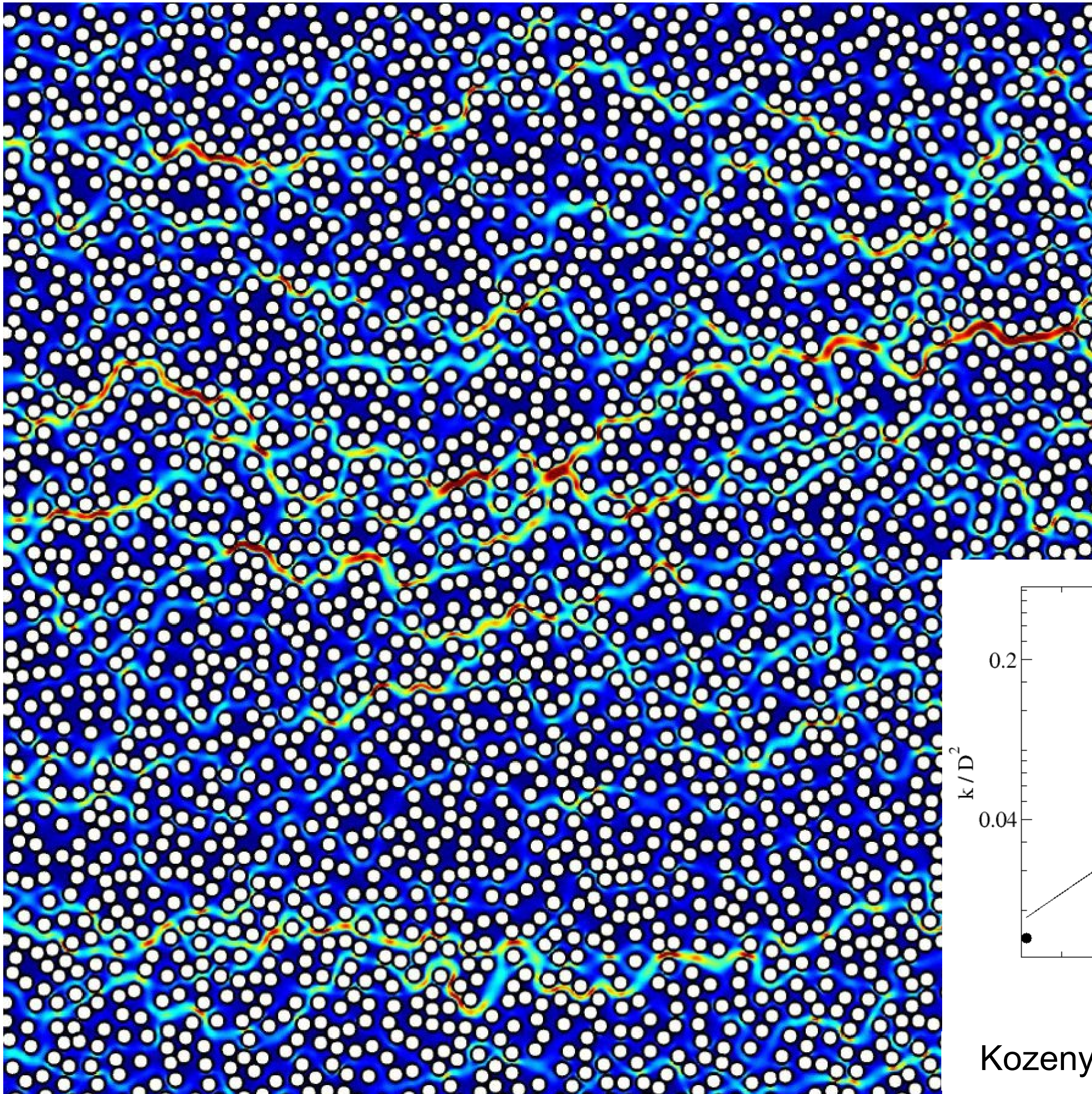
Particle positions and velocities

hydro-dynamic forces

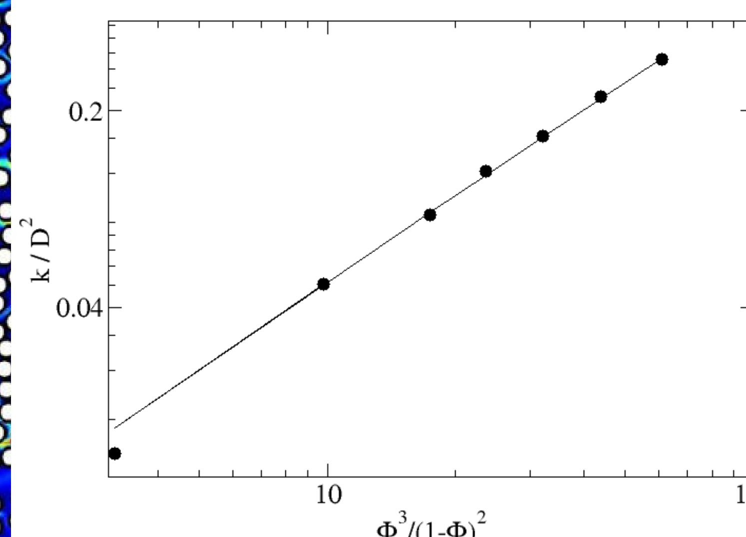


No assumption on fluid/solid interactions: permeability, drag forces, etc... result from the coupling.

Hydraulic conductivity through a fix granular assembly (LBM only)

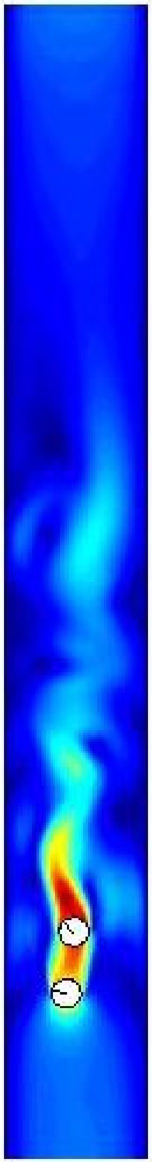


(Lominé, 2010)



Kozeny-Carman relation $\frac{k}{D^2} \propto \frac{\Phi^3}{(1-\Phi)^2}$

Full coupling



Free fall under gravity

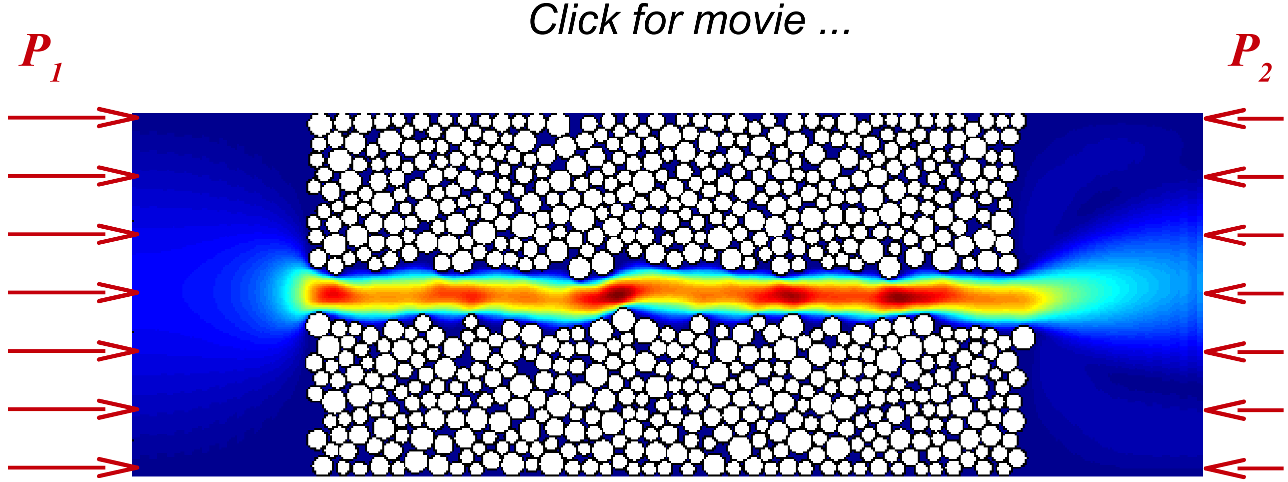
Drafting-kissing-tumbling process

Click for movie ...

(Lominé, 2011)

Hole erosion in a granular packing

Click for movie ...



800 solid particles; fluid lattice of 335 000 nodes

(Lominé et al. IJNAMG, 2013)

Lattice Boltzmann Method

Þ **Based on the probability density or distribution function** $f(\vec{x},t)$

representing the probability of finding a molecule (or particle) around position $\vec{x}$ at time t with a given **momentum**.

Þ **The BGK (Bhatnagar-Gross-Krook, 1954) collision operator**

describes the time and spatial evolution of a distribution function (i.e. of momentum):

$$f(\vec{x},t^{+}) = f(\vec{x},t) - \frac{1}{\tau}\left[f(\vec{x},t) - f^{eq}(\vec{x},t)\right]$$

with $\tau = 3\nu\, dt/h^{2} + 1/2$

Þ **Transfer of momentum from the solid particles to the fluid at solid boundaries**

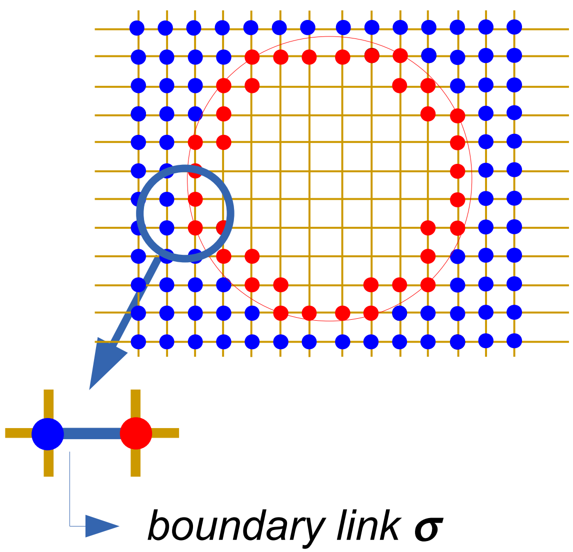
distribution functions affected by a terms involving the solid boundary velocity $\vec{V}_b$

$$f_{-\sigma i}(\vec{x}_{FB}, t + \mathrm{d}t) = f_{\sigma i}(\vec{x}_{FB}, t^{+}) - 2\alpha_i \vec{V}_b . \vec{e}_i$$

Þ **Force applied by the fluid on the solid**

results from the time derivation of the momentum exchange at solid boundaries

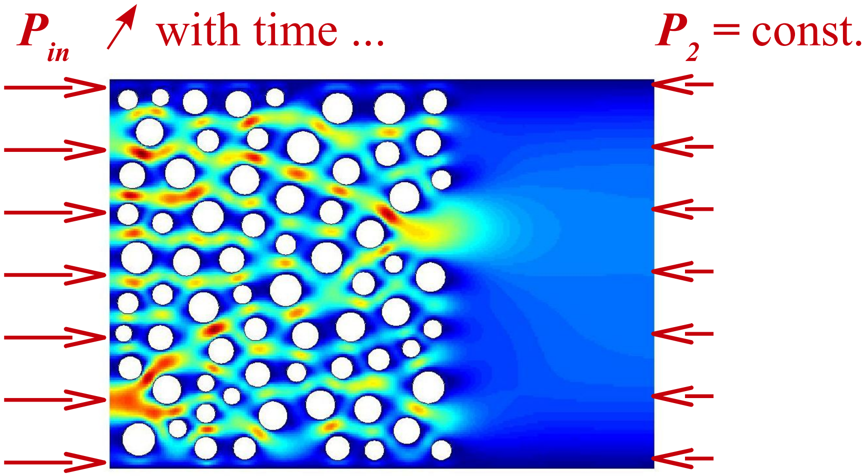
$$\vec{F}_{\sigma}(\vec{x}, t + \frac{1}{2}\mathrm{d}t) = 2\frac{\Omega}{\mathrm{d}t}\left[f_{\sigma i}(\vec{x}, t^{+}) - \alpha_i \vec{V}_b . \vec{e}_i\right] \vec{e}_{\sigma i}$$



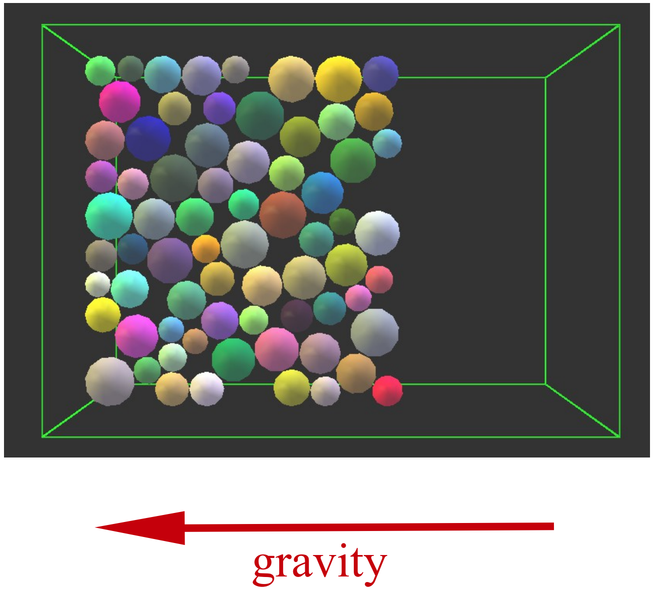
DEM-LBM coupling in YADE

2D example: hydraulic heave

Geometry seen by the LBM → **2D!**



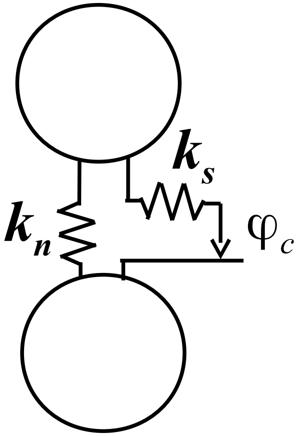
Geometry seen by the DEM



ν : kinematic viscosity.

N_x : number of lattice division (h lattice spacing)

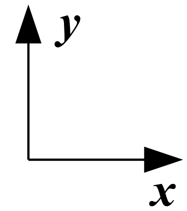
τ : relaxation time

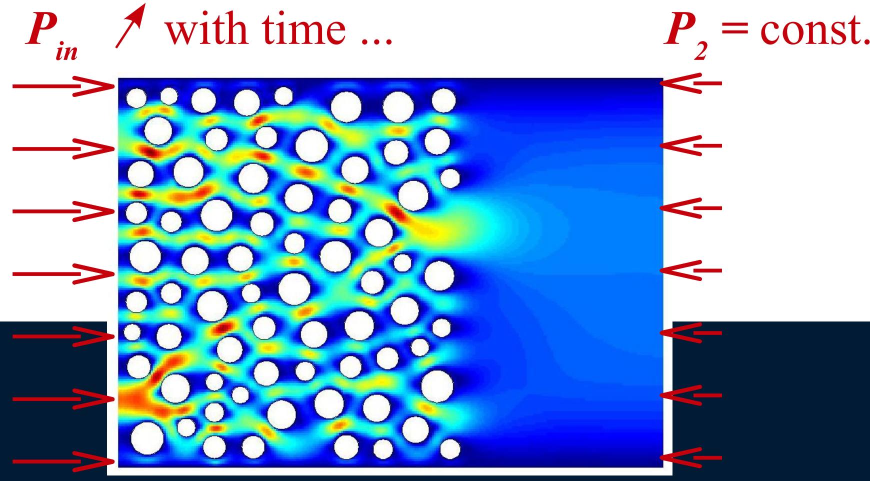


LBM time step is automatically computed and DEM time step adjusted according to LBM one

```
## Build of the engine vector
O.engines=[

        ForceResetter(),
        InsertionSortCollider([Bo1_Sphere_Aabb(),Bo1_Box_Aabb()]),
        InteractionLoop(
                [Ig2_Sphere_Sphere_ScGeom(),Ig2_Box_Sphere_ScGeom()],
                [Ip2_FrictMat_FrictMat_FrictPhys()],
                [Law2_ScGeom_FrictPhys_CundallStrack()]
        ),


        HydrodynamicsLawLBM(    EngineIsActivated=False,
                                WallYm_id=0,
                                WallYp_id=1,
                                WallXm_id=2,
                                WallXp_id=3,
                                WallZp_id=5,
                                WallZm_id=4,
                                useWallYm=1, # Set true if you want that the LBM see the wall in Ym
                                useWallYp=1, # Set true if you want that the LBM see the wall in Yp
                                YmBCType=2,  # Boundary condition for the wall in Ym (-1: unused, 1: pressure condition,
                                YpBCType=2,  # Boundary condition for the wall in Yp (-1: unused, 1: pressure condition,

                                useWallXm=0, # Set true if you want that the LBM see the wall in Xm
                                useWallXp=0, # Set true if you want that the LBM see the wall in Xp
                                XmBCType=1,  # Boundary condition for the wall in Xm (-1: unused, 1: pressure condition,
                                XpBCType=1,  # Boundary condition for the wall in Xp (-1: unused, 1: pressure condition,
                                LBMSavedData='spheres,velXY,rho,nodeBD', # Can use velocity,velXY,forces,rho,bodies,node
                                                                         # observednode,contacts,spheres
                                tau=1.1,
                                dP=(0,0,0),
                                IterSave=200,
                                IterPrint=100,
                                RadFactor=0.6,  # The radius of particles seen by the LBM engine is reduced here by a fa
                                                # to allow flow between particles in this 2D case.
                                Nx=250,         # The number of grid division in x direction
                                Rho=1000,       # Fluid density
                                Nu=1.0e-6,      # Fluid kinematic viscosity
                                periodicity='', # x, y, or z
                                bc='',          # not used
                                applyForcesAndTorques=True, #Switch to apply forces and torques
                                label="Elbm"
        ),


        NewtonIntegrator(damping=0.2,gravity=(Gravity,0.,0.),label="ENewton"),
]
```

LBM engine

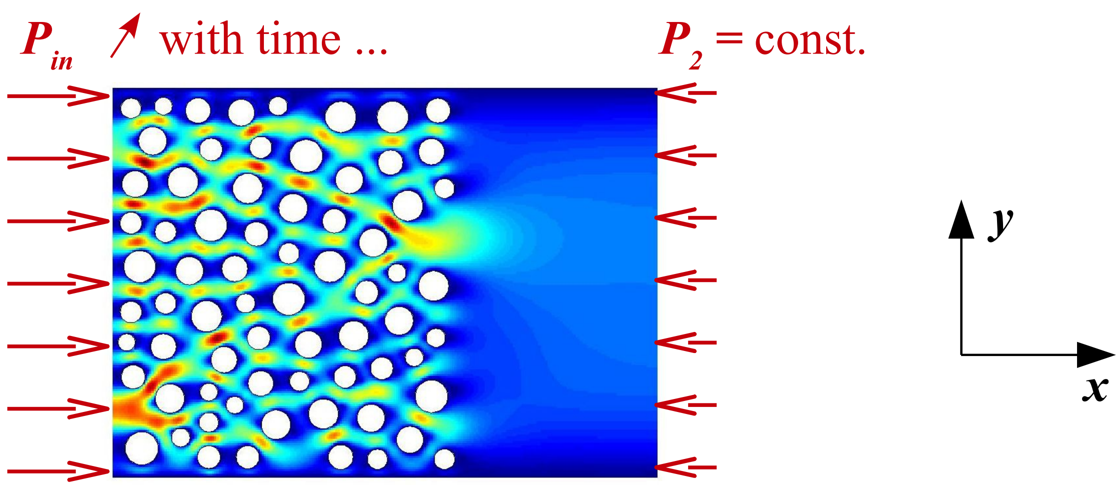





```
M(    EngineIsActivated=False,
      WallYm_id=0,
      WallYp_id=1,
      WallXm_id=2,
      WallXp_id=3,
      WallZp_id=5,
      WallZm_id=4,
      useWallYm=1, # Set true if you want that the LBM see the wall in Ym
      useWallYp=1, # Set true if you want that the LBM see the wall in Yp
      YmBCType=2,  # Boundary condition for the wall in Ym (-1: unused, 1: pressure condition, 2: velocity condition)
      YpBCType=2,  # Boundary condition for the wall in Yp (-1: unused, 1: pressure condition, 2: velocity condition)

      useWallXm=0, # Set true if you want that the LBM see the wall in Xm
      useWallXp=0, # Set true if you want that the LBM see the wall in Xp
      XmBCType=1,  # Boundary condition for the wall in Xm (-1: unused, 1: pressure condition, 2: velocity condition)
      XpBCType=1,  # Boundary condition for the wall in Xp (-1: unused, 1: pressure condition, 2: velocity condition)
      LBMSavedData='spheres,velXY,rho,nodeBD', # Can use velocity,velXY,forces,rho,bodies,nodeBD,newNode,observedptc,
                                               # observednode,contacts,spheres
      tau=1.1,
      dP=(0,0,0),
      IterSave=200,
      IterPrint=100,
      RadFactor=0.6,  # The radius of particles seen by the LBM engine is reduced here by a factor RadFactor=0.6
                      # to allow flow between particles in this 2D case.
      Nx=250,         # The number of grid division in x direction
      Rho=1000,       # Fluid density
      Nu=1.0e-6,      # Fluid kinematic viscosity
      periodicity='', # x, y, or z
      bc='',          # not used
      applyForcesAndTorques=True, #Switch to apply forces and torques
      label="Elbm"
```

Gradual increase of the inlet pressure

```
#definition of a runnable python function to increase progressively the fluid pressure gradient.
def IncrDP():
        currentDP=Elbm.dP
        Elbm.dP=(currentDP[0]+1.0/10000.0,0,0)
        #O.engines[6].dP=currentDP+1.0/100.0
        if not(O.iter%100):
                print "dP=", Elbm.dP[0]
```

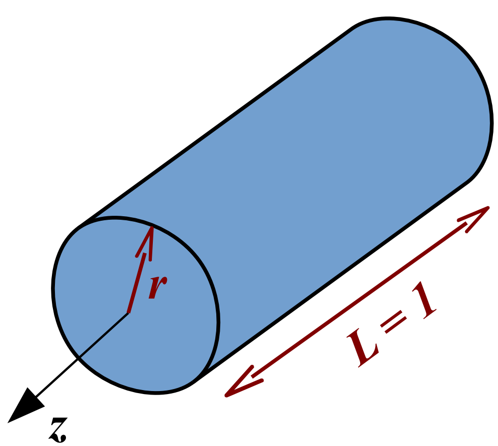


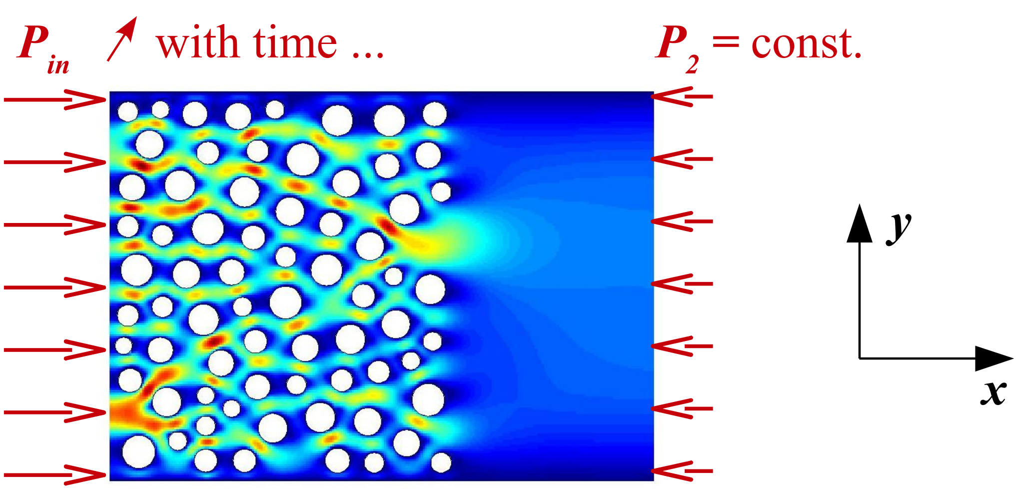
DEM-LBM coupling in YADE

2D model means 2D inertia!

```
# loop over bodies to change their 3D inertia into 2D inertia (by default in YADE particles are 3D spheres, here we want to cylinders with a length=1).
for s in O.bodies:
	if isinstance(s.shape,Box): continue

	r=s.shape.radius
	oldm=s.state.mass
	oldI=s.state.inertia

	m=oldm*3./4./r
	s.state.mass=m

	s.state.inertia[0] = 15./16./r*oldI[0]  #inertia with respect to x and y axes are not used and the computation here is wrong
	s.state.inertia[1] = 15./16./r*oldI[1]  #inertia with respect to x and y axes are not used and the computation here is wrong
	s.state.inertia[2] = 15./16./r*oldI[2]  #only inertia with respect to z axis is usefull
```

Two-dimendisionnal particle
= cylinder with unity length

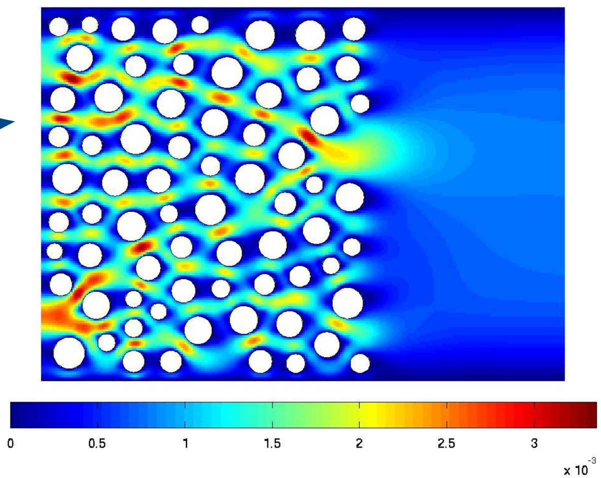




Post-processing and visualisation

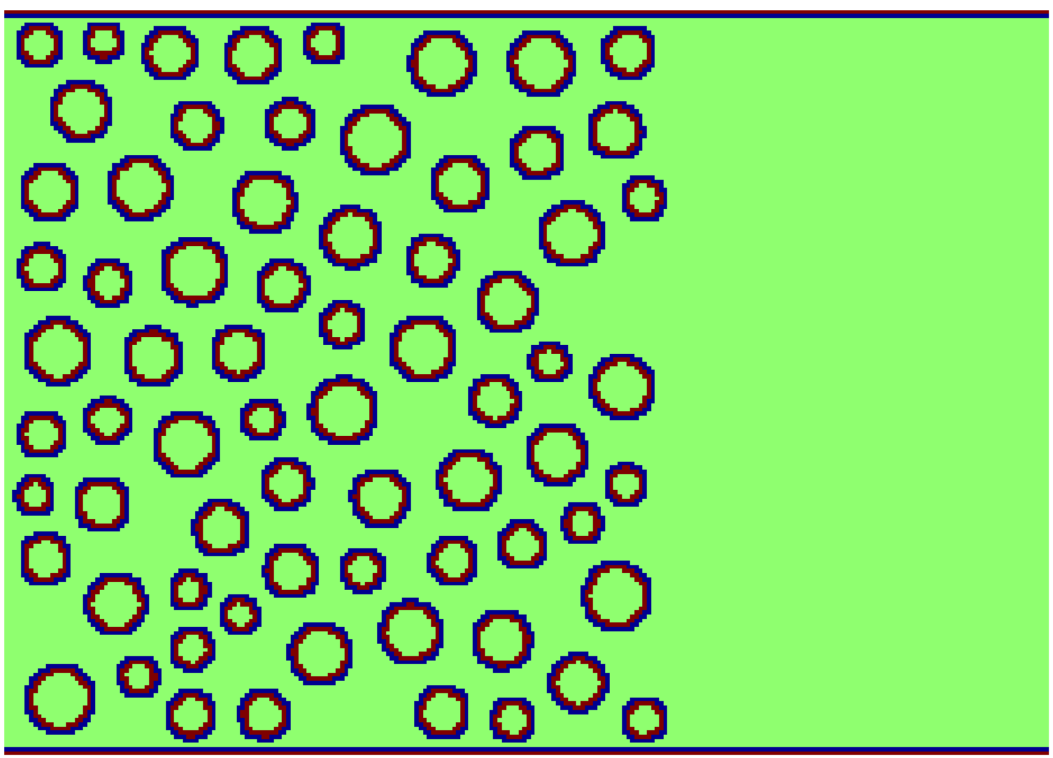
- Not all the variables are reachable through Python ...
- Data are saved in text files.
- Visualisation with matlab “yadeLBMvisu.m”
  (but it could be with Python or anything else)

Velocity fields

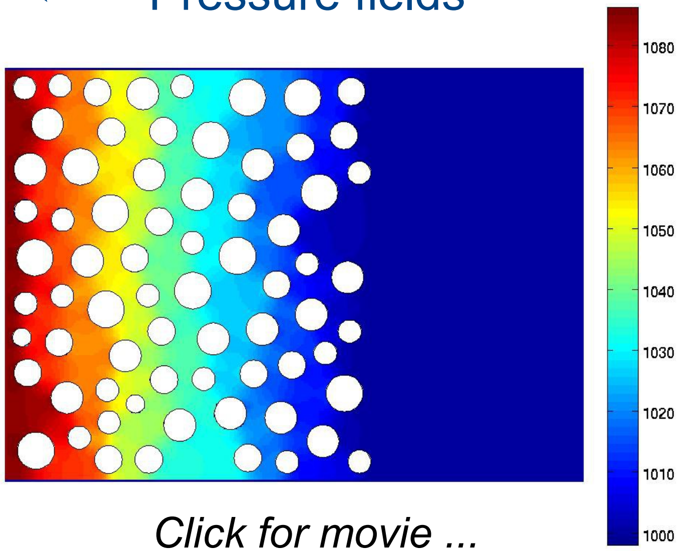


Click for movie ...

Qualitative information LBM nodes ...



Pressure fields



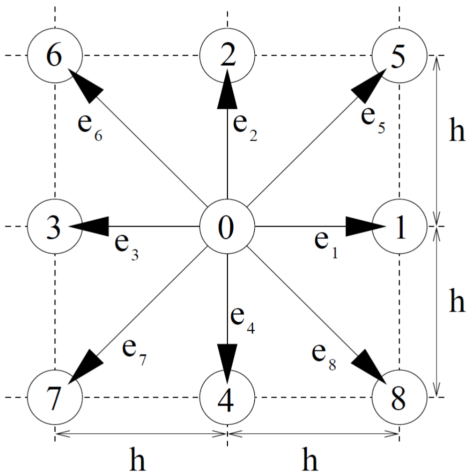
Click for movie ...

Some limitations:

Þ Incompressible flow assumption and conservation of mass, momentum and energy hold for:

- low Mach number: $M = \frac{v_{max}}{C} < 0.1\ (0.01)$ with $C = h / dt$ the lattice velocity
- small density variations (in classical LBM the fluid is slightly compressible):

$$p = c_s^2 \rho$$



... describing large pressure gradient may be quite difficult ...

Þ Fluid viscosity depends on time and space discretization:

$$\nu = \frac{1}{3}\left(\tau - \frac{1}{2}\right) Ch$$

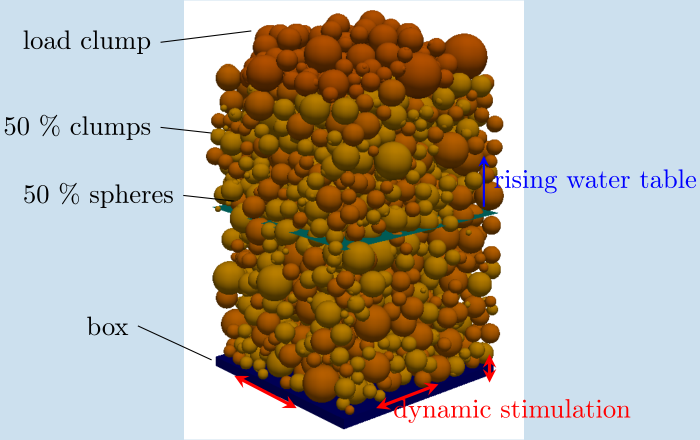
... fixing the numerical parameters may be quite difficult ...

Micromechanical modelling of lique-faction sensible sands

Dipl.-Geophys. Christian Jakob

Model type 1

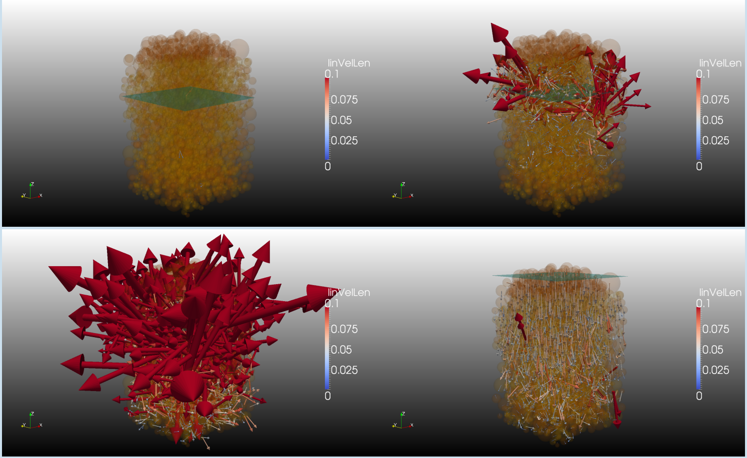
Complex periodic model



Periodic model with clumps, dynamic stimulation and liquid bridges

Model type 1

Instability

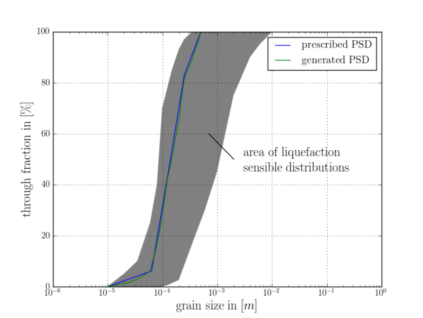


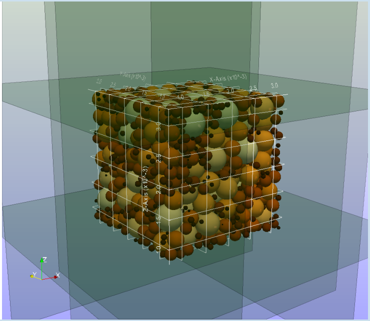
Erase liquid bridges from bottom to top

Model type 2

TriaxialStressController

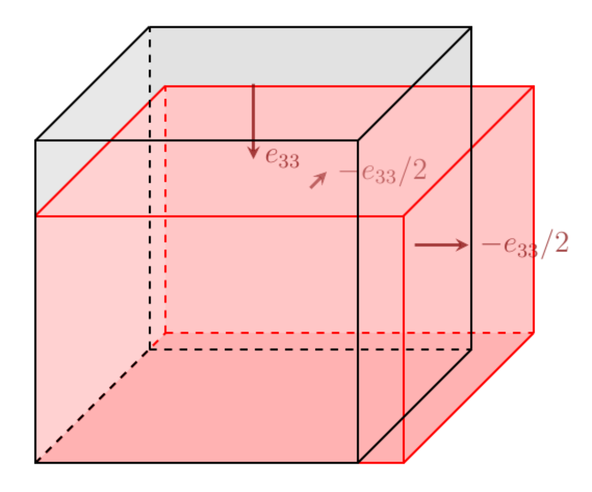
- 2,5 mm edge length
- 1300 particles
- $\sigma_v = \sigma_h = 200$ kPa
- Porosity 44,7 % ($\mu = 10$)





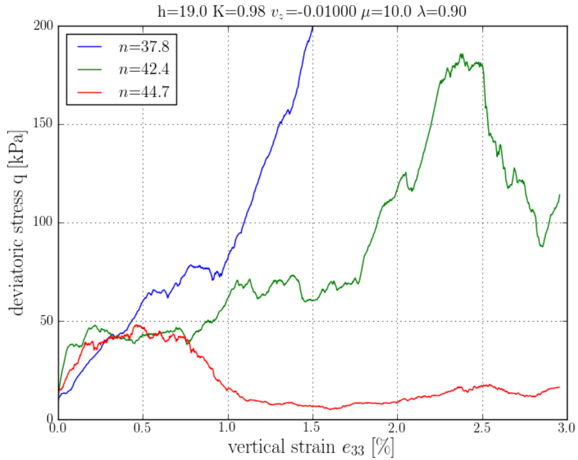
Model type 2

Example: Volume-constant deformation



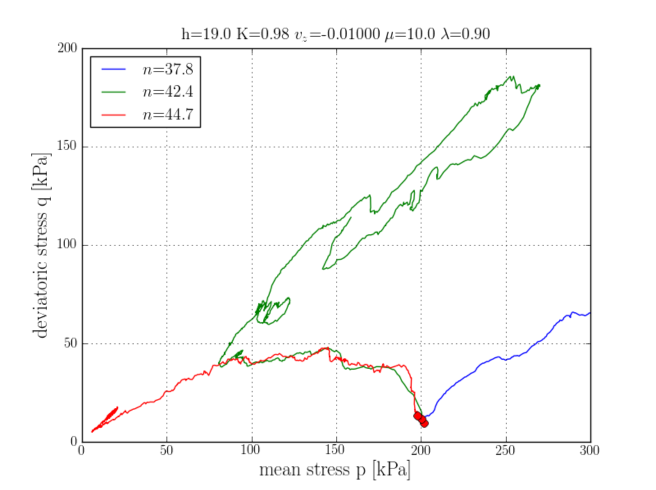
Model type 2

Stress-Strain relationship



Model type 2

Stress path



DEM-fluid coupling applied to bedload transport

Maurin R.[1], Chauchat J.[2], Chareyre B.[3], Frey P.[1]

[1] Irstea Grenoble, [2]LEGI , [3]3SR

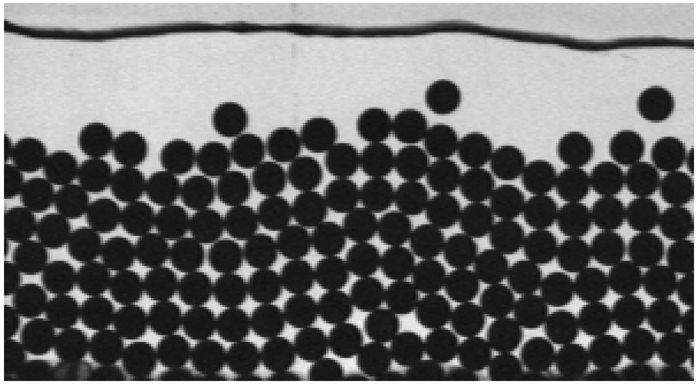
Yade workshop - july 8th, 2014

Bedload

/media/F25F-4420/Presentation3SR/bedload.mp4

markbsplace.net

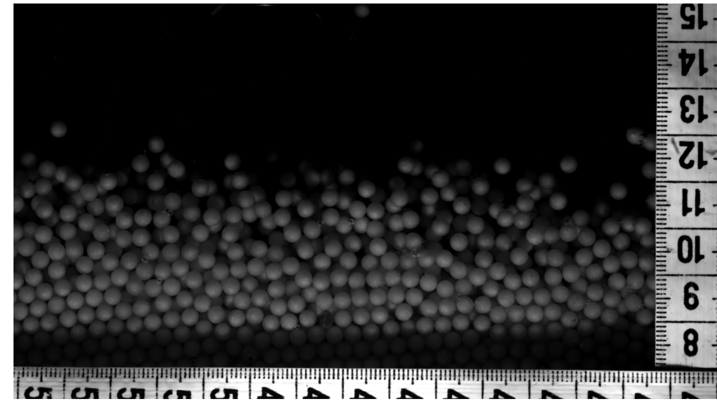
- Part of the sediment transport in « contact » with the bed
  → Rolling, sliding, saltating
- Turbulent flow, dynamic phenomenon
- In link with river/mountain stream/coastal sediment transport
  → Sediment/solid flux
- Incomplete understanding/predictions
- Importance of granular behavior in the process
- Interests in modeling :
  studying at the particle scale
  3D



Numerical model

Idea :

- Particle scale
- Simple fluid description



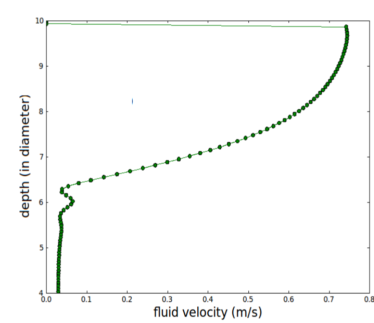
Model principle

Particle phase : DEM

Fluid phase : momentum balance
Averaged in volume, unidirectional $u_x(z)$
RANS 1D

- Turbulence model : mixing length

- Coupling : drag force

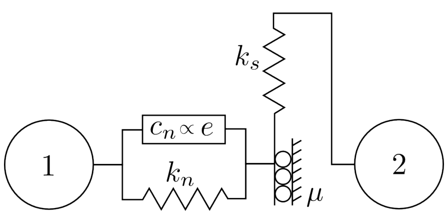


Numerical model

In practice in Yade

Fluid code
Fortran + Boost python

Library nsmp1d_yade with function to evaluate the fluid



Classical in granular flows
Quasi-static to dynamic

Yade

Evaluate ϕ,<fd>

import nsmp1d_yade
nsmp1d_yade(ϕ,<fd>)

Framework definition
Spring-dashpot law :
Law2_ScGeom_ViscElPhys_Basic()
GSTS(viscEl = True)

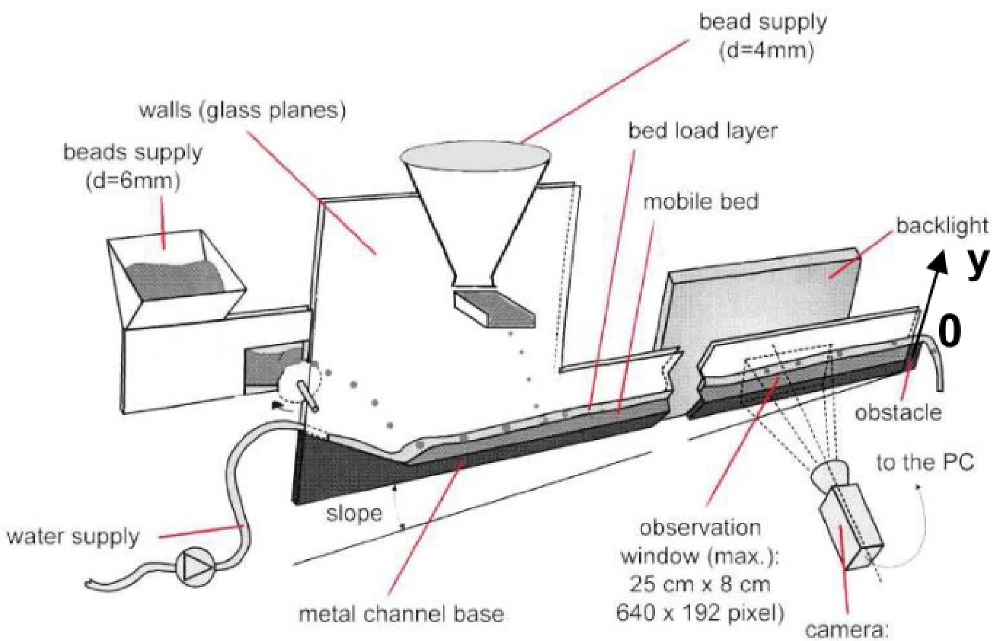
HydroForceEngine(uf)

$u^f(z)$

Evaluate the drag, lift and buoyancy for each particle

Experimental context

Movie
/home/raphael/Bureau/presentationMGMAS/ExpReducedd10.avi



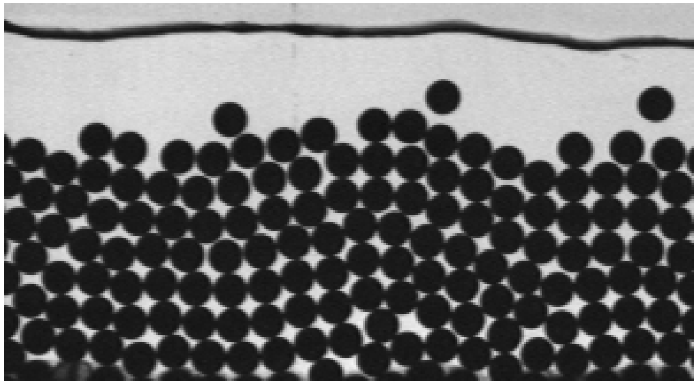
Experimental set-up

- Quasi **2D inclined channel** (slope 0.1)
- Supercritical **free-surface turbulent** flow on erodible bed
- Spherical glass particles diameter 6 mm

Measures

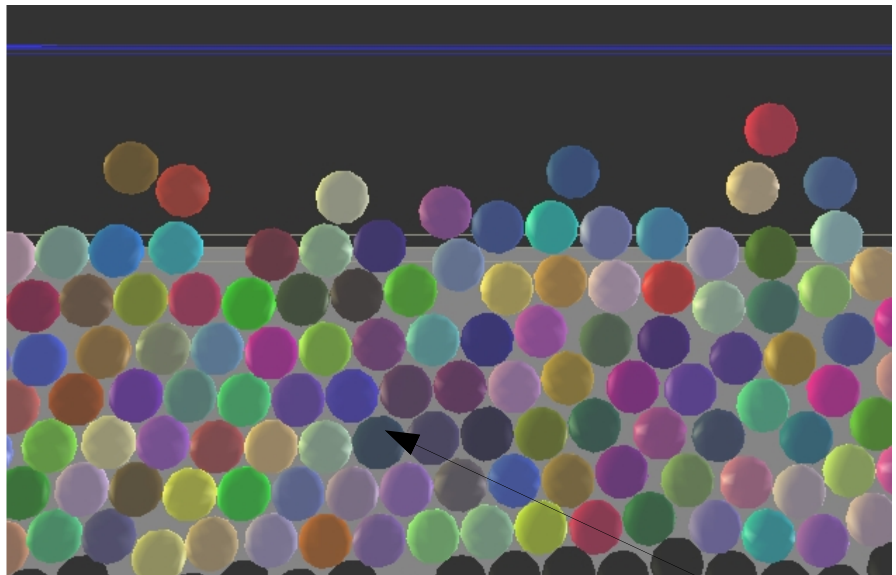
- Transport equilibrium
- **Particle tracking**

Experiment
P. Frey, *ESPL **39**:646-655 (2014)*



Experimental validation

No fitting parameters



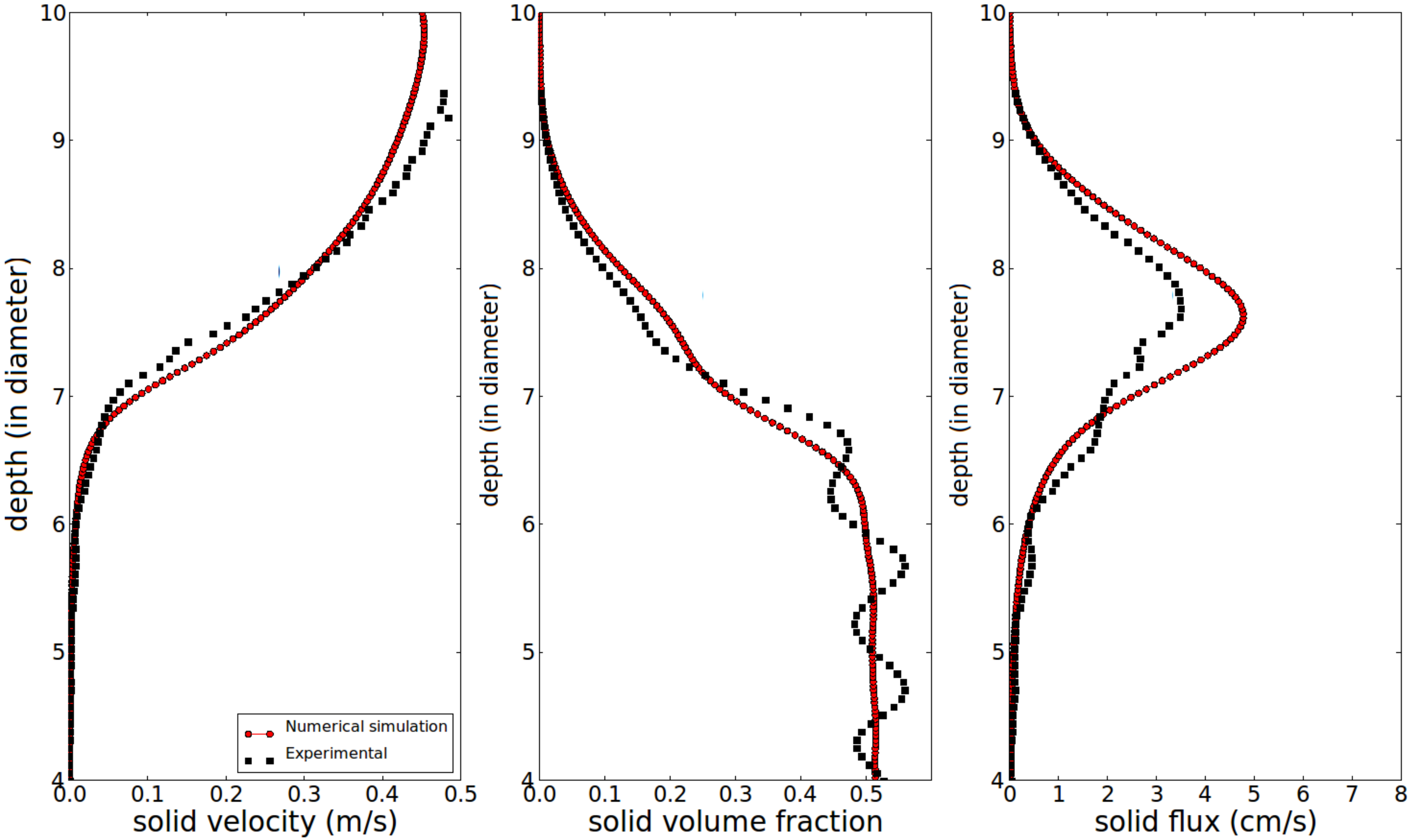
H : impose the free surface elevation (from exp.)

Periodic boundary condition

Impose the number of beads (from exp.)

- **Qualitative results**
  - Video : /home/raphael/Bureau/presentationMGMAS/Numd10tailleExp.ogv

Experimental validation



Conclusion

Conclusion

- Coupling Yade with simple turbulent fluid description, in the framework of sediment transport
- Model able to reproduce well experimental results

Perspectives

- Other experimental confirmation (different exp., regime)
- 3D analysis of the phenomenon
- Segregation

Pore Scale fluid models and coupling

Emanuele Catalano[1], Bruno Chareyre[2], Eric Barthélémy3

[1]Itasca France

[2]3SR, Grenoble INP, UJF, CNRS UMR 5521, CNRS, France

[3]LEGI, Grenoble INP, UJF, CNRS UMR 5519, CNRS, France

Yade Workshop 2014, Grenoble

Laboratoire
Sols Solides Structures Risques

Grenoble INP

Layout

1. DEM-PFV coupling
   - Pore Scale Finite Volumes
   - Incompressible Stokes Flow
   - Strong poromechanical coupling
   - Benchmark tests

Layout

1. DEM-PFV coulping
   - Pore Scale Finite Volumes
   - Incompressible Stokes Flow
   - Strong poromechanical coupling
   - Benchmark tests

Pore Scale Finite Volumes

A variety of methods are being developed to couple the DEM with fluid flow models. Two main groups of methods emerge (review paper: Zhu et al. (2007)):

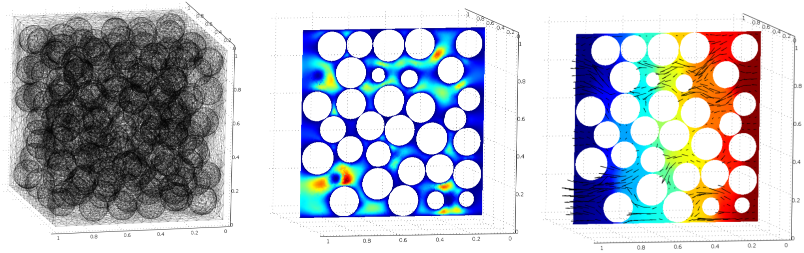
- Sub-particle scale for the fluid (DNS-DEM, LB-DEM, SPH-DEM,...)
- Continuum scale for the fluid (CFD-DEM)

The DEM-PFV coupling has a fluid scale of the order of the particles sizes, aiming at:

- A compromise in terms of computational cost vs. accuracy of per-particle forces
- An efficient integration scheme for strong poromechanical couplings

A closer look at how the fluid flows

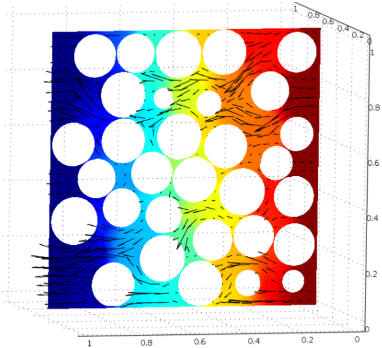
DEM-DNS simulation enables detailed study of fluid flow at the micro-scale (Chareyre et al. (2012)).



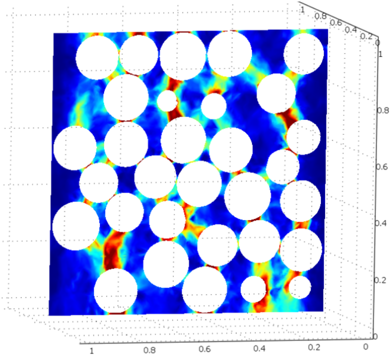
(a) Finite element discretisation (b) velocity (c) pressure

A closer look at how the fluid flows

The pressure drop along the flow path is highly localized.



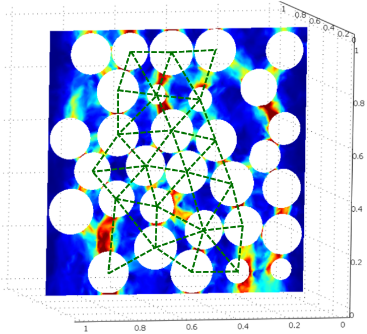
(d) Pressure field + velocity



(e) Pressure gradient

PFV: partitionning the pore space

A *pore* is that part of the void space enclosed in the cell of a triangulation, in which pressure is approximately constant.

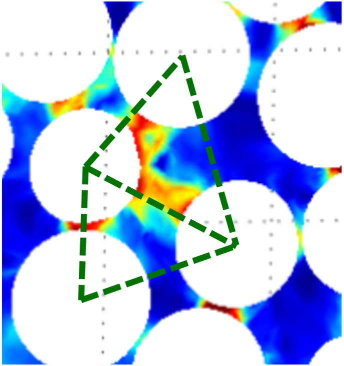


- Side note: it is of the upmost importance to employ a suitable type of triangulation. Delaunay triangulation would be irrelevant for polydispersed packings. *regular triangulation* (Pion and Teillaud, 2006) is a solution.

Incompressible Stokes Flow

Governing equations & num. scheme

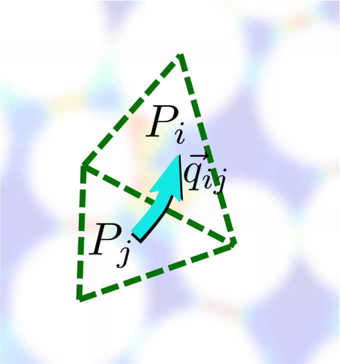
- Stokes flow:
  $\int_{f_{ij}} \vec{u}_w^* \cdot \vec{n} ds = q_{ij}^* = k_{ij}(P_j - P_i)$
  (u_w^*: relative velocity)
- Continuity:
  $\int_{\partial\Omega} \vec{u}_w \cdot \vec{n} ds = 0$ (incompressible)
  or: $\int_{\partial\Omega} (\vec{u}_w^* + \vec{u}_s) \cdot \vec{n} dS = 0$
  linking fluid velocity and deformation rate:
  $\int_{\partial\Omega} \vec{u}_w^* \cdot \vec{n} ds = \dot{V}_i$
- implicit dependency of P on particles velocity:
  $\sum_{j=1}^{4} k_{ij}(P_j - P_i) = \dot{V}_i$
- P solution of the linear system:
  $\mathbf{K}\,\mathbf{P} = \mathbf{E}\dot{\mathbf{X}} + \mathbf{Q}_{BC}$
- Forces on the particles function of P:
  $\mathbf{F}_w = \mathbf{S}\mathbf{K}^{-1}(\mathbf{E}\dot{\mathbf{X}} + \mathbf{Q}_{BC})$



Incompressible Stokes Flow

Governing equations & num. scheme

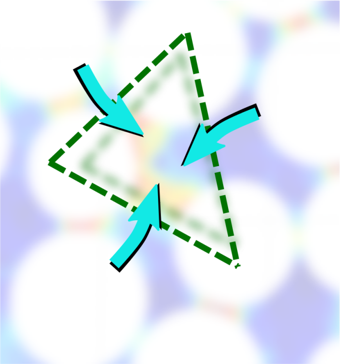
- Stokes flow:
  $\int_{f_{ij}} \vec{u}_w^* \cdot \vec{n} ds = q_{ij}^* = k_{ij}(P_j - P_i)$
  (u_w^*: relative velocity)
- Continuity:
  $\int_{\partial\Omega} \vec{u}_w \cdot \vec{n} ds = 0$ (incompressible)
  or: $\int_{\partial\Omega} (\vec{u}_w^* + \vec{u}_s) \cdot \vec{n} dS = 0$
  linking fluid velocity and deformation rate:
  $\int_{\partial\Omega} \vec{u}_w^* \cdot \vec{n} ds = \dot{V}_i$
- implicit dependency of P on particles velocity:
  $\sum_{j=1}^{4} k_{ij}(P_j - P_i) = \dot{V}_i$
- P solution of the linear system:
  $\mathbf{K}\,\mathbf{P} = \mathbf{E}\dot{\mathbf{X}} + \mathbf{Q}_{BC}$
- Forces on the particles function of P:
  $\mathbf{F}_w = \mathbf{S}\mathbf{K}^{-1}(\mathbf{E}\dot{\mathbf{X}} + \mathbf{Q}_{BC})$



Incompressible Stokes Flow

Governing equations & num. scheme

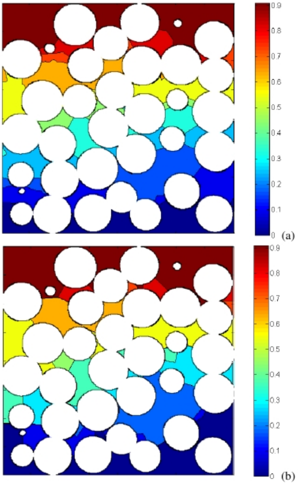
- Stokes flow:
  $\int_{f_{ij}} \vec{u}_w^* \cdot \vec{n} ds = q_{ij}^* = k_{ij}(P_j - P_i)$
  (u_w^*: relative velocity)
- Continuity:
  $\int_{\partial\Omega} \vec{u}_w \cdot \vec{n} ds = 0$ (incompressible)
  or: $\int_{\partial\Omega} (\vec{u}_w^* + \vec{u}_s) \cdot \vec{n} dS = 0$
  linking fluid velocity and deformation rate:
  $\int_{\partial\Omega} \vec{u}_w^* \cdot \vec{n} ds = \dot{V}_i$
- implicit dependency of P on particles velocity:
  $\sum_{j=1}^{4} k_{ij}(P_j - P_i) = \dot{V}_i$
- P solution of the linear system:
  $\mathbf{K}\,\mathbf{P} = \mathbf{E}\dot{\mathbf{X}} + \mathbf{Q}_{BC}$
- Forces on the particles function of P:
  $\mathbf{F}_w = \mathbf{S}\mathbf{K}^{-1}(\mathbf{E}\dot{\mathbf{X}} + \mathbf{Q}_{BC})$



Incompressible Stokes Flow

Governing equations & num. scheme

- Stokes flow:
  $\int_{f_{ij}} \vec{u}^*_w \cdot \vec{n} ds = q^*_{ij} = k_{ij}(P_j - P_i)$
  (u^*_w: relative velocity)
- Continuity:
  $\int_{\partial\Omega} \vec{u}_w \cdot \vec{n} ds = 0$ (incompressible)
  or: $\int_{\partial\Omega} (\vec{u}^*_w + \vec{u}_s) \cdot \vec{n} dS = 0$
  linking fluid velocity and deformation rate:
  $\int_{\partial\Omega} \vec{u}^*_w \cdot \vec{n} ds = \dot{V}_i$
- implicit dependency of P on particles velocity:
  $\sum_{j=1}^{4} k_{ij}(P_j - P_i) = \dot{V}_i$
- P solution of the linear system:
  $\mathbf{K}\,\mathbf{P} = \mathbf{E}\dot{\mathbf{X}} + \mathbf{Q}_{BC}$
- Forces on the particles function of P:
  $\mathbf{F}_w = \mathbf{S}\mathbf{K}^{-1}(\mathbf{E}\dot{\mathbf{X}} + \mathbf{Q}_{BC})$

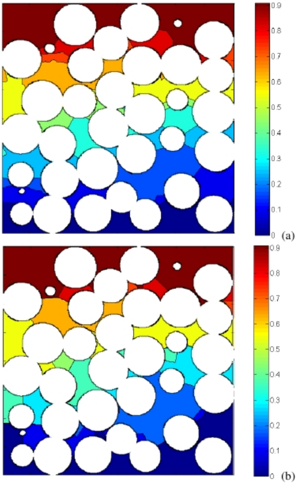


Solution DNS (a) and PFV (b)

Incompressible Stokes Flow

Governing equations & num. scheme

- Stokes flow:
  $\int_{f_{ij}} \vec{u}_w^* \cdot \vec{n} ds = q_{ij}^* = k_{ij}(P_j - P_i)$
  (u_w^*: relative velocity)
- Continuity:
  $\int_{\partial\Omega} \vec{u}_w \cdot \vec{n} ds = 0$ (incompressible)
  or: $\int_{\partial\Omega} (\vec{u}_w^* + \vec{u}_s) \cdot \vec{n} dS = 0$
  linking fluid velocity and deformation rate:
  $\int_{\partial\Omega} \vec{u}_w^* \cdot \vec{n} ds = \dot{V}_i$
- implicit dependency of P on particles velocity:
  $\sum_{j=1}^{4} k_{ij}(P_j - P_i) = \dot{V}_i$
- P solution of the linear system:
  $\mathbf{K}\,\mathbf{P} = \mathbf{E}\dot{\mathbf{X}} + \mathbf{Q}_{BC}$
- Forces on the particles function of P:
  $\mathbf{F}_w = \mathbf{S}\mathbf{K}^{-1}(\mathbf{E}\dot{\mathbf{X}} + \mathbf{Q}_{BC})$



Solution DNS (a) and PFV (b)

Strong poromechanical coupling

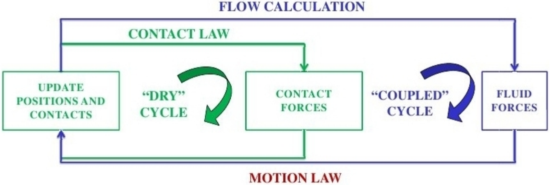
- A discrete analog of the equations of continuum (Biot's) poromechanics. See Catalano et al. (2013).

$$\mathbf{M}\ddot{\mathbf{X}} = \mathbf{F}_c + \mathbf{W} + \mathbf{S}\mathbf{K}^{-1}(\mathbf{E}\dot{\mathbf{X}} + \mathbf{Q}_{BC}) \qquad (1)$$

- The coupling term leads to instantaneous long range interactions between the particles (special care needed for choosing the numerical scheme).
- Semi-implicit scheme implemented in Yade-DEM (Smilauer et al., 2010) (and freely available at http://yade-dem.org).

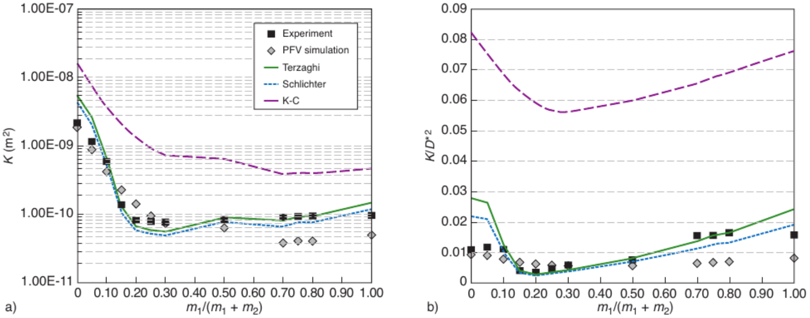
YADE

OPEN SOURCE
DISCRETE MODELLING



Benchmark tests

Permeability predictions:
Experiments on mixtures of two-sized glass beads compared to PFV and empirical/semi-empirical relations (Tong et al., 2012).



Benchmark tests

Consolidation problem:
Time evolution of a saturated medium under external load

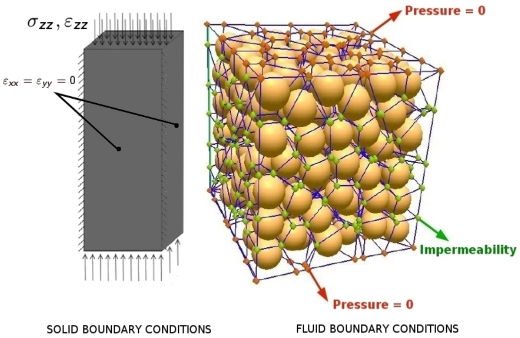
Terzaghi's theory of consolidation

Coefficient of consolidation:

$$C_v = \frac{kE_{oed}}{\gamma} \quad (2)$$

Consolidation time:

$$T_v = \frac{C_v t}{H^2} \quad (3)$$



Benchmark tests

Consolidation problem:
Time evolution of a saturated medium under external load

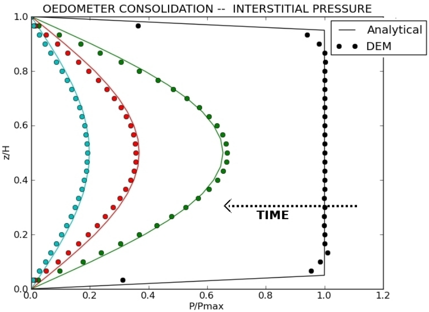
Terzaghi's theory of consolidation

Coefficient of consolidation:

$$C_v = \frac{kE_{oed}}{\gamma} \quad (2)$$

Consolidation time:

$$T_v = \frac{C_v t}{H^2} \quad (3)$$



Bibliography

H. Zhu, Z. Zhou, R. Yang, and A. Yu, "Discrete particle simulation of particulate systems: theoretical developments," *Chemical Engineering Science*, vol. 62, no. 13, pp. 3378–3396, 2007.

B. Chareyre, A. Cortis, E. Catalano, and E. Barthélemy, "Pore-Scale Modeling of Viscous Flow and Induced Forces in Dense Sphere Packings," *Transport in Porous Media*, vol. 92, no. 2, pp. 473–493, 2012.

S. Pion and M. Teillaud, "3D triangulations," *CGAL Editorial Board, editor, CGAL-3.2 User and Reference Manual*, 2006.

E. Catalano, B. Chareyre, and E. Barthélémy, "Pore-scale modeling of fluid-particles interaction and emerging poromechanical effects," *International Journal for Numerical and Analytical Methods in Geomechanics*, 2013, DOI:10.1002/nag.2198.

V. Smilauer, E. Catalano, B. Chareyre, S. Dorofenko, J. Duriez, A. Gladky, J. Kozicki, C. Modenese, L. Scholtès, L. Sibille, J. Stransky, and K. Thoeni, "Yade Reference Documentation," in *Yade Documentation*, 1st ed., V. Smilauer, Ed. The Yade Project, 2010, http://yade-dem.org/doc/.

A.-T. Tong, E. Catalano, and B. Chareyre, "Pore-Scale Flow Simulations: Model Predictions Compared with Experiments on Bi-Dispersed Granular Assemblies," *Oil & Gas Science and Technology-Revue d'IFP Energies Nouvelles*, vol. 67, no. 5, pp. 743–752, 2012.

YADE and rheology of dense suspensions

Donia Marzougui [1], Bruno Chareyre [1] and Julien Chauchat [2]

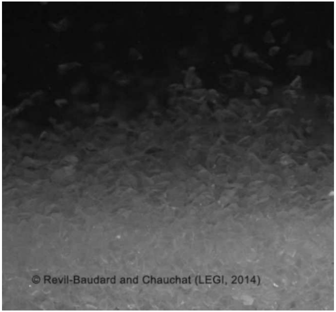
[1]3S-R

[2]LEGI

08 - 07 - 2014

Motivations

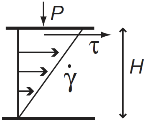
SheetFlow



Immersed granular material:

solid-solid interactions and solid-fluid interactions.

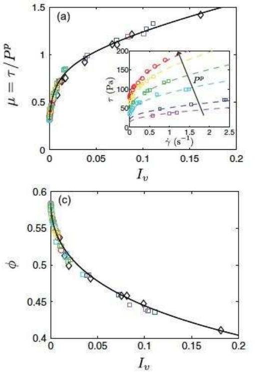
Rheology of the material: simple shear test.



What are these hydrodynamic interactions?

How they can play an important role to the rheology of the suspension?

Motivations



Empirical laws from the experiments of Boyer et al.

Boyer et al, Physical Review Letters (2011)

Immersed granular material:

solid-solid interactions and solid-fluid interactions.

Rheology of the material: simple shear test.

viscous ratio $\mu = \sigma_{xy}/P$ and volume fraction Φ vs viscous number I_v:

$$I_v = \frac{\eta \dot{\gamma}}{P}$$

Plan

Plan

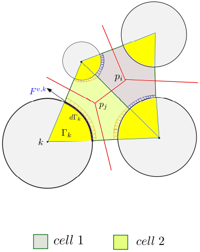
1. DEM-fluid model including lubrication forces
2. Rheology of dense suspensions
3. Conclusions

hydrodynamic interactions

Long-range interactions

DEM-PFV

(Catalano et al (2012))



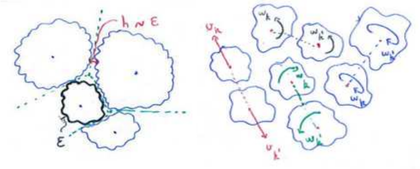
$$F^k = + \int_{\partial\Gamma_k} p \ \mathbf{n} ds + \int_{\partial\Gamma_k} \tau \ \mathbf{n} ds$$

$$F^k = \underbrace{F^{p,k}}_{pressure} + \underbrace{F^{v,k}}_{viscous\ stress}$$

Short-range interactions

Lubrication theory

(Jeffrey & Onishi (1984), Frankel & Acrivos (1967))



$$F_k^n = -F_{k'}^n = \frac{3}{2} \pi \eta \frac{a^2}{h} \mathbf{v}_n$$

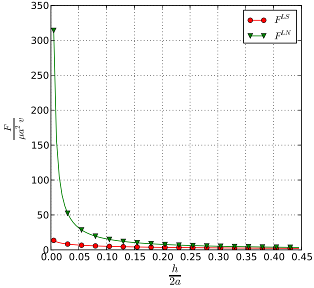
$$F_k^s = -F_{k'}^s = \frac{\pi\eta}{2} (-2a + (2a+h) \ ln(\frac{2a+h}{h})) \ \mathbf{v}_t$$

$$C_{k,k'}^s = (a_{k,k'} + \frac{h}{2}) \ F_{k,k'}^s$$

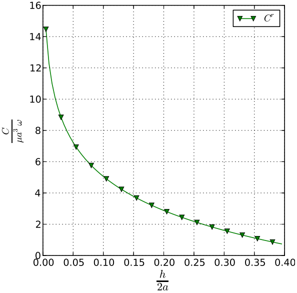
$$C_k^r = - C_{k'}^r = \pi \ \eta \ a^3 \ f^r(\frac{h}{a}) \ (\omega_k - \omega_{k'}) \cdot \mathbf{t} \ \mathbf{t}$$

$$C_k^t = - C_{k'}^t = \pi \ \eta \ a^2 \ f^t(\frac{h}{a}) \ (\omega_k - \omega_{k'}) \cdot \mathbf{n} \ \mathbf{n}$$

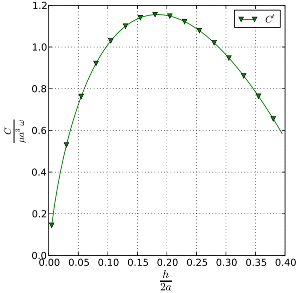
hydrodynamic interactions



Normal and shear lubrication force vs h



Rolling torque vs h



Twist torque vs h

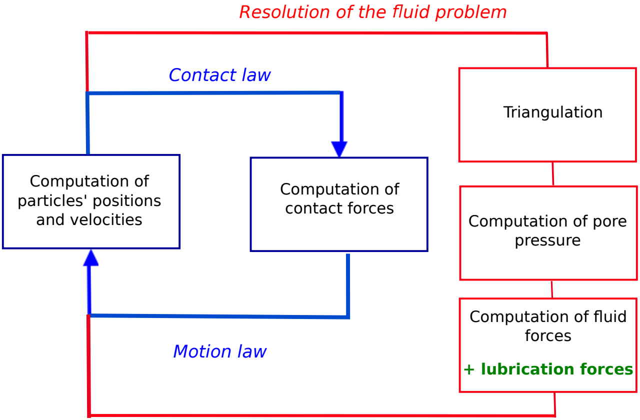
→ Divergence for very close particles.

→ Zero for very far particles.

→ Except twist torque: zero for two particles in contact

Jeffrey & Onishi (1984), Frankel & Acrivos (1967)

Computing cycle



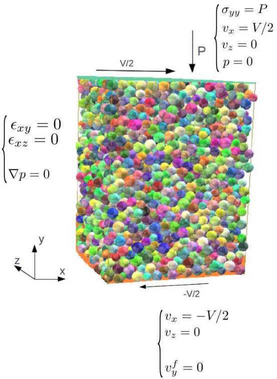
Plan

1. DEM-fluid model including lubrication forces
2. Rheology of dense suspensions
3. Conclusions

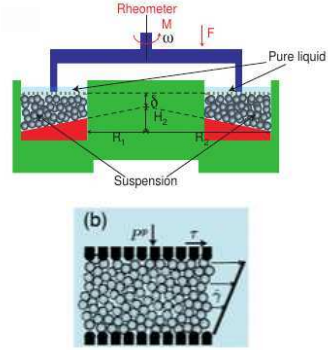
Configuration

Numerical configuration

shearFlow



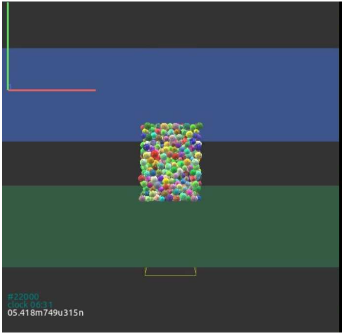
Experimental configuration of Boyer et al



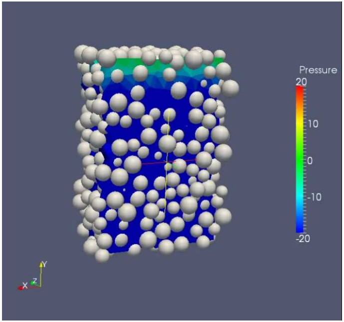
Boyer et al, Physical Review Letters (2011)

Configuration

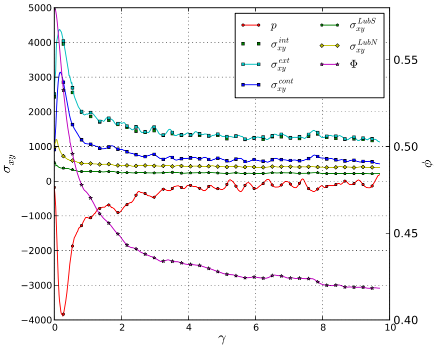
shearFlow



Pressure



Shear stress



- $\boldsymbol{\tau} = \mathbf{F} / S$
- $\boldsymbol{\sigma} = \boldsymbol{\sigma}^C + \boldsymbol{\sigma}^{LN} + \boldsymbol{\sigma}^{LS} + p\mathbf{I} + \boldsymbol{\sigma}^I$

  Contact stress:

  $$\boldsymbol{\sigma}^C = \frac{1}{Vol} \sum_i \mathbf{F}_i^C \otimes \mathbf{l}_i$$

  Lubrication stress (normal + shear):

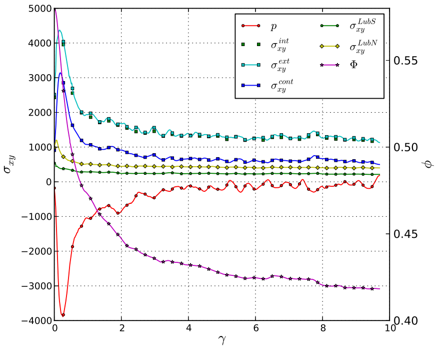
  $$\boldsymbol{\sigma}^L = \frac{1}{Vol} \sum_i \mathbf{F}_i^L \otimes \mathbf{l}_i$$

  Inertial stress:

  $$\boldsymbol{\sigma}^I = \sum_i m_i \mathbf{v}_i \otimes \mathbf{v}_i$$

- $\boldsymbol{\tau} = \boldsymbol{\sigma} \cdot \mathbf{y}$

Shear stress



For $I_v = 0.21$:

Inertial stress: $\boldsymbol{\sigma}^I < 2.5\,\%\,\boldsymbol{\sigma}$

Contact stress: $\boldsymbol{\sigma}^C \sim 50\,\%\,\boldsymbol{\sigma}$

Lubrication stress: $\boldsymbol{\sigma}^{LN} \sim 30\,\%\,\boldsymbol{\sigma}$

$\boldsymbol{\sigma}^{LS} \sim 20\,\%\boldsymbol{\sigma}$

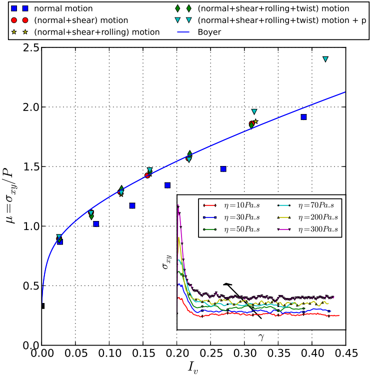
→ Non-inertial regime.

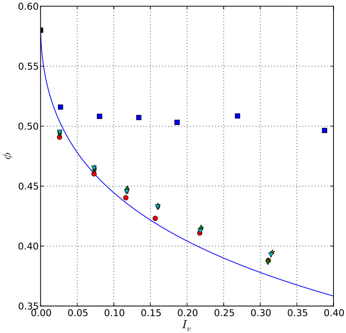
→ Contacts are predominant.

→ Normal lubrication > Shear lubrication.

→ Shear lubrication is not negligible to the normal one (contrary to what is sometimes postulated in the litterature !).

Rheology





σ^{LN} is not sufficient to have a satisfying behavior of the suspension.

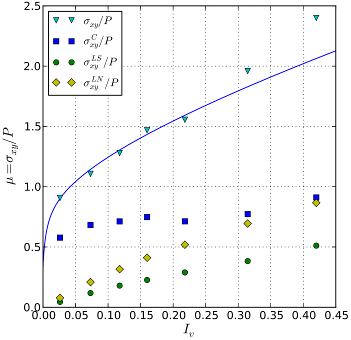
σ^{LS} contributes to σ and especially to the dilatancy.

C^r et C^t contribute to σ and especially to the dilatancy.

The poromechanical coupling doesn't contribute (in steady regime).

Rheology

$$\boldsymbol{\sigma}_{xy} = \boldsymbol{\sigma}_{xy}^{C} + \boldsymbol{\sigma}_{xy}^{LN} + \boldsymbol{\sigma}_{xy}^{LS}$$



Contacts are predominant.

σ^{LS} and σ^{LN} increase with I_v.

Plan

1. DEM-fluid model including lubrication forces
2. Rheology of dense suspensions
3. Conclusions

* The DEM-fluid coupled model including lubrication theory is a good candidate to describe the rheology of the suspension.

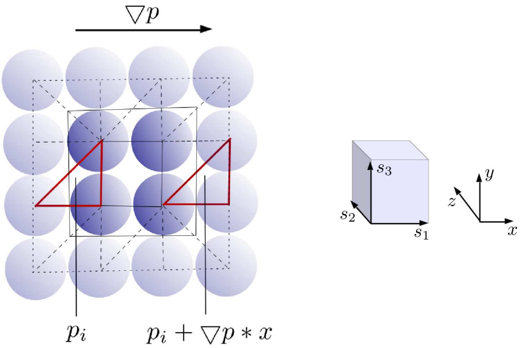
* The normal lubrication is certainly important but isn't enough to describe the behavior of the suspension.

* The shear is important to the rheology of the susupension and contributes wall to the dilatancy of the medium.

* The contribution of rolling and twist motions is very weak to the the rheology of the suspension.

* Thanks to YADE who was very useful to study such a problem.

Thanks for your attention



(a): 2D periodic cell (b): Simulation period of size $S = [s_1, s_2, s_3]$ *in the coordinate system* $[x, y, z]$

$$\underline{r}' = \underline{r} + \underline{\underline{S}} \cdot \underline{i}$$

$$p' = p + \nabla p \cdot S * i$$

$\underline{r}$: position of a point in the real volume

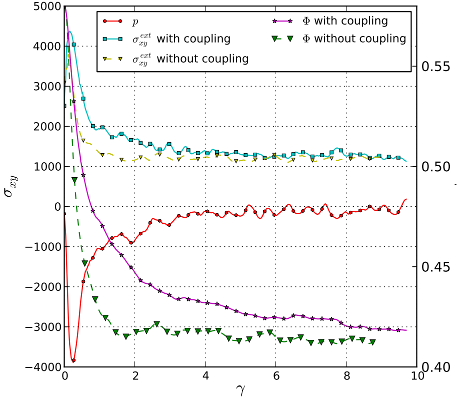
$\underline{r}'$: position of the image point.

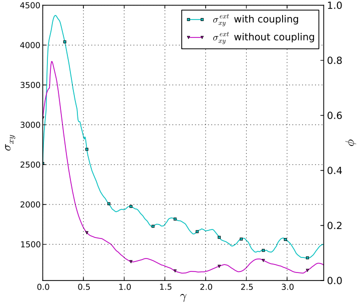
p: pressure of a point in the real volume.

p': pressure of the image point.

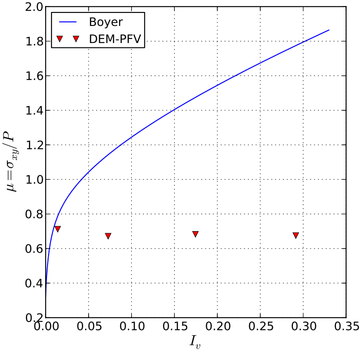
∇p: pressure gradient.

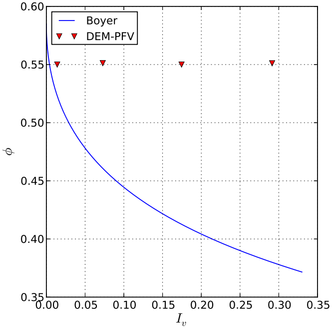
Shear stress





The poromechanical coupling has an effect on the transient regime and not on the steady regime.


Boyer
DEM-PFV
$\mu = \sigma_{xy}/P$
I_v
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35


Boyer
DEM-PFV
ϕ
I_v
0.60
0.55
0.50
0.45
0.40
0.35
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35

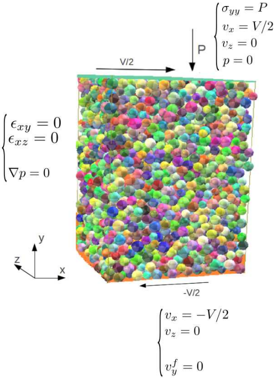
```
flow=PeriodicFlowEngine(
        bndCondIsPressure=[0,0,0,1,0,0],
        wallIds=[-1,-1,1,0,-1,-1],
        meshUpdateInterval=1000,
        gradP = (0,0,0),
        useSolver=3,
        viscosity =300,
        eps = 0.035,
        pressureForce=0,
        normalLubrication=0,
        shearLubrication=0,
        pumpTorque=0,
        twistTorque=0
        )
```

```
O.engines=[
        ForceResetter(),
        InsertionSortCollider([Bo1_Box_Aabb(),Bo1_Sphere_Aabb(),Bo1_Facet_Aabb(),Bo1_Wall_Aabb()],verletDist=-0.1,allowBiggerThanPeriod=True),
        InteractionLoop(
                [Ig2_Sphere_Sphere_ScGeom6D(),Ig2_Box_Sphere_ScGeom6D()],
                [Ip2_CohFrictMat_CohFrictMat_CohFrictPhys(label='ip2')],
                [Law2_ScGeom6D_CohFrictPhys_CohesionMoment()]
        ),
        flow,
        GlobalStiffnessTimeStepper(active=1,timeStepUpdateInterval=100,timestepSafetyCoefficient=0.6,
        defaultDt=utils.PWaveTimeStep(),label="ts"),
        PeriTriaxController(dynCell=True,mass=10,maxUnbalanced=0.002,relStressTol=5e-3,goal=(-P,-P,-
        P),stressMask=7,globUpdate=5,maxStrainRate=(10,10,10),doneHook='compactionDone()',label='triax'),
        newton
]
```

Configuration

Numerical configuration

shearFlow



Cell size $= (1.2 * 1.2 * 1.8)m^3$

$N = 1000$

$a = 0.05 \pm 0.01$

$\eta = 200 Pa.s$

$P = 750 Pa$

$V = 1.5 m/s$

$O.dt = 1e - 6$

Almost 1 day of calculation

Performance Benchmarking with YADE

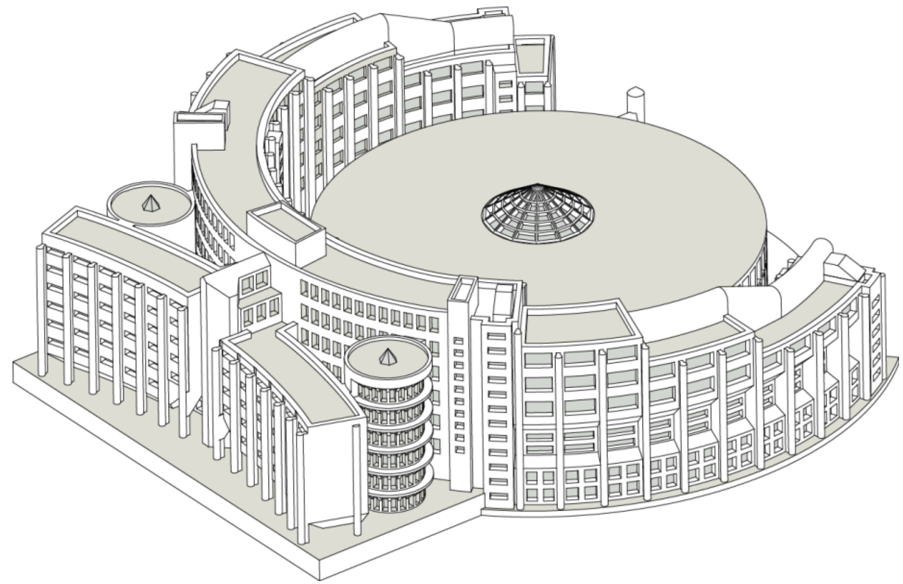
Production Technology Center Berlin

Prof. Dr. h. c. Dr.-Ing. Eckart Uhlmann

M. Sc. Alexander Eulitz

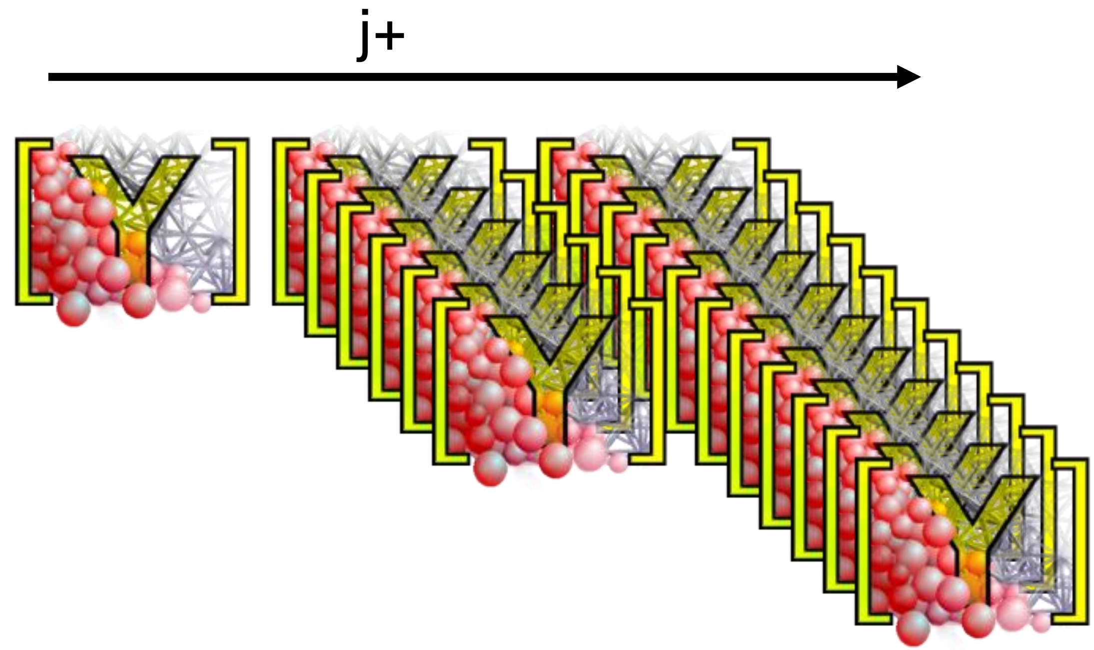
Institute for Maschine Tools and Factory Management
Technische Universität Berlin

Fraunhofer-Institute Production Systems and Design Technology



Agenda

- Motivation
- Performance Benchmark 2013
- Performance Benchmark 2014



Performance Benchmarking with YADE

Motivation

- How many threads should a simulation use?
- Which kind of hardware is the right? A lot of cores at low frequency (AMD) or some cores with higher frequency (Intel)?
- Does Yade benefit from Hyperthreading?
- How good is Yade parallized? -> Bruno's talk about the Parallel Collider
- Which architecture is better AMD or INTEL? -> Klaus

Performance Benchmarking with YADE

Hardware

- 2 x Intel Xeon E5-2687W @ 3.1GHz with 8 physical cores, cache 20 MB:
  - Without hyper-threading (woHT) → 16 cores
  - With hyper-threading (HT)→ 32 virtual cores
- 128 GB RAM
- IPMI access for Bios modification

Software

- Ubuntu 12.04 LTS 64 bit accessed by Windows Clients using "Remote Desktop Connection" in combination with Xrdp package at server
- Yade from source with slightly modified checkPerf.py
  - yade-2014-06-05: parallel collider
  - yade-2014-01-25: serial collider

Performance Benchmarking with YADE

Automated Performance Benchmarking – How To?

- Bash: sequentially invoking Yade instances with increasing number of threads. Saving terminal output.
- Yade: performanceScript.py (e.g. built-in performance test "–performance" which is equivalent to checkPerf.py)

Bash

```
BEGIN=$1
END=$2
[…]
echo "Beginning Yade Performance Test with Yade version " $VERSION
for (( I=$BEGIN; $I <= $END; I++ )); do
        DATE=$(date)
        echo $I" core(s); beginning: "$DATE
        ./$VERSION -j$I checkPerf.py>> Logs_${VERSION}/BuiltInPerf_${VERSION}_j$I.log
done [….]
```

Evaluate.py

- Extract performance data from single tests and save it

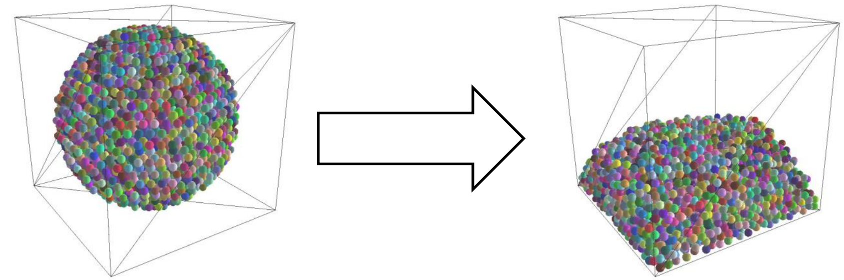
checkPerf.py

- test more particles (500k)
- more repetitions numberTests = 5
- sphere count corrected

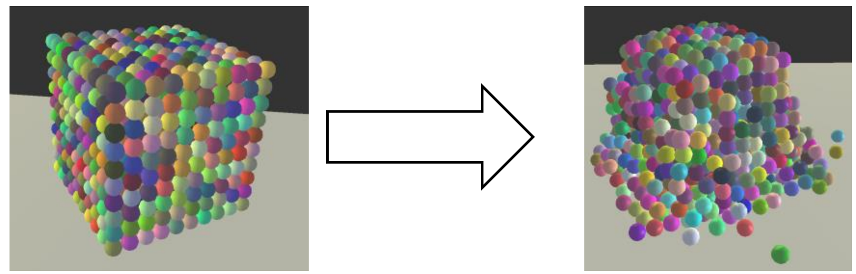
Performance Benchmarking with YADE

Performance Benchmarks

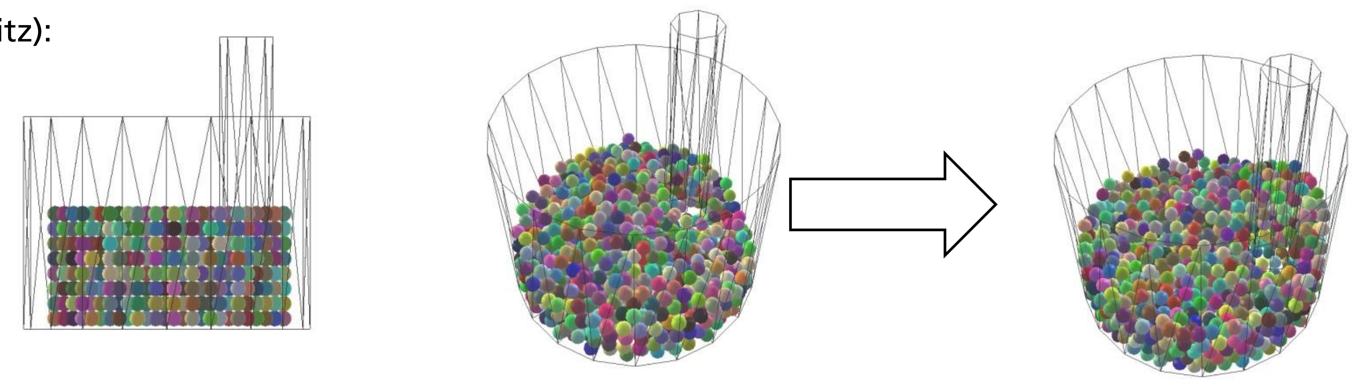
Built-in Performance Test (Anton Gladky, Klaus Thoeni):



Boxpacking on inclined plane (Christian Jakob):



Non-Vibrating Tub (Alexander Eulitz):



Performance Benchmarking with YADE

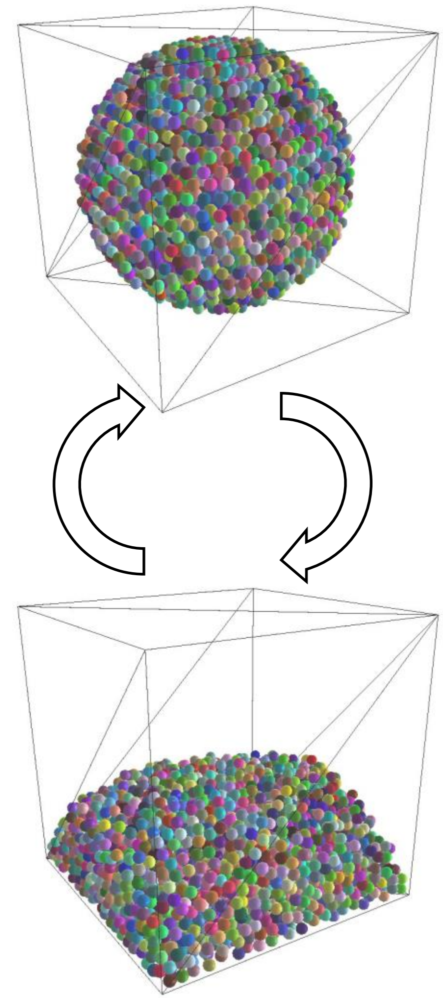
Built-In Performance Benchmark

- Spherical packing of spheres inside a box is exposed to gravitational deposition
- Repeated test (3, 5, 20 repetitions)
- Dimensions of the packing, i.e. number of particles and dimensions of the box change:

Number of particles	Iterations	coefficient
5,025	12,000	110
25,091	2,500	28
502,38	1,400	18
100,455	800	9
200,801	200	2
501,994	10	0.1

→ Changing number of iterations, is it comarable?
Given a constant time step and constant radius of spheres shouldn‘t number of iterations be constant?

- Test was not created for benchmarking but for regression tests after crucial commits

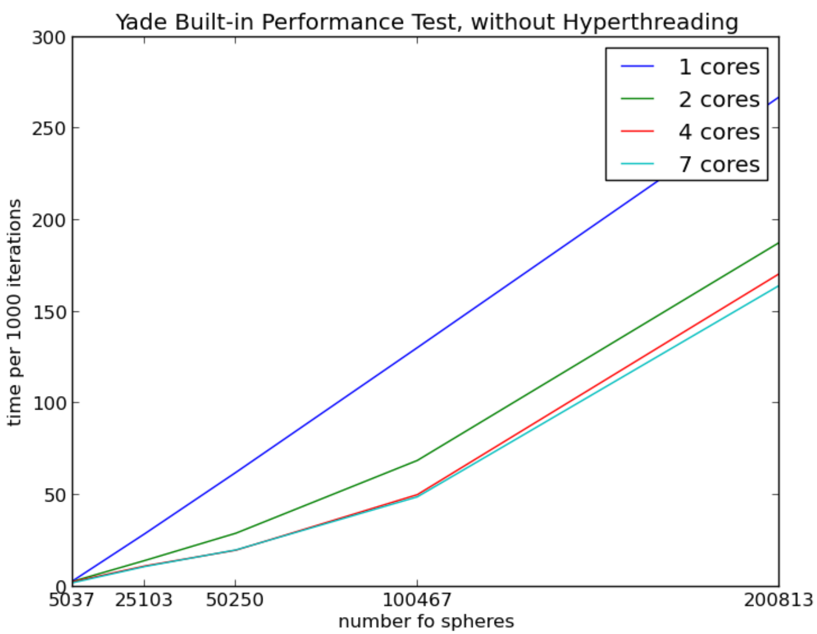


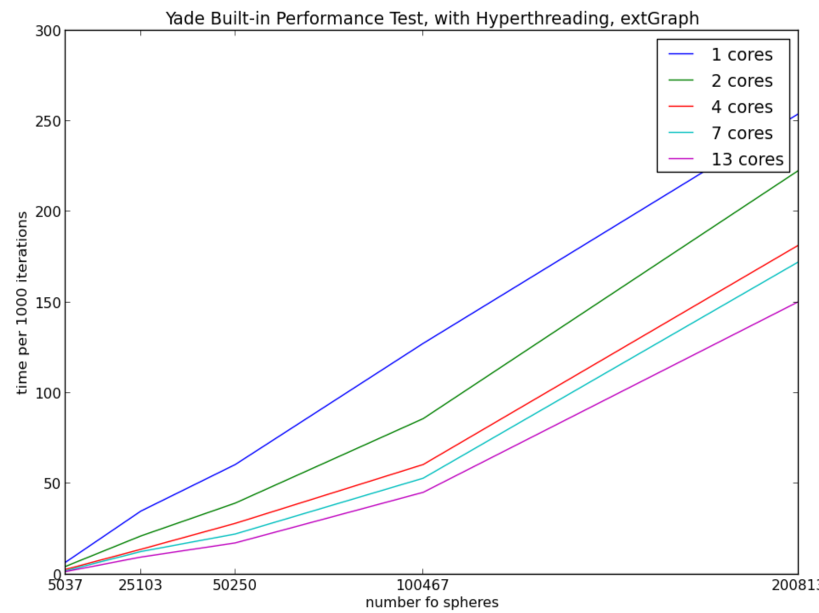
Model setup of built-in performance test

Performance Benchmarking with YADE

Performance Benchmarking 2013 - Results

- Built-in Performance test (--performance; see checkPerf.py in trunk/examples/test/performance)

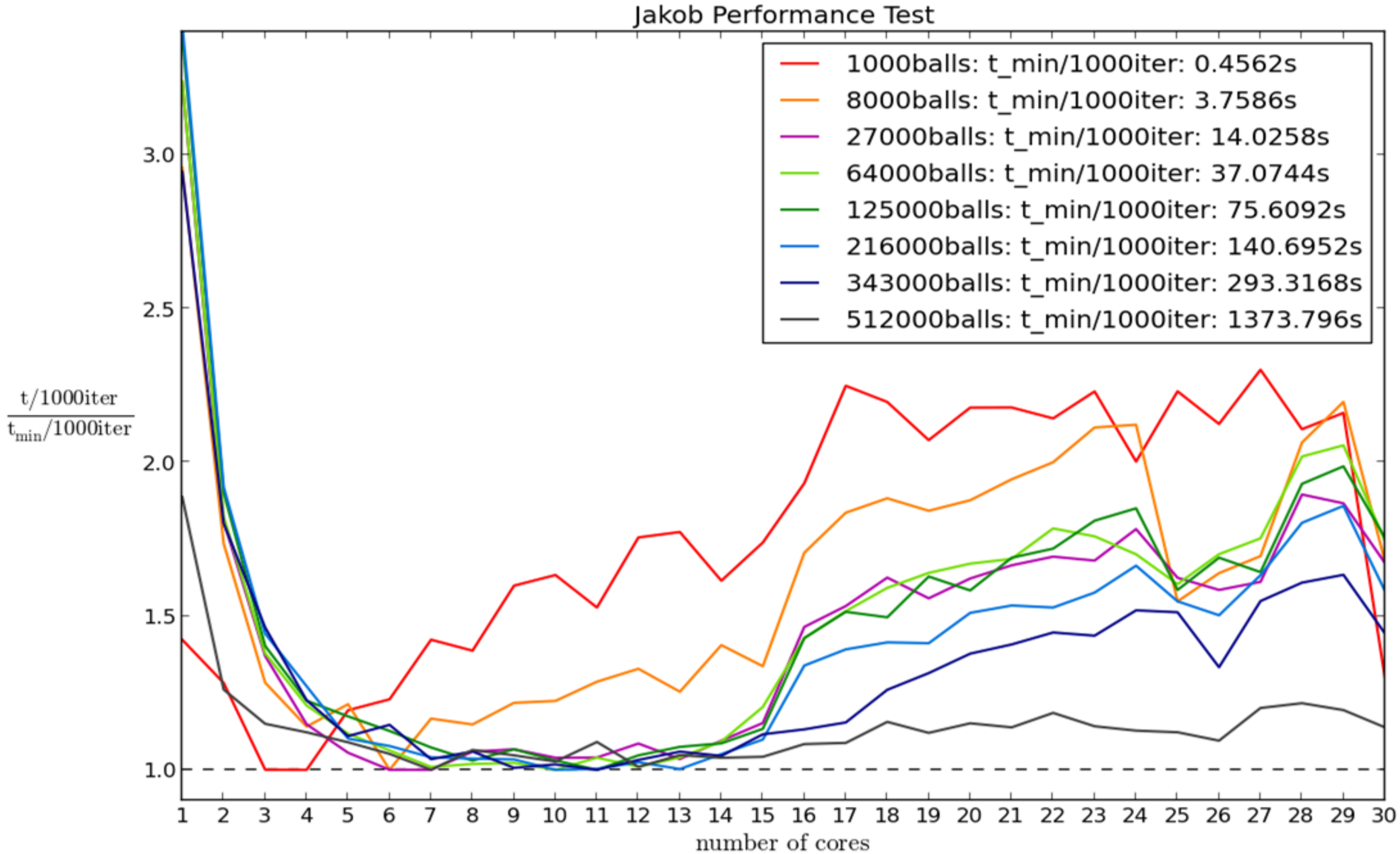




Performance Benchmarking with YADE

Performance Benchmarking 2013 - Results

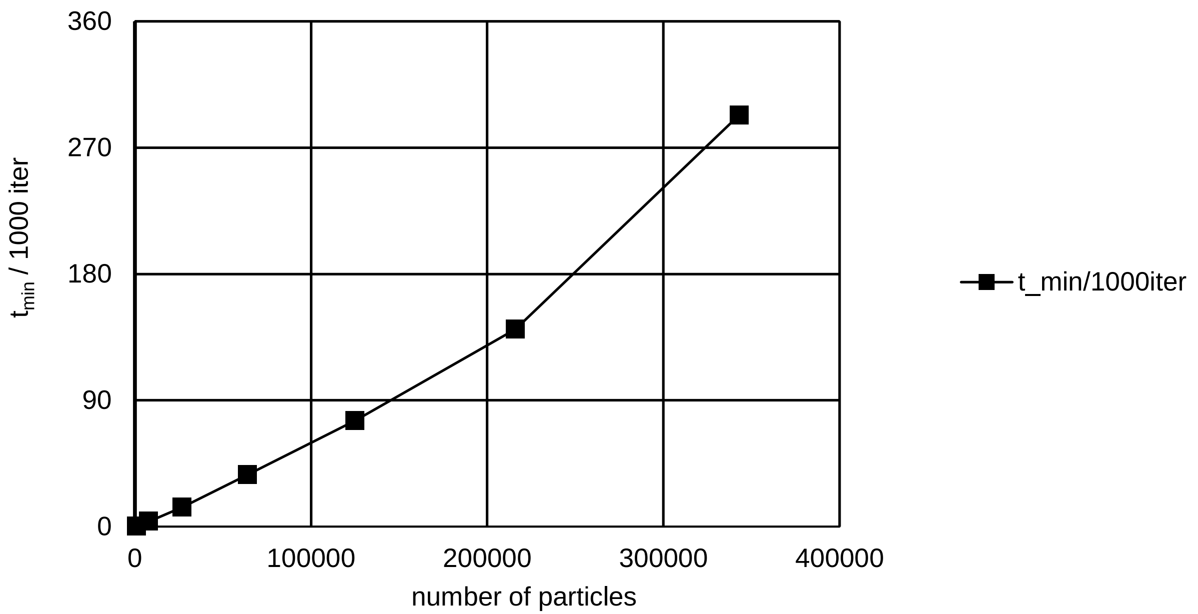
- Yade<->PFC3D



Performance Benchmarking with YADE

Performance Benchmarking 2013 - Results

- Yade<->PFC3D (Christian Jakob, see wikipage)



Performance Benchmarking with YADE

Performance Benchmarking 2013 - Results

- Yade<->PFC3D (Christian Jakob, see wikipage)

Performance Benchmarking with YADE

Performance Benchmarking 2013 - Conclusions

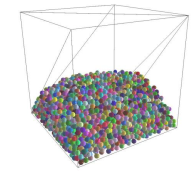
- Because of OpenMP implementation Yade should benefit from an increasing number of CPU cores
- Surprisingly shorter simulation times are only achieved for a rather small number of cores (depending on simulation setup between 4 and 7).
- Possible explanation:
  - increasing communication and synchronization effort
  - OpenMP is not implemented for all engines

Performance Benchmarking with YADE

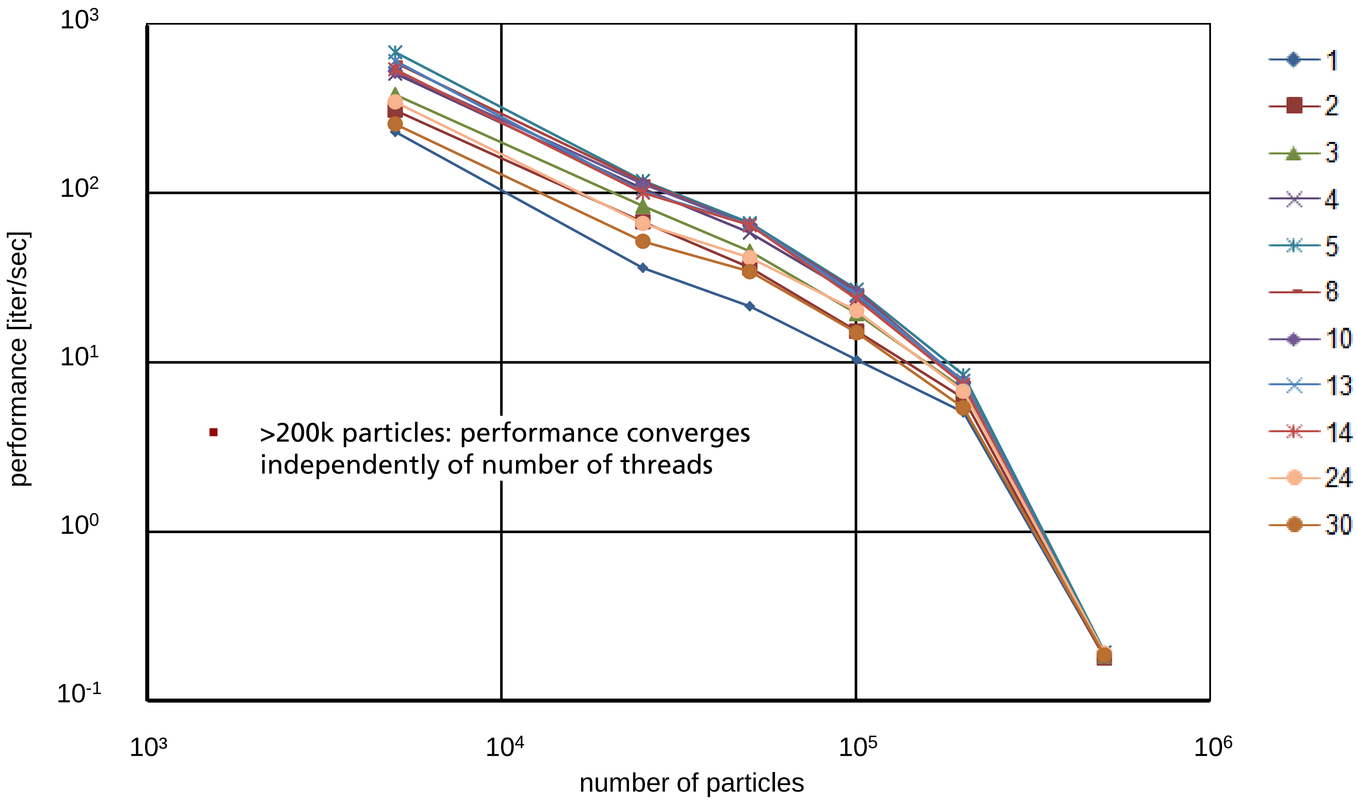
Performance Benchmarking 2014

- Major change in implementation of insertion sort collider
- Meanwhile openBlas broke multicore operations for Ubuntu 12.04 (there is a fix on that: https://bugs.launchpad.net/yade/+bug/1304878)

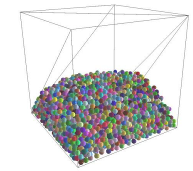
Performance Benchmarking with YADE



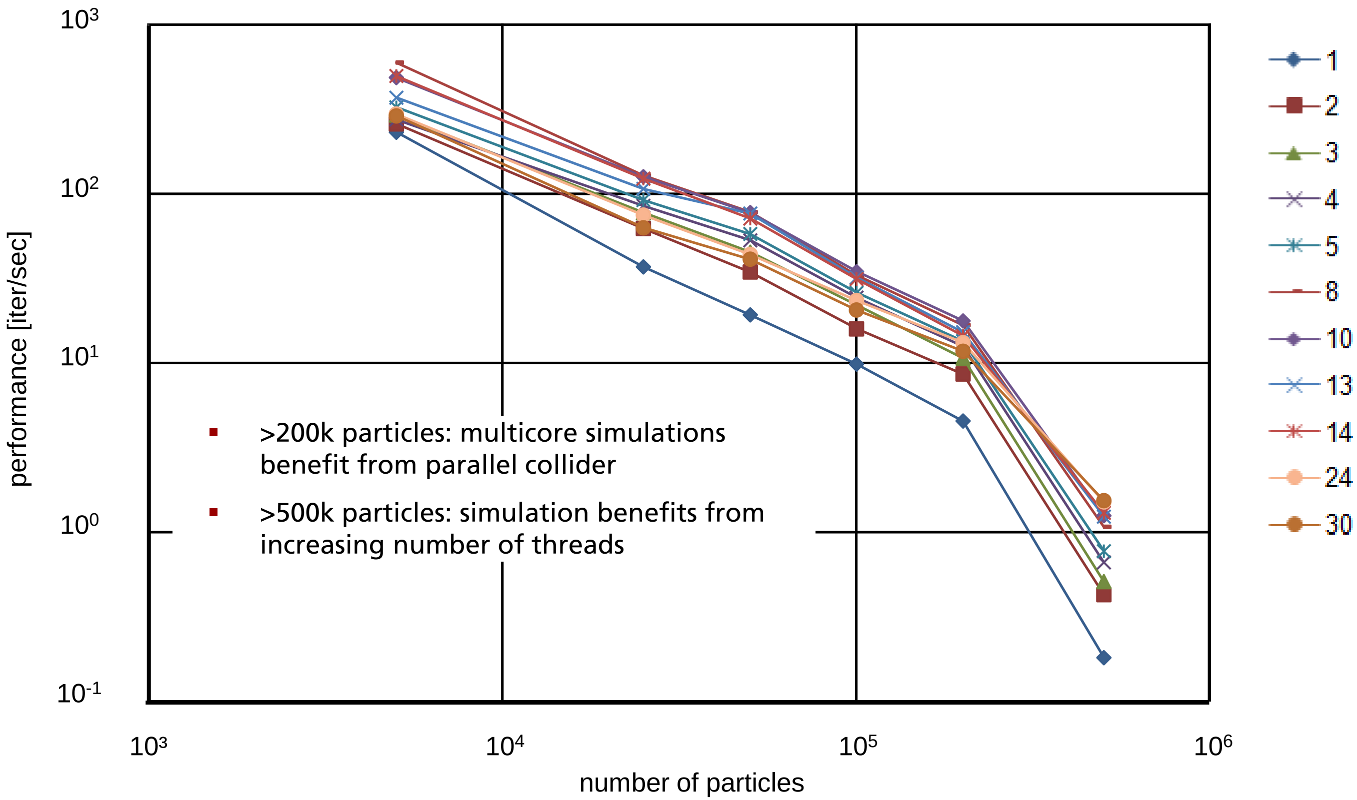
Performance Benchmark 2014 – Hyperthreading, Serial (old) Collider yade-2014-01-25



Performance Benchmarking with YADE

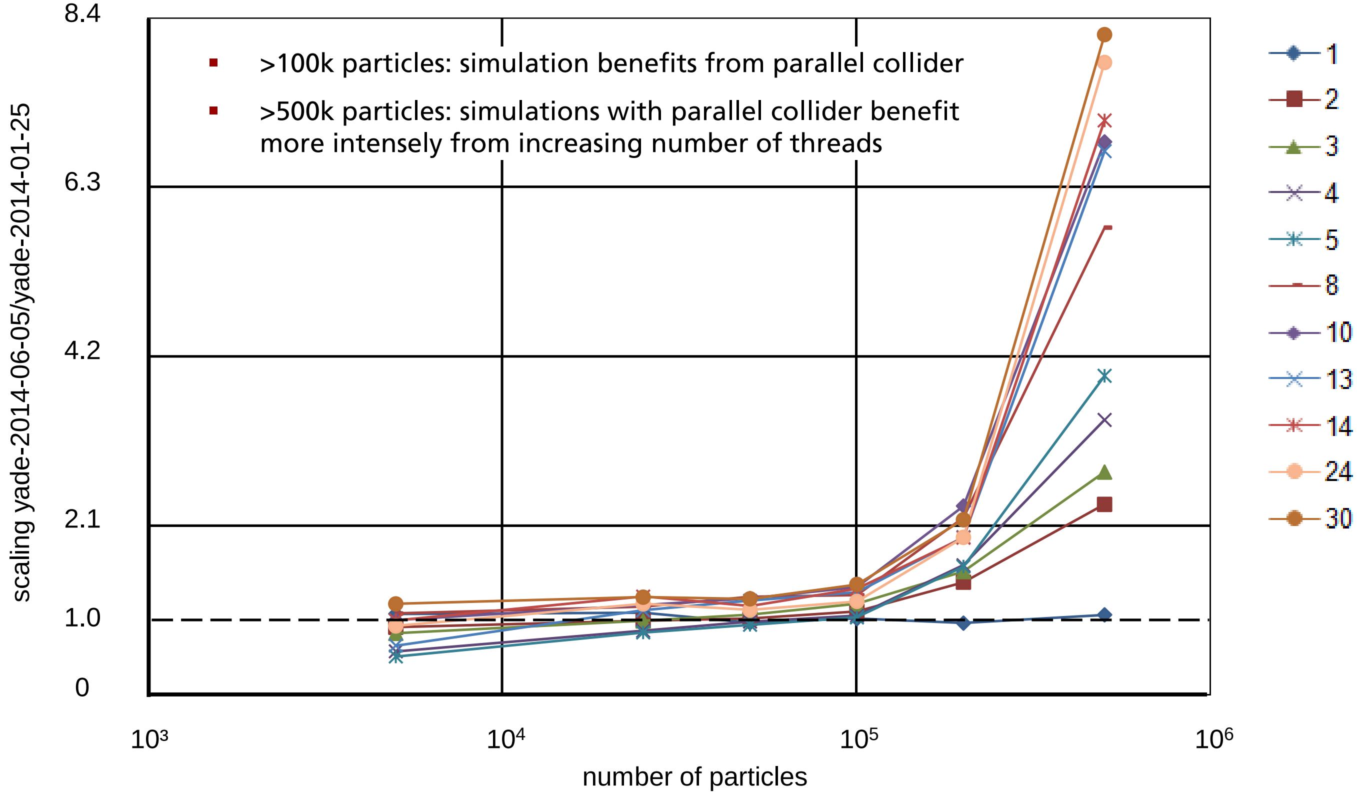


Performance Benchmark 2014 – Hyperthreading, Parallel Collider yade-2014-06-05



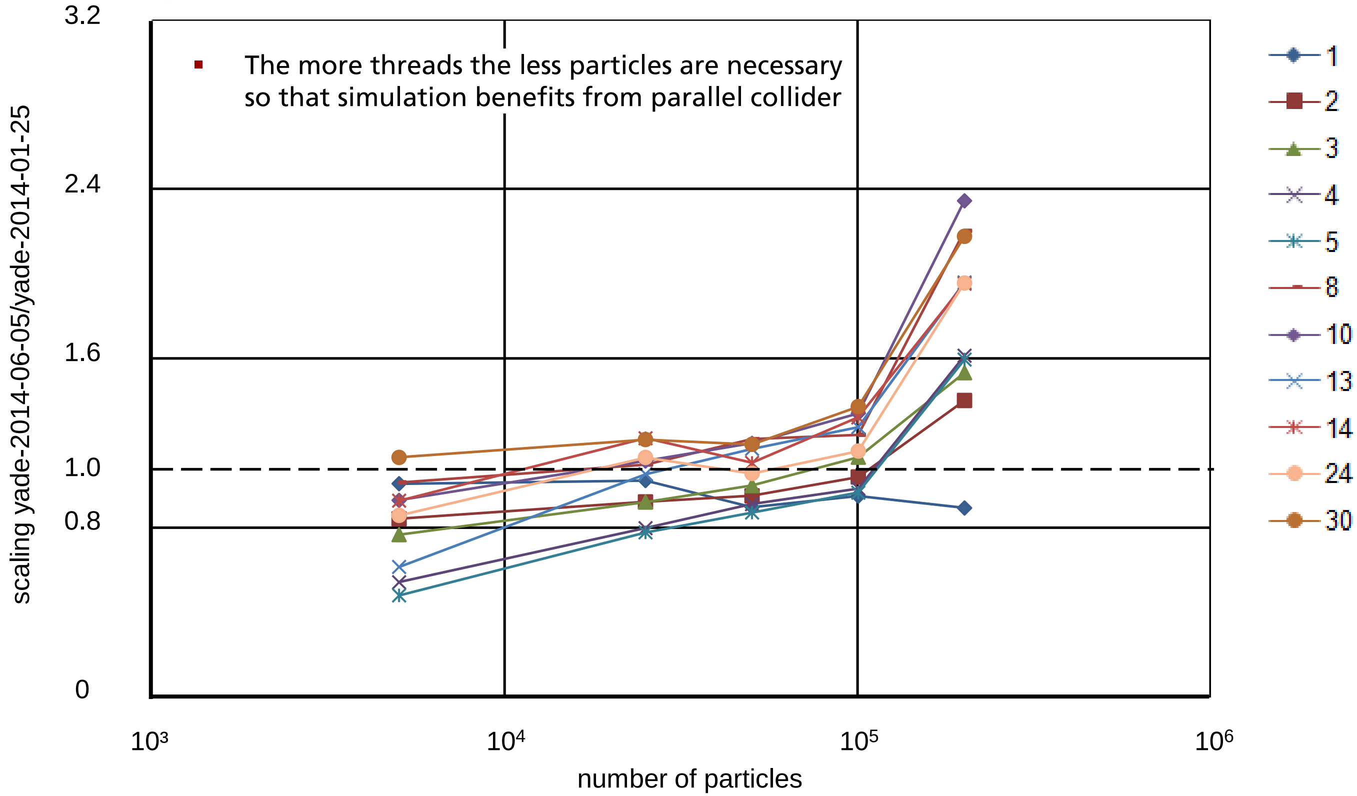
Performance Benchmarking with YADE

Performance Benchmark 2014 – Hyperthreading, Scaling of the Parallel Collider yade-2014-06-05 VS. yade-2014-01-25

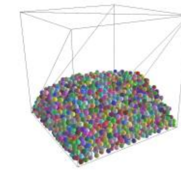


Performance Benchmarking with YADE

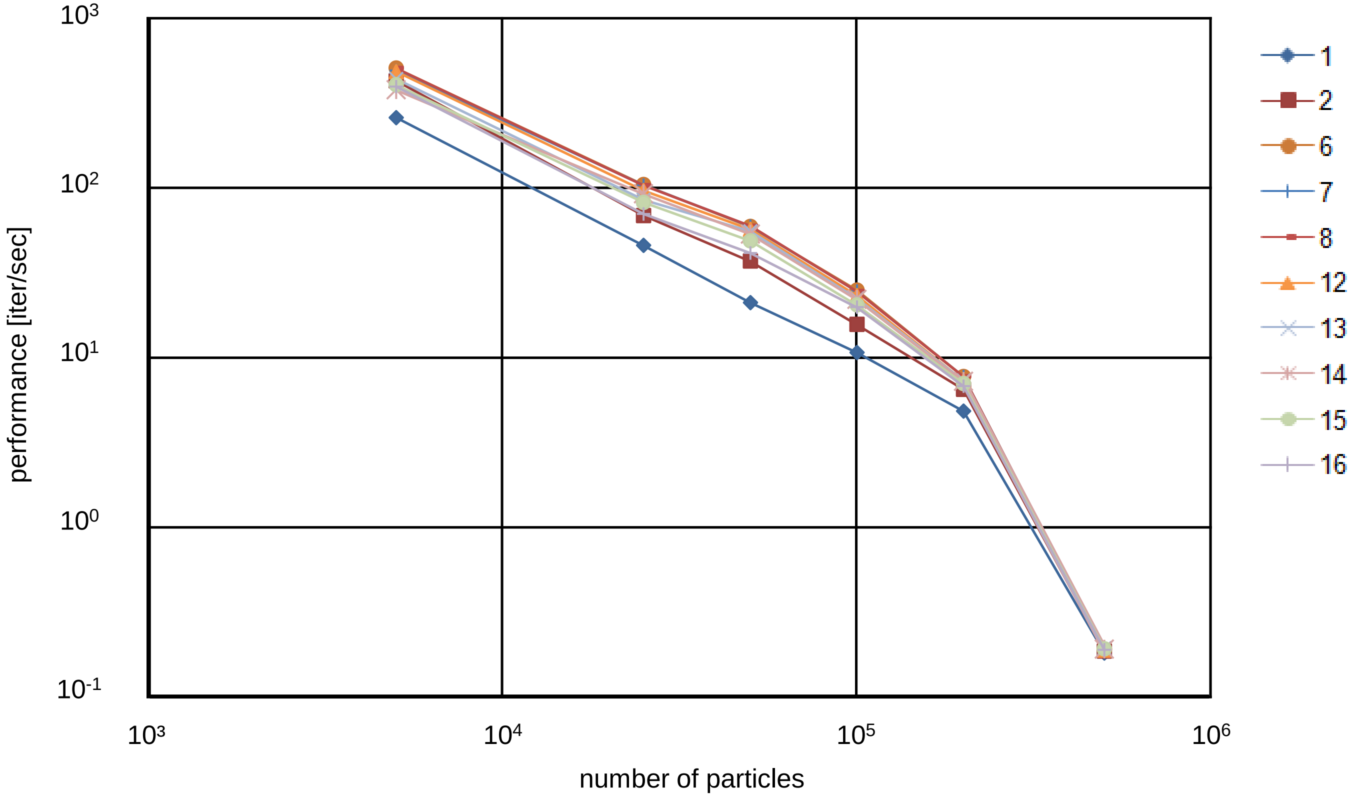
Performance Benchmark 2014 – Hyperthreading, Scaling of the Parallel Collider (detail) yade-2014-06-05 VS. yade-2014-01-25



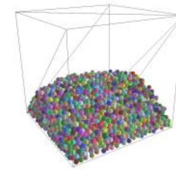
Performance Benchmarking with YADE



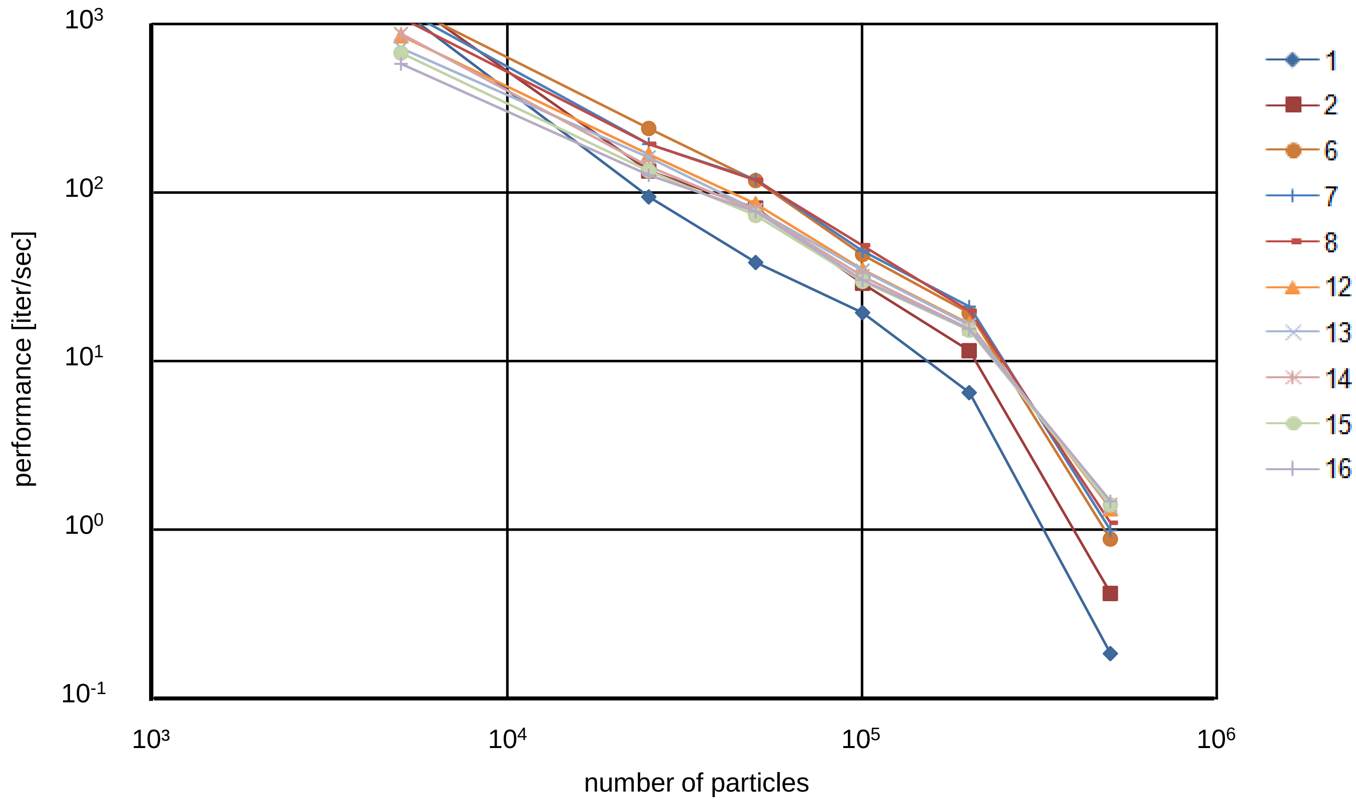
Performance Benchmark 2014 – *without Hyperthreading*, Serial Collider yade-2014-01-25



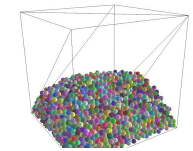
Performance Benchmarking with YADE



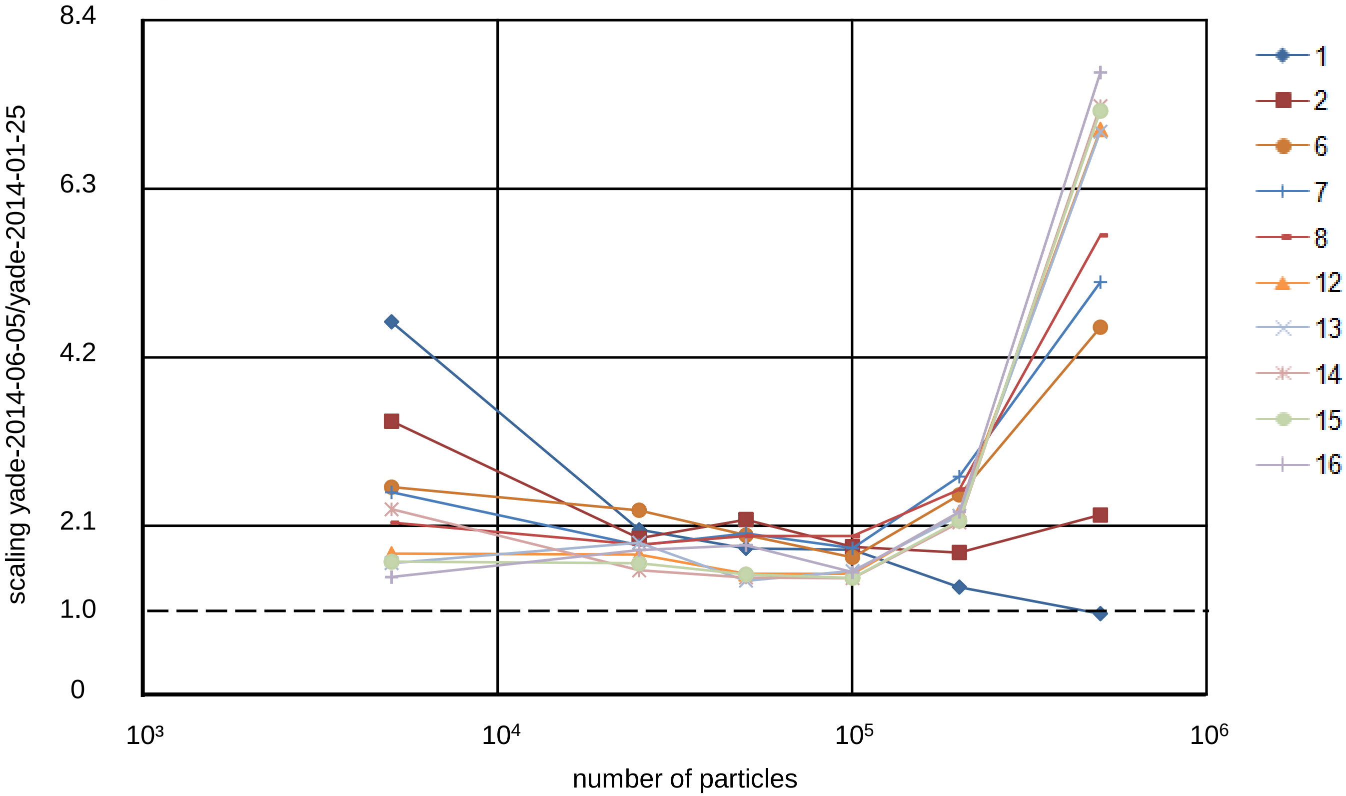
Performance Benchmark 2014 – *without Hyperthreading*, Parallel Collider yade-2014-06-05



Performance Benchmarking with YADE

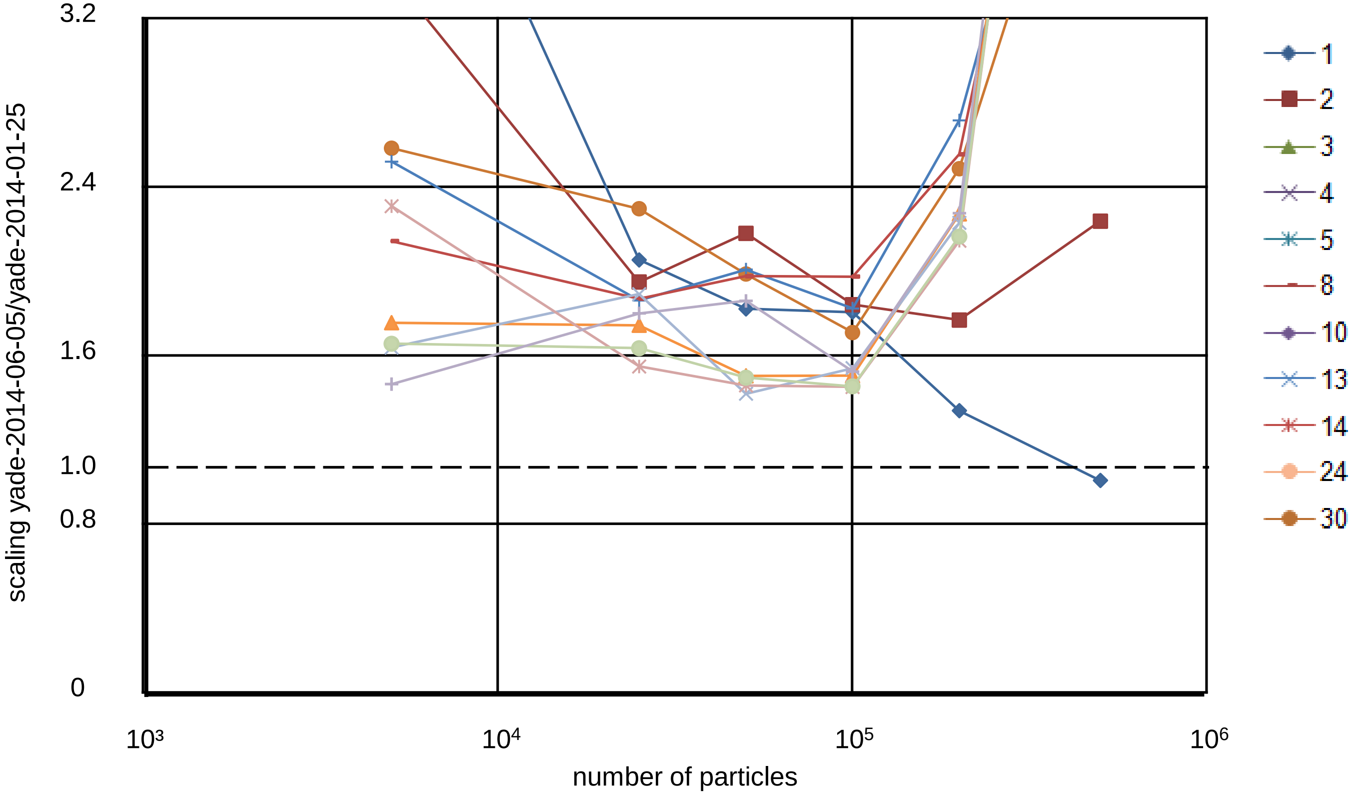


Performance Benchmark 2014 – *without Hyperthreading*, Scaling of the Parallel Collider yade-2014-06-05 VS. yade-2014-01-25

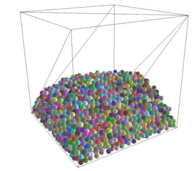


Performance Benchmarking with YADE

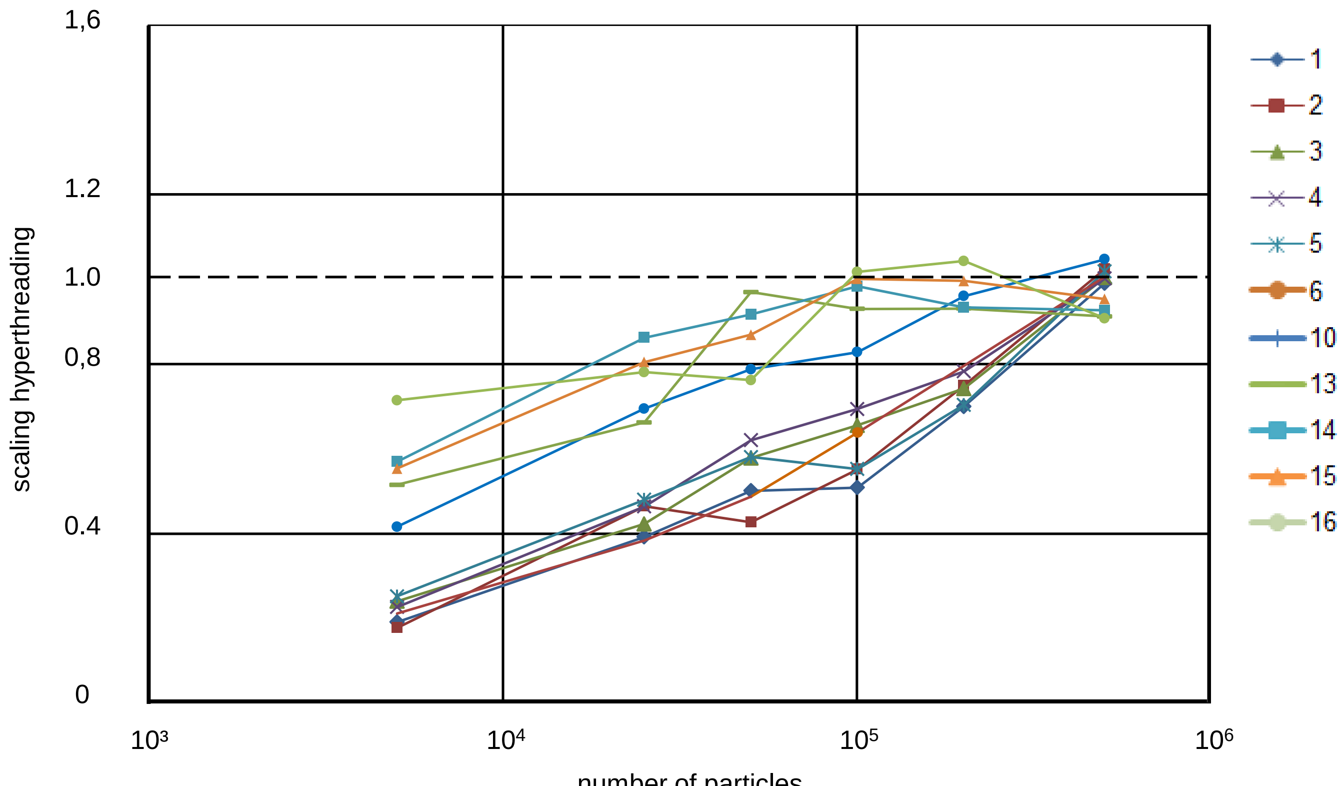
Performance Benchmark 2014 – *without Hyperthreading*, Scaling of the Parallel Collider (detail) yade-2014-06-05 VS. yade-2014-01-25



Performance Benchmarking with YADE

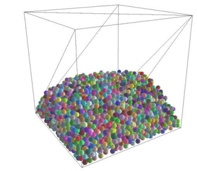


Performance Benchmark 2014 – Scaling of Hyperthreading using Parallel Collider yade-2014-06-05

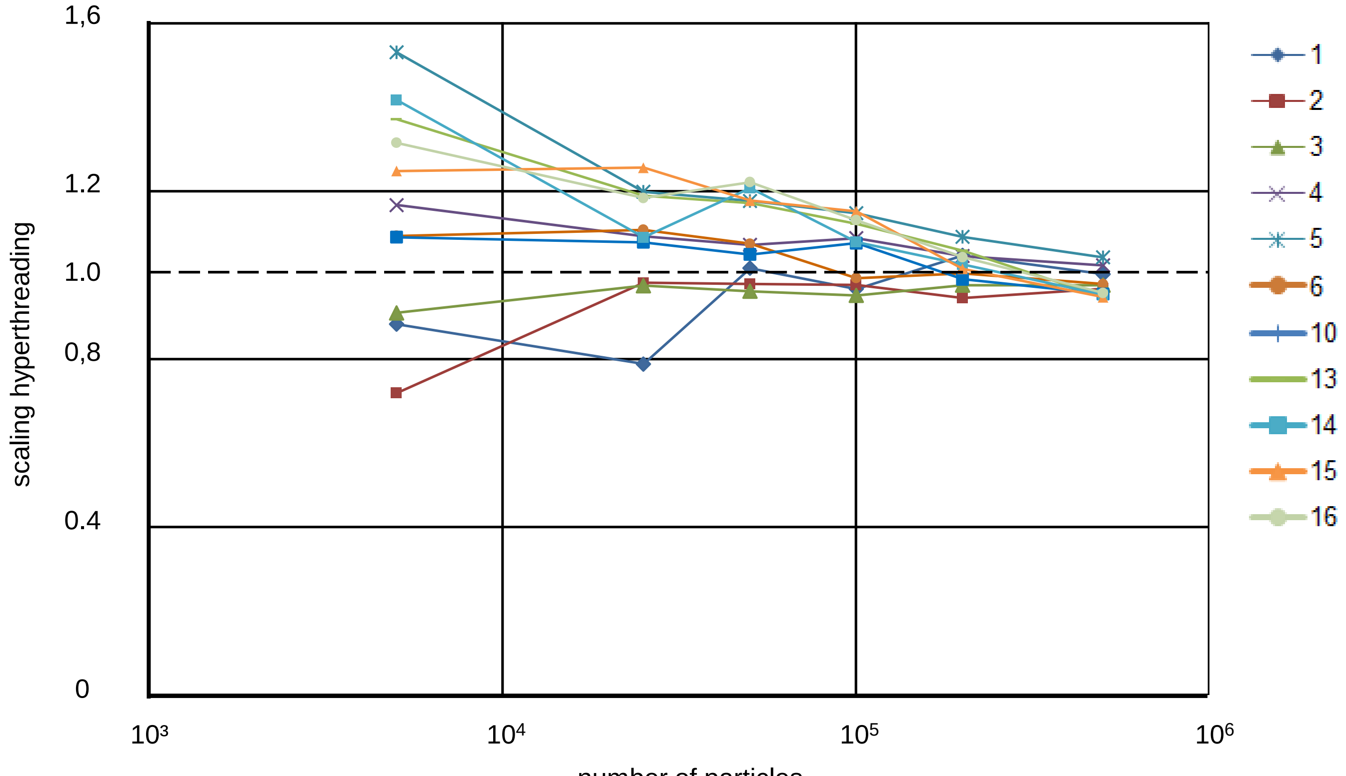


Performance is compared for equal number of threads (cores) in case of „using Hyperthreading" and „not-using Hyperthreading". Comparison of j threads for „not-using Hyperthreading" and 2xj threads for „using Hyperthreading" will be added.

Performance Benchmarking with YADE



Performance Benchmark 2014 – Scaling of Hyperthreading using Serial Collider yade-2014-01-25



Performance is compared for equal number of threads (cores) in case of „using Hyperthreading“ and „not-using Hyperthreading“. Comparison of j threads for „not-using Hyperthreading“ and 2xj threads for „using Hyperthreading“ will be added.

Performance Benchmarking with YADE

Performance Benchmark 2014 - repeatability, variance without HT

Coefficient of variation:
COV = std. dev. / mean
COV >= 0.1 is colored red

yade-2014-01-25 (Serial Collider)

4 repetitions with 5 runs for each thread-particle-combination -> each cell is a mean of up to 20 values

Threads	5025	25091	50238	100455	200801	501994	Mean row
1	0.08	0.11	0.05	0.08	0.08	0.03	0.07
2	0.19	0.07	0.03	0.08	0.06	0.05	0.08
3	0.10	0.08	0.02	0.05	0.03	0.02	0.05
4	0.01	0.04	0.06	0.03	0.06	0.03	0.04
5	0.07	0.03	0.03	0.01	0.04	0.04	0.04
6	0.08	0.06	0.03	0.03	0.02	0.01	0.04
7	0.04	0.03	0.02	0.02	0.02	0.03	0.03
8	0.03	0.02	0.03	0.02	0.03	0.01	0.02
9	0.02	0.02	0.05	0.04	0.02	0.03	0.03
10	0.07	0.04	0.04	0.02	0.02	0.01	0.03
11	0.07	0.01	0.03	0.02	0.00	0.02	0.02
12	0.17	0.09	0.05	0.05	0.04	0.02	0.07
13	0.13	0.08	0.04	0.11	0.04		0.08
14	0.03	0.18	0.07	0.04	0.03		0.07
15	0.13	0.09	0.08	0.07	0.03		0.08
16	0.13	0.00	0.02	0.02	0.01		0.04
Mean column	0.08	0.06	0.04	0.04	0.03	0.02	0.05

yade-2014-06-05 (Parallel Collider)

5 repetitions with 5 runs for each thread-particle-combination -> each cell is a mean of up to 25 values

Threads	5025	25091	50238	100455	200801	501994	Mean row
1	0.03	0.15	0.10	0.05	0.06	0.01	0.07
2	0.14	0.07	0.11	0.03	0.06	0.02	0.07
3	0.16	0.11	0.04	0.06	0.06	0.02	0.08
4	0.25	0.12	0.08	0.07	0.06	0.01	0.10
5	0.23	0.12	0.11	0.08	0.05	0.01	0.10
6	0.08	0.23	0.16	0.12	0.06	0.02	0.11
7	0.19	0.11	0.19	0.12	0.08	0.05	0.12
8	0.16	0.15	0.17	0.13	0.08	0.03	0.12
9	0.11	0.16	0.11	0.07	0.06	0.06	0.09
10	0.16	0.03	0.12	0.04	0.04	0.03	0.07
11	0.24	0.07	0.09	0.03	0.06	0.03	0.09
12	0.21	0.02	0.02	0.01	0.01	0.03	0.05
13	0.02	0.05	0.03	0.02	0.06		0.04
14							
15	0.10	0.11	0.01	0.04	0.06		0.06
16							
Mean column	0.15	0.11	0.10	0.06	0.06	0.03	0.08

Performance Benchmarking with YADE

Performance Benchmark 2014 - -j vs. --cores

- -j set NUM_OMPTHREADS
  - Use: yadedaily -j4
  - Specified number of cores is at good usage but asignment of threads to cores varies (also abserced by Giulia [1])
- --cores allocates threads to specified (virtual) cores
  - Use: yadedaily --cores "0,1,2,3"
    start yade with 4 threads AND one master thread at the first specified core
  - Partly only 2 or 3 threads are used. First one is always active
- Using --cores (yade-2014-06-05 (Parallel Collider)) does not reduce variance between runs

[1] https://answers.launchpad.net/yade/+question/244634

Performance Benchmarking with YADE

Performance Benchmark 2014 - -j vs. --cores

--cores “1,2,3,4“ -> python variable cores: [1, 2, 3, 4]

```
# OpenMP env variables must be set before loading yade libs ("import yade" below)
# changes have no effeect after libgomp initializes
if opts.cores:
    if opts.threads: print 'WARNING: --threads ignored, since --cores specified.'
    try:
        cores=[int(i) for i in opts.cores.split(',')]
    except ValueError:
        raise ValueError('Invalid --cores specification %s, should be a comma-separated
    opts.nthreads=len(cores)
    os.environ['GOMP_CPU_AFFINITY']=' '.join([str(cores[0])]+[str(c) for c in cores])
    os.environ['OMP_NUM_THREADS']=str(len(cores))
elif opts.threads: os.environ['OMP_NUM_THREADS']=str(opts.threads)
else: os.environ['OMP_NUM_THREADS']='1'
```

GOMP_CPU_AFFINITY
= ‘1 1 2 3 4’

```
# OpenMP env variables must be set before loading yade libs ("import ya
# changes have no effeect after libgomp initializes
if opts.cores:
    if opts.threads: print 'WARNING: --threads ignored, since --cores s
    try:
        cores=[int(i) for i in opts.cores.split(',')]
    except ValueError:
        raise ValueError('Invalid --cores specification %s, should be a
    opts.nthreads=len(cores)
    os.environ['GOMP_CPU_AFFINITY']=' '.join([str(c) for c in cores])
    os.environ['OMP_NUM_THREADS']=str(len(cores))
elif opts.threads: os.environ['OMP_NUM_THREADS']=str(opts.threads)
else: os.environ['OMP_NUM_THREADS']='1'
```

GOMP_CPU_AFFINITY
= ‘1 2 3 4’

Performance Benchmarking with YADE

Performance Benchmark 2014 - -j vs. –cores:

- withHT parallel collider –j4

	Parallel collider		Serial Collider	
particles	performance [iter/sec]	Std.Dev.	performance [iter/sec]	Std.Dev.
5025	822.17	18.81%	328.38	8.74%
25091	160.63	9.10%	77.22	8.59%
50238	80.34	4.31%	48.97	5.95%
Score:	17674		8463	
Common time:	1046.76 s		2055.20 s	

Measured performance for each test, Parallel Collider

5025	**1248.2**	**1208.0**	883.5	637.1	841.4
	734.1	693.2	865.8	816.1	788.2
	664.0	720.0	780.0	921.7	711.4
	818.4	794.4	752.5	865.8	699.7
25091	**187.4**	**190.8**	153.7	178.2	170.0
	151.4	141.8	152.6	134.1	149.6
	164.6	152.2	150.7	177.2	164.2
	174.2	150.0	162.1	151.8	156.0
50238	**86.9**	74.0	81.5	78.7	82.4
	83.8	82.4	**86.0**	79.8	**85.0**
	78.7	80.0	79.0	75.1	76.3
	78.2	75.8	81.3	79.9	82.0

iterations/second

Performance Benchmarking with YADE

Performance Benchmark 2014 - -j vs. --cores: test with 4 threads on two different sets of cores, n=20

- withHT parallel Collider mit --cores "8,9,10,11" checkPef_2014-06-05.py
- A: --cores “8,9,10,11”; B: --cores “12,13,14,15”

A+B simultaneously

	A	+	B	
particles	performance [iter/sec]	Std.Dev.	performance [iter/sec]	Std.Dev.
5025	1386.06	27.32%	1423.66	25.31
25091	158.83	23.59%	163.05	25.31
50238	76.86	26.37%	75.54	25.88
Score:	22542		22962	
Common time:	998.65 s		994.28 s	

Only B

	B	
	performance [iter/sec]	Std.Dev.
	1743.47	16.92%
	183.56	22.55%
	86.91	24.99%
Score:	27234	
Common time:	847.85 s	

- Performance between instances of Yade started simultaneously is comparable for different core sets
- Performance of a single Yade instance is higher then when running simultaneously

Performance Benchmarking with YADE

Performance Benchmark 2014 - -j vs. --cores: test with 4 threads on two different sets of cores, n=20

- withHT parallel Collider
- Sample output for A: --cores “8,9,10,11”:

5025	1184.0	**2094.7**	863.3	1468.5	1349.4
	1109.5	**1932.6**	846.2	1593.4	1349.4
	1113.6	**1945.1**	924.5	1610.5	1313.9
	1113.6	**1957.8**	905.0	1585.0	1461.3
25091	128.7	170.0	161.6	120.4	**217.4**
	119.1	168.6	158.3	119.8	**222.8**
	118.2	169.5	160.8	120.9	**225.2**
	117.9	169.1	164.6	121.4	**222.1**
50238	78.2	56.7	**112.0**	56.0	80.4
	79.3	56.9	**113.1**	56.3	80.4
	79.4	57.3	**110.2**	56.9	80.0
	78.6	57.2	**112.0**	56.1	80.1

iterations/second

- Performance is repeating every 5 runs
- Maximum performance is achieved in different runs

Performance Benchmarking with YADE

Performance Benchmark 2014 - -j vs. --cores: test with 4 threads on two different sets of cores, n=20

- withHT parallel Collider with cores Correction
- Sample output for B: --cores “12,13,14,15”:
- Slight decrease in performance concerning maximum values but less variation

5025	1762.3	1692.6	1731.6	1772.8	**1838.0**
	1721.6	1655.2	1752.0	1712.0	1731.7
	1721.6	1712.0	1721.8	1721.8	1646.0
	1838.0	1702.2	1721.8	1702.2	1731.7
25091	184.1	184.6	185.7	185.1	183.5
	184.1	**198.1**	184.1	196.8	184.1
	185.2	**198.1**	184.1	184.6	184.6
	187.4	**198.1**	182.5	195.0	185.2
50238	**91.7**	89.7	90.9	90.7	88.0
	89.5	89.1	**91.9**	90.7	89.6
	90.2	91.1	90.7	90.3	90.4
	90.2	90.5	90.3	86.9	**91.5**

iterations/second

Performance Benchmarking with YADE

Performance Benchmark 2014 - -j vs. --cores: test with 4 threads on two different sets of cores, n=20

- withHT serial collider
- A: --cores "8,9,10,11"; B: --cores "12,13,14,15"

A+B simultaneously

	A	+	B	
particles	performance [iter/sec]	Std.Dev.	performance [iter/sec]	Std.Dev.
5025	323.31	22.28%	323.03	22.52%
25091	74.88	16.47%	74.87	17.04%
50238	41.38	18.94%	41.48	19.37%
Score:	7914		7914	
Common time:	2253.95 s		2254.84 s	

A+B separately

	A		B	
	performance [iter/sec]	Std.Dev.	performance [iter/sec]	Std.Dev.
	497.68	21.26%	491.13	16.92%
	84.58	16.95%	85.08	16.96%
	47.24	21.85%	47.33	21.54%
Score:	10169		10133	
Common time:	1818.85 s		1820.71 s	

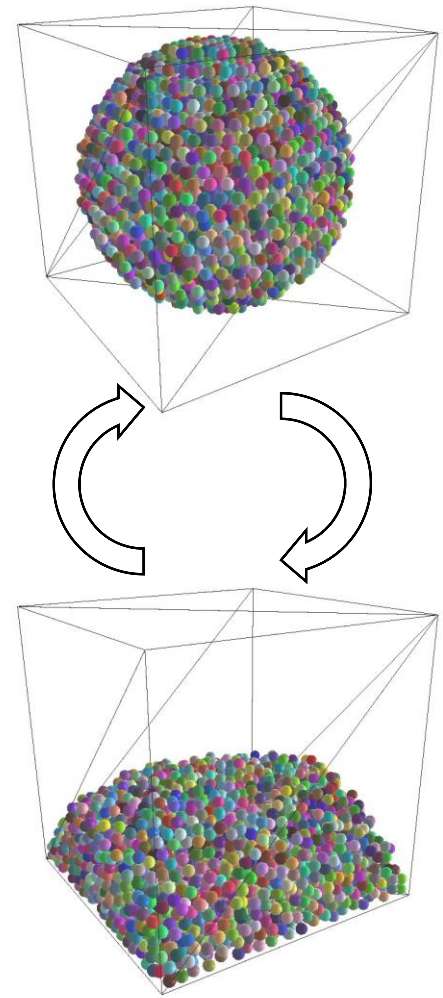
Performance Benchmarking with YADE

Built-In Performance Benchmark – Constant Number of Iterations

- Spherical packing of spheres inside a box is exposed to gravitational deposition
- Repeated test (3, 5, 20 repetitions)
- Dimensions of the packing, i.e. number of particles and dimensions of the box change:

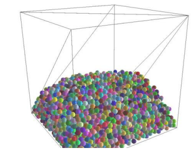
Number of particles	Iterations	Coefficient
5,025	12,000	110
25,091	12,000	110
502,38	12,000	110
100,455	12,000	110
200,801	12,000	110
501,994	12,000	110

- Test was not created for benchmarking but for regression tests after crucial commits

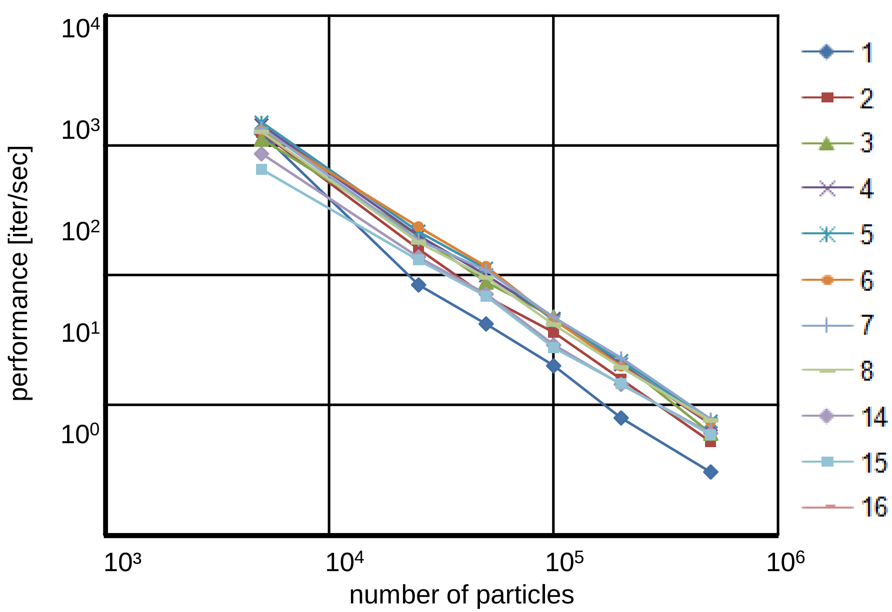


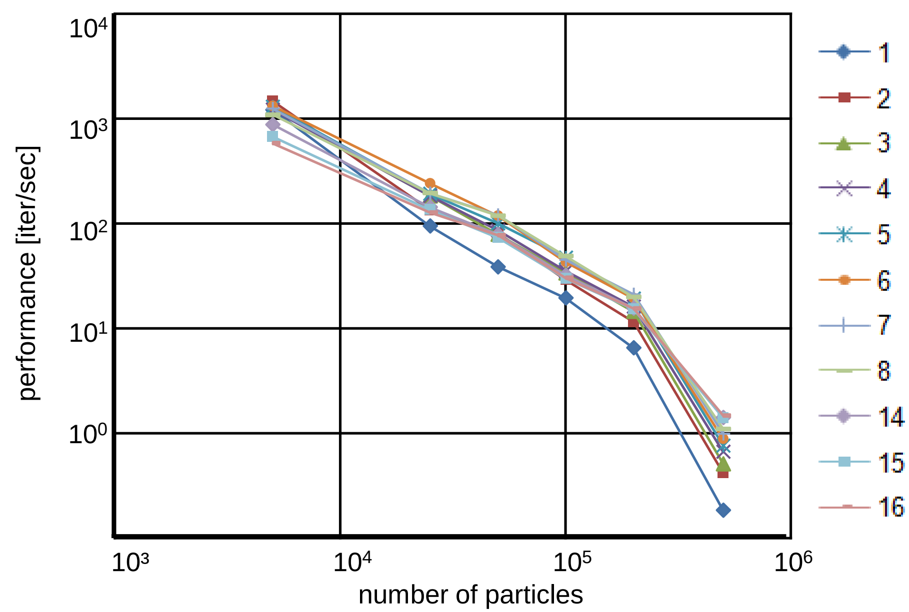
Model setup of built-in performance test

Performance Benchmarking with YADE

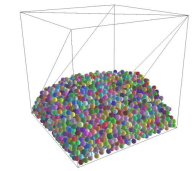


Built-In Performance Benchmark – Constant Number of Iterations – woHT, Parallel Collider yade-2014-06-05



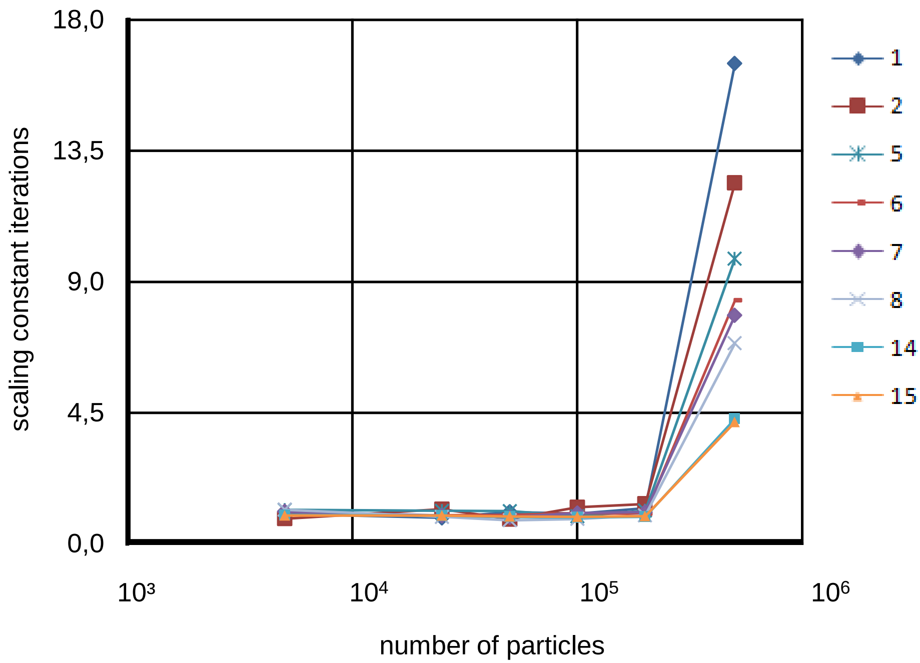


Performance Benchmarking with YADE

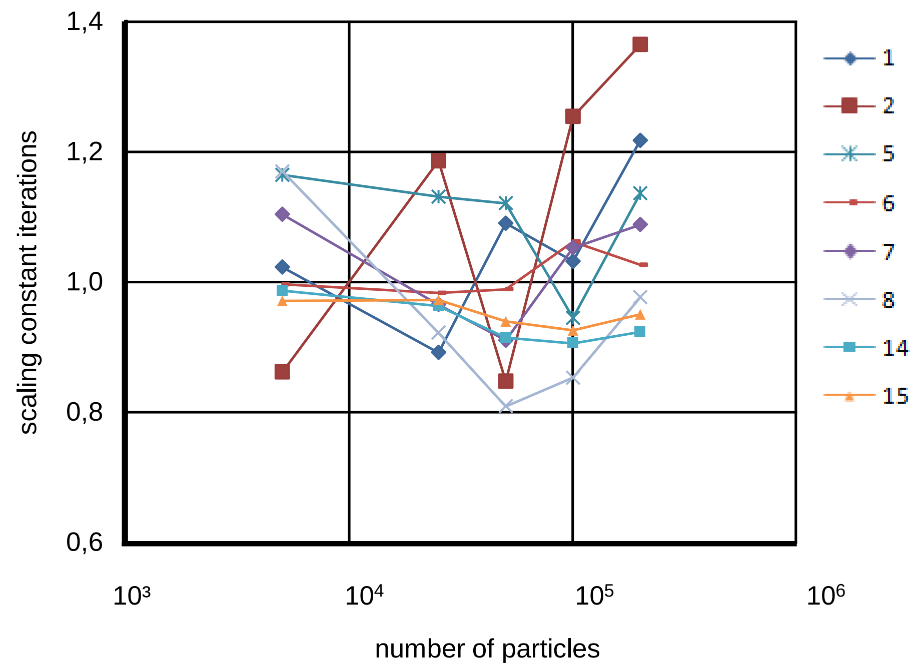


Built-In Performance Benchmark – Constant Number of Iterations – Scaling between constant and changing iterations

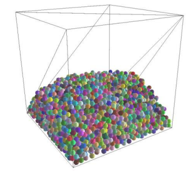
total



detail

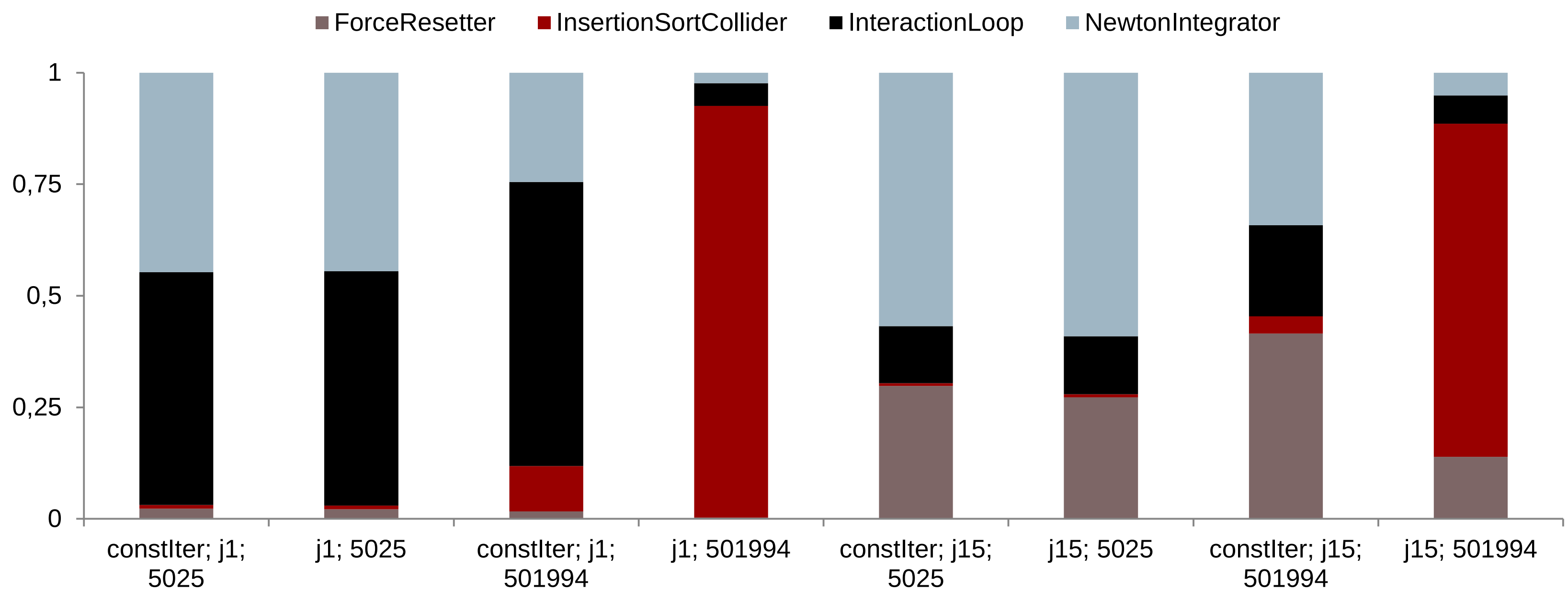


Performance Benchmarking with YADE



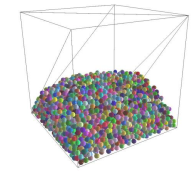
Built-In Performance Benchmark – Timing Stats: constant vs. changing Iterations Parallel Collider woHT yade-2014-06-05

- Sample output from timing stats: 5025 particles 501994 particles using 1 or 15 cores:



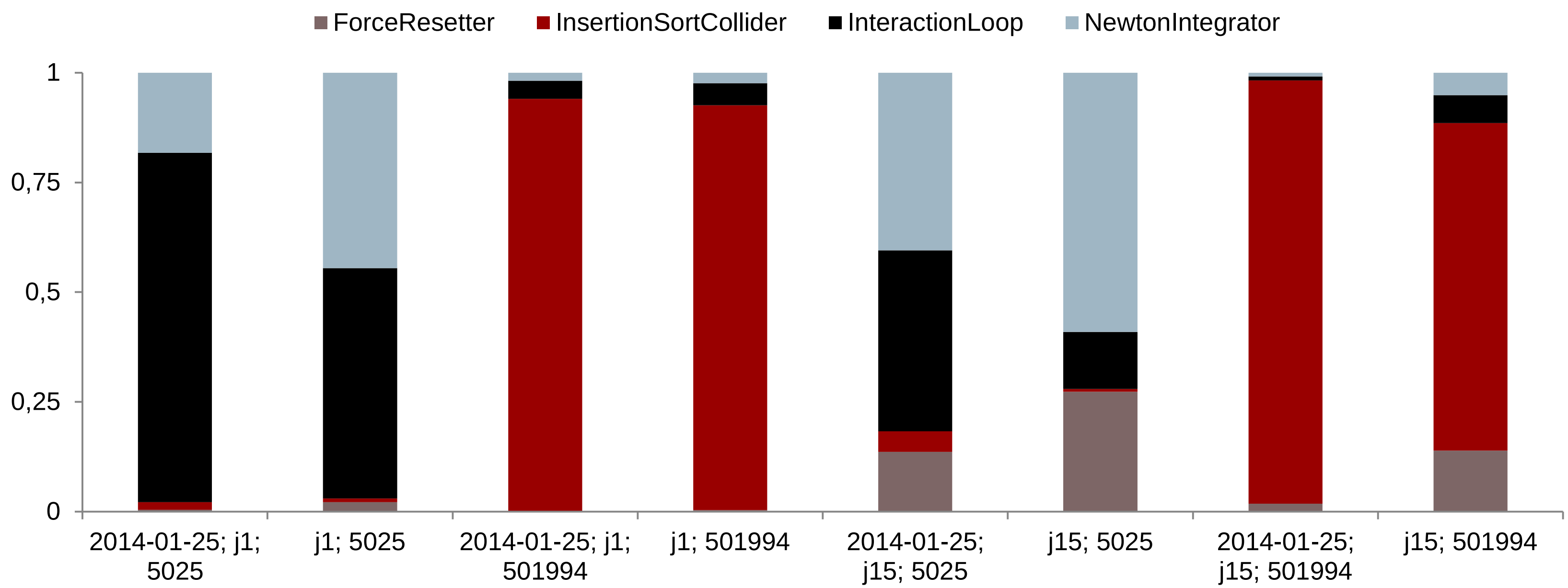
- 500k particles:
  - if nbIter = 10 -> immense share of InsertionSortCollider
- nbIter = 12,000 (constant): ForceResetter share increases
- Increasing number of threads: share of NewtonIntegrator and ForceResetter increase

Performance Benchmarking with YADE



Built-In Performance Benchmark – Timing Stats: Serial vs. Parallel Collider woHT

- Sample output from timing stats: 5025 particles 501994 particles using 1 or 15 cores:

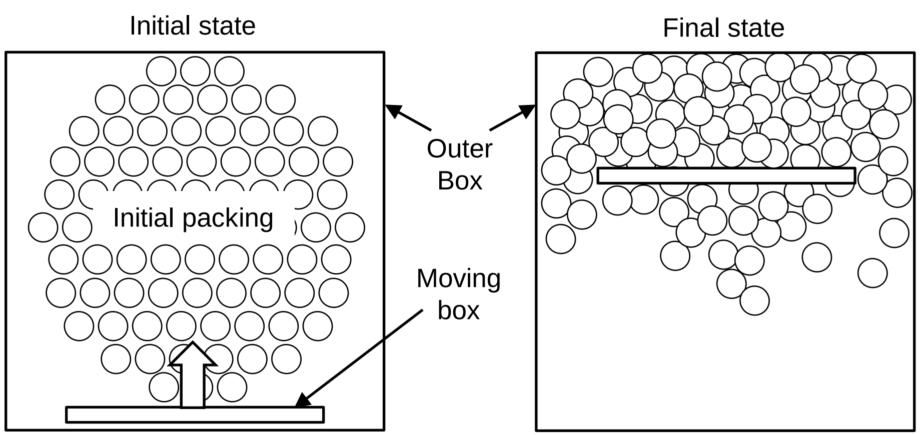


- 500k particles:
  - parallel InsertionSortCollider decreases time for collision detection
- 5k particles:
  - Share of NewtonIntegrator and ForceResetter increse; InteractionLoop and Collider decrease

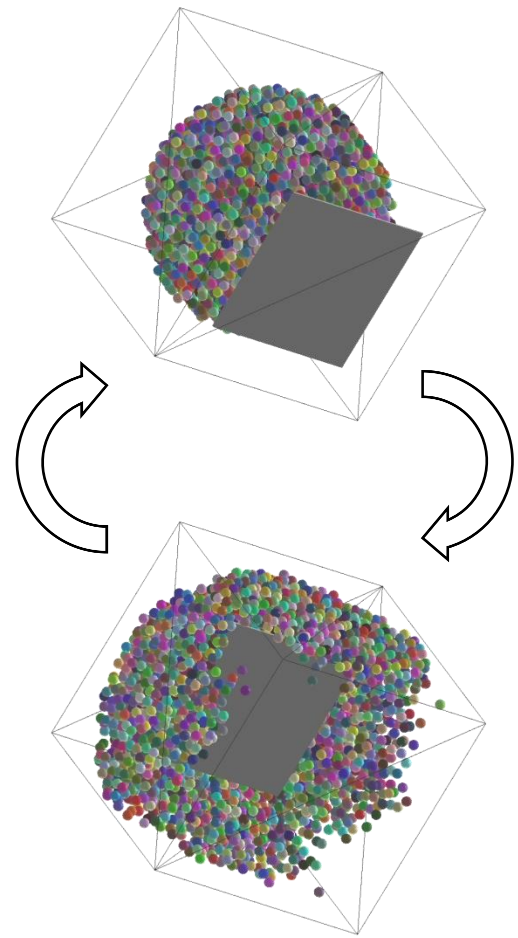
Performance Benchmarking with YADE

Dynamic Performance Benchmark

- Remember changing number of iterations in built-in performance test?
- Now timestep has to be adjusted to dynamics of simulation, i.e. velocity of facets (provided a constant sphere radius)
- What is a suitable way of testing dynamic performance?



Model setup of dynamic performance test



Performance Benchmarking with YADE

Launchpad Discussions, Wiki-Articles

- Dynamic setup: tumbling bowl (Martin Nie): http://answers.launchpad.net/yade/+question/242644
- Comparisons with PFC3D – Yade: https://yade-dem.org/wiki/Comparisons_with_PFC3D
- OpenMP -> MPI: https://answers.launchpad.net/yade/+question/216187
- https://answers.launchpad.net/yade/+question/215540
- https://answers.launchpad.net/yade/+question/246489
- …
- Results shown here shall be included in wiki-page created by Klaus: https://yade-dem.org/wiki/Performance_Test

Summary and Outlook

Summary

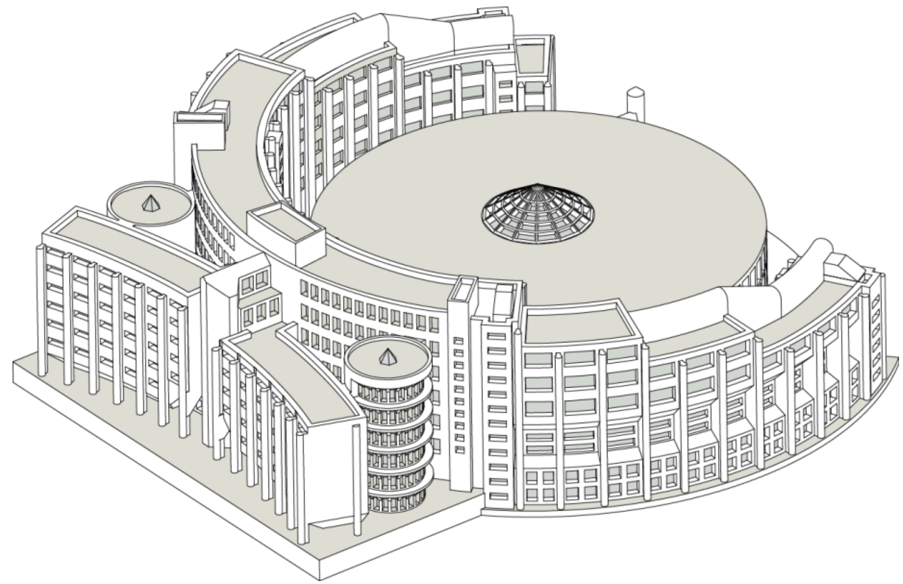
- Performance is … depending on what you simulate
- Serial collider took a lot of time with large amount of particles -> New collider is doing way better. Especially without Hyperthreading.
- --cores option should be revised
- variation and repeatability should be considered

Outlook

- Test whether candidate for dynamic performance testing is stable
- Evaluate the data for non-vibrating tub
- Implement timing stats as bar chart in Qt-Controller (enabled by check box)

Thank you!

M. Sc. Alexander Eulitz
Research Engineer
E-Mail: eulitz@iwf.tu-berlin.de
Phone: +49 30 314 24963
Office PTZ 1
Pascalstraße 8-9
10587 Berlin



IS Collider

Bzr3000
(branch history no longer available it seems... only a git merge, sorry for that)

Small scale optimizations.
Exemple 1:

```
-  if(unlikely(!I->functorCache.geom || !I->functorCache.phys)){
+ if(unlikely(!I->functorCache.geom)){
I->functorCache.geom=geomDispatcher->getFunctor2D(b1_->shape,b2_-
>shape,swap);
```

IS Collider

Bzr3000

Small scale optimizations.
Exemple 2:
Skip 50% of the bound inversions:
- min vs. min
- max vs. max

IS Collider

Bzr3000

Different logic: the collider doesn't care when bboxes are separating

Skip 50% of the bound inversions again: max vs. min

Overall 75% of the inversions are skipped

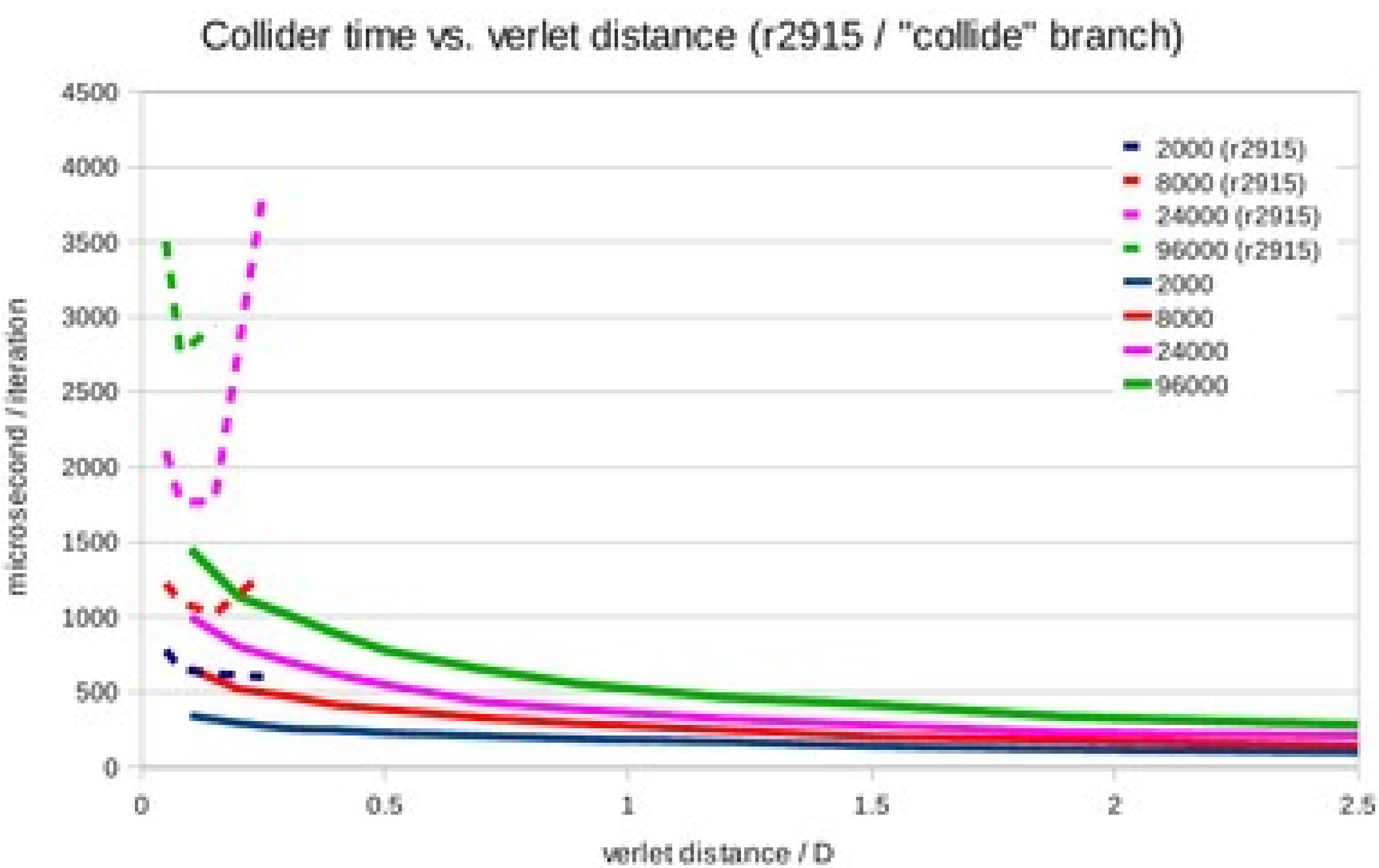
IS Collider

Bzr3000

Striding based on actual motion instead of abs. value of max velocity
+
Motion-based update of the sweep length

IS Collider

Bzr3000



Parallel IS Collider

- Parallel sorting algorithms are notorious and well documented in online contents
- It seems our case is very specific though, due to the highly pre-ordered lists we have
- cutting the lists in smaller chunks naively scales almost linearly (!) for large N
- almost all the other loops are parallelized (exception for the erase loop – after a bad bug reported by Anton)
- running the 3 axis in parallel could be also tested (?)

DEM on GPU

Václav Šmilauer

woodem.eu ib-keramik.de

July 2014

Abstract

Woo featured an experimental cldem module implementing some DEM algorithms in OpenCL, suitable for running on variety of computing devices including GPUs. The implementation is briefly described and suitability of the GPGPU architecture for DEM is evaluated.

DEM on GPU

Václav Šmilauer

woodem.eu ib-keramik.de

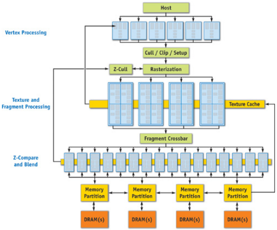
July 2014

Outline

What is (GP)GPU?

- use the computing power of Graphics Processing Units (GPU) for General Purposes (GP) instead of just graphics
- specific massively parallel architecture: same code run over thousands of different data
- CUDA: nVidia proprietary, nVidia-GPU only; more high-level API
- OpenCL: cross-device (GPU, CPU, ...), standard, relatively low-level

What is (GP)GPU?

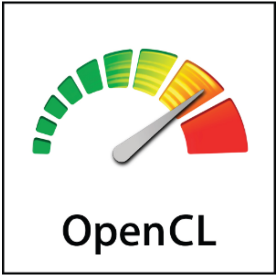


- use the computing power of Graphics Processing Units (GPU) for General Purposes (GP) instead of just graphics
- specific massively parallel architecture: same code run over thousands of different data
- CUDA: nVidia proprietary, nVidia-GPU only; more high-level API
- OpenCL: cross-device (GPU, CPU, ...), standard, relatively low-level

What is (GP)GPU?

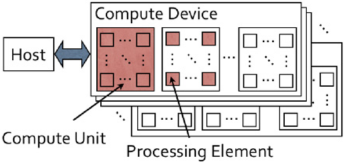
- use the computing power of Graphics Processing Units (GPU) for General Purposes (GP) instead of just graphics
- specific massively parallel architecture: same code run over thousands of different data
- CUDA: nVidia proprietary, nVidia-GPU only; more high-level API
- OpenCL: cross-device (GPU, CPU, ...), standard, relatively low-level

What is (GP)GPU?



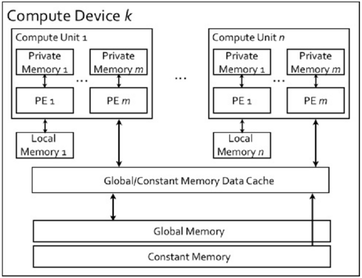
- use the computing power of Graphics Processing Units (GPU) for General Purposes (GP) instead of just graphics
- specific massively parallel architecture: same code run over thousands of different data
- CUDA: nVidia proprietary, nVidia-GPU only; more high-level API
- OpenCL: cross-device (GPU, CPU, ...), standard, relatively low-level

OpenCL architecture



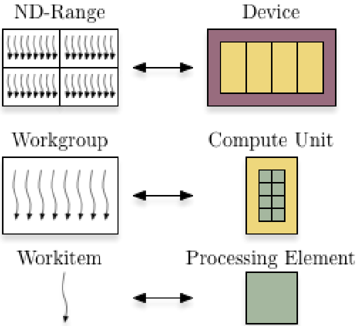
- device / compute unit / processing element
- corresponding 3 types of memory
  - private: very fast, small (e.g. 64KB)
  - local: slower, rel. small (e.g. 512KB)
  - global: very slow, big (e.g. 4GB)
- division of labor: all data / work-group / work-item

OpenCL architecture



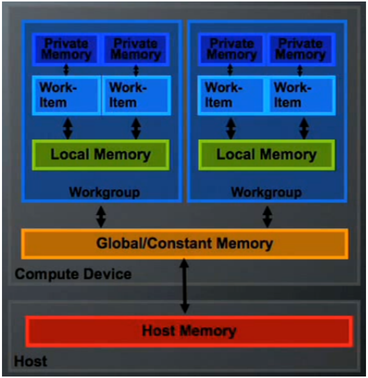
- device / compute unit / processing element
- corresponding 3 types of memory
  - private: very fast, small (e.g. 64KB)
  - local: slower, rel. small (e.g. 512KB)
  - global: very slow, big (e.g. 4GB)
- division of labor: all data / work-group / work-item

OpenCL architecture



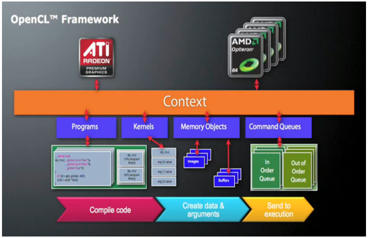
- device / compute unit / processing element
- corresponding 3 types of memory
  - private: very fast, small (e.g. 64KB)
  - local: slower, rel. small (e.g. 512KB)
  - global: very slow, big (e.g. 4GB)
- division of labor: all data / work-group / work-item

OpenCL architecture



- device / compute unit / processing element
- corresponding 3 types of memory
  - private: very fast, small (e.g. 64KB)
  - local: slower, rel. small (e.g. 512KB)
  - global: very slow, big (e.g. 4GB)
- division of labor: all data / work-group / work-item

OpenCL execution model



- intialize "context" (e.g. choose the OpenCL device, create queues)
- ask OpenCL to compile your source code to device-specific machine code
- enqueue (as many times you want):
  - copy buffers from machine RAM to the device (GPU)
  - kernels (routines in your source) to be executed
  - copy buffers from the device to RAM
- ask OpenCL to execute the queue, wait (or not) for it to finish

DEM on the (GP)GPU

GPGPU features

1. massive parallel performance
2. predictible memory access is the key (keep it private/local)
3. local/shared memory capacity rather low (e.g. 512KBs)
4. all processing elements ("cores") on the compute unit must execute *the same code*: if you branch, others wait

DEM requirements

1. relatively parallel
2. unpredictable (moving particles) and non-local memory access necessary
3. lots of local data (e.g. 640b / Particle)
4. lots of branching necessary (different particle shapes)

Outline

1. (GP)GPU & OpenCL intro
2. `clDem`
3. Examples

What was done in clDem

- implementation of math routines (3x3 matrices, quaternions, ...)
- DEM with sphres, walls, clumps, two contact models
- collision detection still on the CPU
- data in global memory, write done using atomics & many tricks
- Woo ↔ clDem simulation translation
- Python wrapper of clDem, export to VTK; integration into Woo (OpenGL)
- auto-checking of positions/velocities/contacts/. . . against Woo, running the same simulation on the CPU
- 7 months of work of me + 1 employee, paid consultations with streamcomputing.eu
- still slower than CPU

Lessons learned

- OpenCL platforms are buggy, incl. miscompilation and non-conforming behavior

  gold nVidia is **the most buggy**; don't buy their products!

  silver AMD is in the middle

  bronze Intel is the best, but still with issues

- hard to write code without assuming memory sizes
- (GP)GPUs are great for tasks with strong locality: (some) matrix operations, image processing
- the hype was greater than the truth (impressive marketing)

Verdict

Is (GP)GPU usable for DEM?

No.

Is (GP)GPU usable for DEM?

No.

Outline

Show us the code!

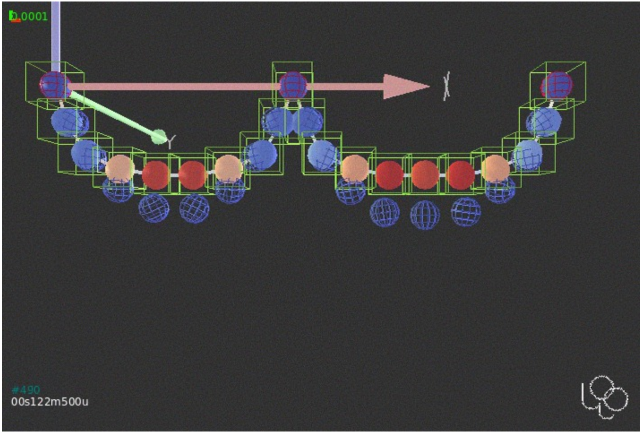
- math: https://code.launchpad.net/~eudoxos/+junk/cl-math
- DEM itself: https://code.launchpad.net/~eudoxos/+junk/cl-dem0

Example explanation

- in examples, wire particles run in Woo on the CPU; full particles run via OpenCL.
- OpenCL not updated at every step (only until GPU sends the data back)

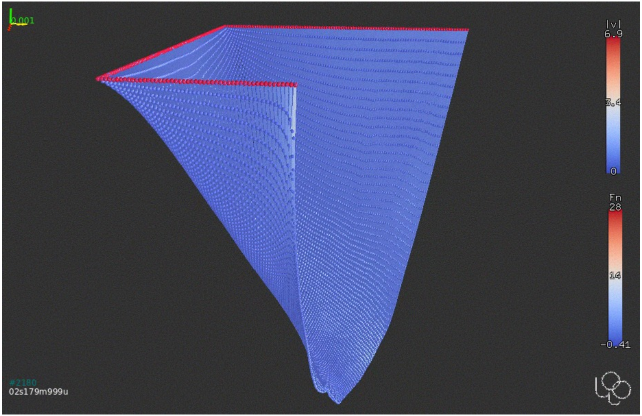
Example: chain (test-chain-woo.py)

Pre-existing contacts between many particles. GPU synced every 100 steps.



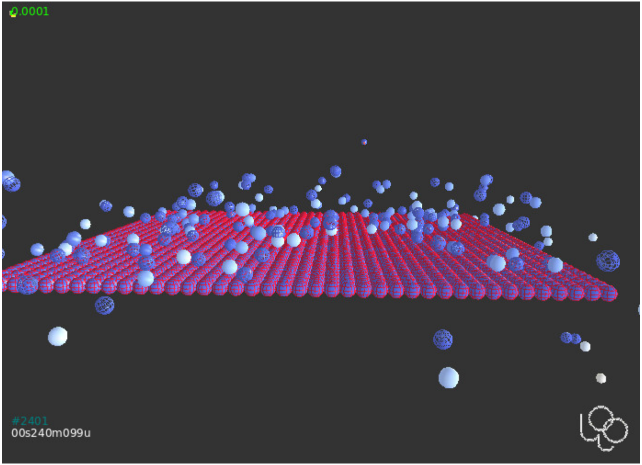
Example: cloth (test-cloth.py)

Pre-existing contacts between particles. GPU synced every 100 steps, that's why full particles are lagging and catching-up.



Example: jumping spheres (test-jump2.py)

This simulation includes collision detection, which is done on the CPU (as the first step). GPU not synced regularly, depending on when the need for collision detection arises and data are sent back to the CPU.



Woo: Windows port

Václav Šmilauer

woodem.eu ib-keramik.de

July 2014

Abstract

Woo supports the Windows platform. This talk summarizes the effort required, changes and limitations. The result (Windows binaries) of this work can be downloaded from http://launchpad.net/woo.

Windows platform

- widely used
- 32bit and 64bit variants (both XP and W7)
- no good shell (needed for automation)
- no integrated set of development tools
- no ABI, library packages, standard header locations

Compilation for Windows

- cross-compilation on Linux – MXE project
- – does not support Python
- use Microsoft Visual Studio
- – bad support for c++11; many libs needed don't support it
- use Windows port of GCC: MinGW-w64
- – recompile all libs by yourself (boost, VTK, Qt, Python, ...)

Prepare libs

1. install MinGW-w64
2. install MSYS (shell)
3. install Python 2.7 (windows build)
4. TONS of little workarounds and patches
5. boost
6. libXML, libiconv, gettext, glib, gts
7. cmake, VTK
8. Qt, SIP, PyQt, QGLViewer, freeglut, gle
9. pre-compiled: numpy, matplotlib, PIL

Prepare libs: summary

Repeat 100×

- google up upstream tarball for the lib
- `wget http://...; tar xfz ...`
- `configure --prefix=/c/MinGW64`
- `mingw32-make -j8`
- google up errors, find Windows-specific patches and workarounds, apply those, back to configure, iterate until no errors show up
- `make install`

Compile Woo

- use `setup.py` (distutils; alternate: setuptools, distribute, distutils2)
- – must *hardcode* paths for libs, includes, ...
- – does not recompile changed files
- COFF instead of ELF: file too big (2GB?), chunks
- slow linker
- 23× `#ifdef __MINGW64__`
  - use Windows API funcs instead of POSIX (`getpid()`/`GetCurrentProcessId()`)
  - guard Linux-only functionality (debugging)

Run Woo

- 12× `if sys.platform=='win32'`
  - some modules don't work under Windows (HDF5)
  - several copies of env. vars, use win32 API for updating
  - no real support for symlinks
  - device names: `nul`, `/dev/null`

Freezing, distributing

- http://pyinstaller.org
- http://nsis.sf.net
- libs (rarely updated, 42MB) + Woo (11MB)
- no automatic updates (like packages)

Conclusion

A Woo runs on Linux.

C We don't use Linux, it absolutely must run under Windows.

A *(2 months later)* Woo supports Windows as platform.

B Does it run under Linux?

A Yes. But C said you needed the Windows port.

B We have a Linux cluster here.

A ??

C I though we were not allowed to use it.

B The numerics guys can use it.

Conclusion

A Woo runs on Linux.

C We don't use Linux, it absolutely must run under Windows.

A *(2 months later)* Woo supports Windows as platform.

B Does it run under Linux?

A Yes. But C said you needed the Windows port.

B We have a Linux cluster here.

A ??

C I though we were not allowed to use it.

B The numerics guys can use it.

Pore-scale simulation of drainage for two-phase flow in dense sphere packings

Chao Yuan, Bruno Chareyre and Felix Darve

Grenoble-INP, UJF, CNRS UMR 5521, 3SR Lab.

July 7th-9th, 2014

1st Yade Workshop

Layout

Introduction

Objective

- Assign a pore-scale numerical model to simulate the drainage process of initially saturated granular materials.
- Apply the model to investigate capillary pressure and saturation ($P^c - S_r$) relationship.

Assumptions

- **Two-phase flow**: Nonwetting phase (NW-phase) and wetting phase (W-hase). Contact angle $\theta = 0$.
- **Quasi-static regime**: Capillary number (Ca) is quite small, capillary force dominate. P^c is applied in slow time scale, to maintain the system to equilibrate.

Network

A pore network in 3Ds with **Regular Triangulation** method is assigned to upscale from a pore unit to a dense spheres packing, which is based on PFV scheme (Chareyre et al. (2012)).

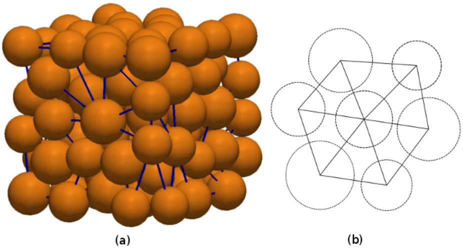
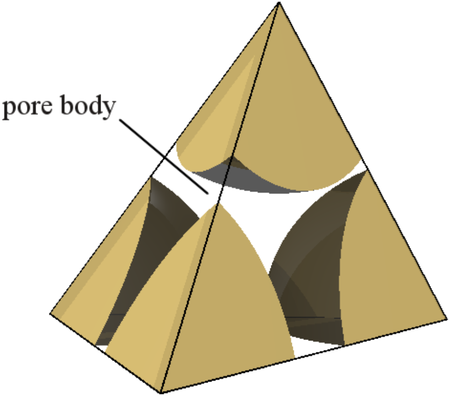


Figure : Defination of pore network, generated by regular triangulation in 3D(a) and 2D(b)

Pore body and pore throat

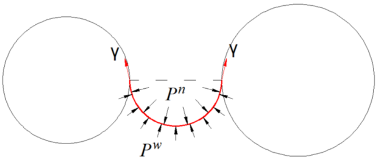
Pore body defined by tetrahedral element of the finite volume decomposition.

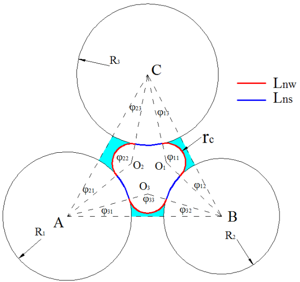


Entry capillary pressure P_e^c

The approach for calculating P_e^c is based on MS-P method suggested by Ma et al. (1996); Mayer and Stowe (1965); Princen (1969) and Joekar-Niasar et al. (2010), which follows from the balance of forces for pore throat section.

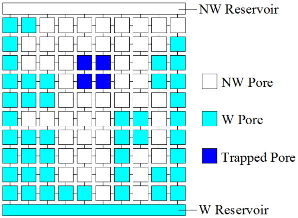
$$\sum F = F^p + T^\gamma = 0$$





Drainage and trapping

- Receding W-phase maybe form clusters which are disconnected with the main wetting body/water reservoir.(residual saturation)
- If consider film flow or evaporation, trapped W-phase also can be drained.



Demonstration of NW-phase invasion and W-phase trapping in network (in 2D mapping for clarify).

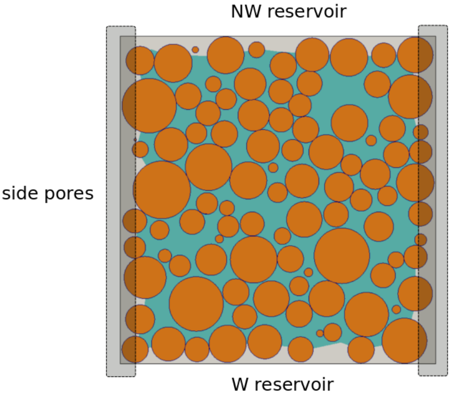
Boundary conditions

Reservoir boundaries

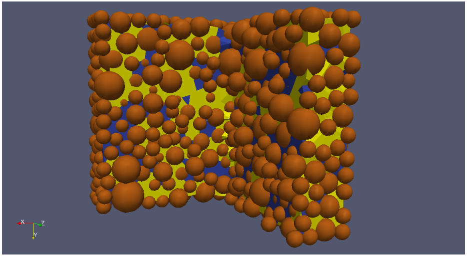
All pore bodies on top and bottom boundaries are assumed to be connected to NW-phase and W-phase reservoirs, respectively.
(Correspondingly, S_r will not take those pores into account.)

Side boundaries

Pore throats connected to the side boundaries are closed/open.
(S_r needs consistency.)



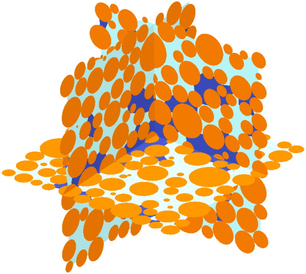
Test



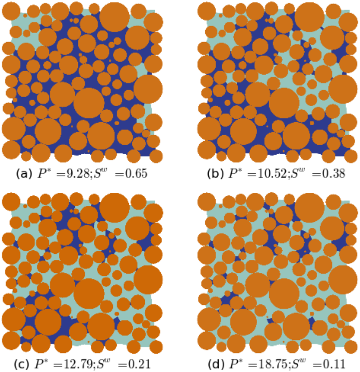
Yuan C., Chareyre B., Darve F. (2014), XXth CMWR, Stuttgart

Tests

Drainage mode:
W-phase can be trapped; Side pore throat is open.



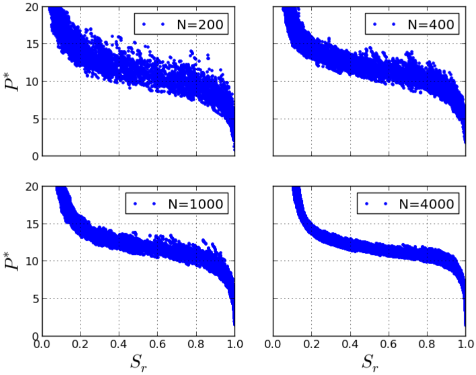
NW-phase trapping. Brown is solid, dark blue is W-phase, light green is NW-phase.



process of drainage (sphere number=1000)

REV

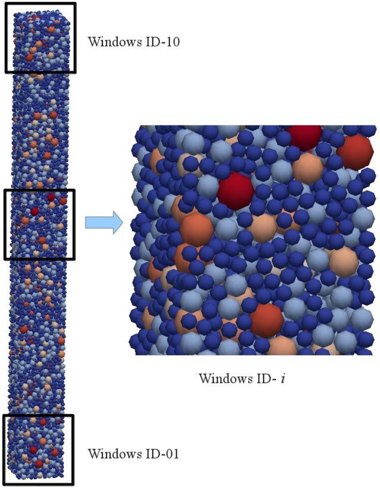
Drainage mode
W-phase can be trapped; Side pore throat closed.



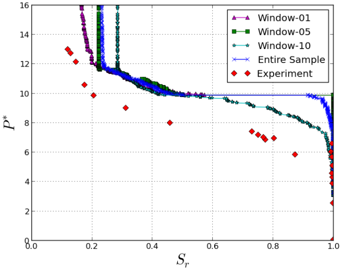
Comparison with experiment

Numerical setup

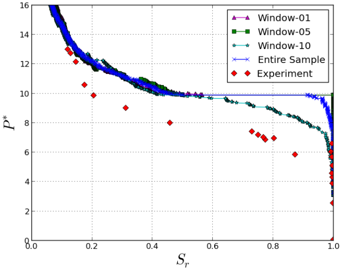
- size: 7mm*7mm*70mm (cuboid)
- porosity and PSD: same as work of Culligan et al.,2004
- reservoirs: top-NW; bottom-W
- mode: side open
- no gravity effects.
- test zone: 10 windows zoom



Comparison with experiment



W-phase trapped

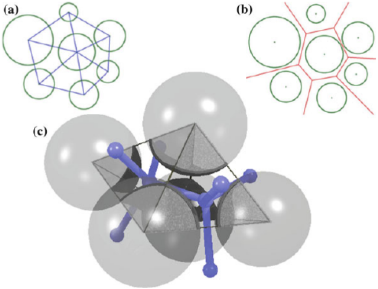


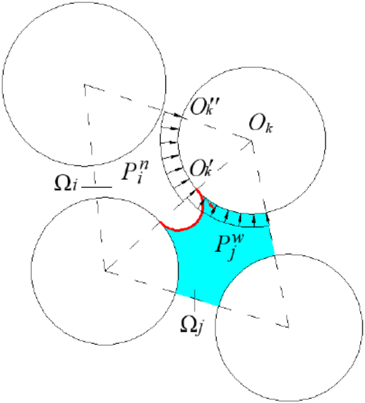
W-phase not trapped

Capillary force and deformation(In progress)

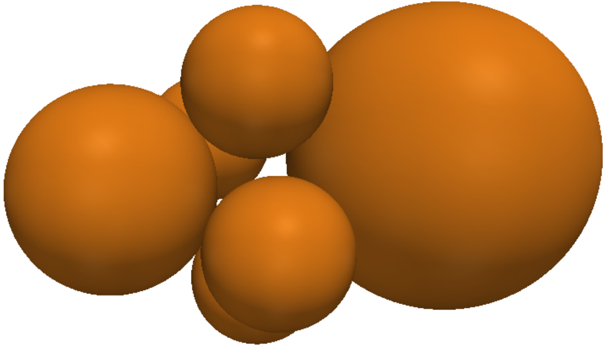
The total force F^k generated on particle k includes the effects of W-phase P^w, NW-phase P^n and NW-W interface tension σ^{nw}.

$$F^k = F^{n,k} + F^{w,k} + F^{\sigma,k}$$

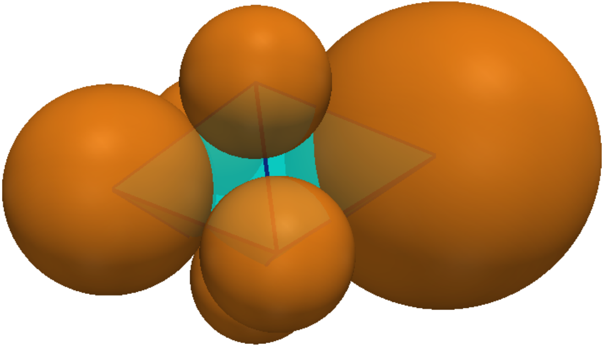




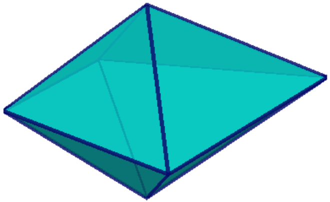
Capillary force and deformation



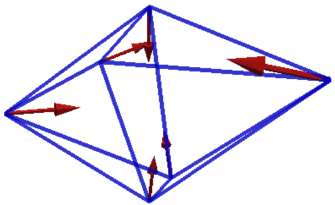
Capillary force and deformation



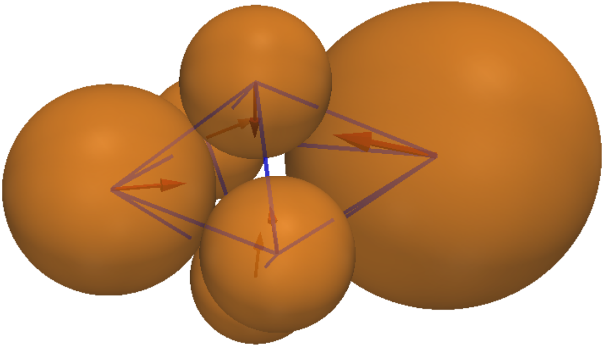
Capillary force and deformation



Capillary force and deformation

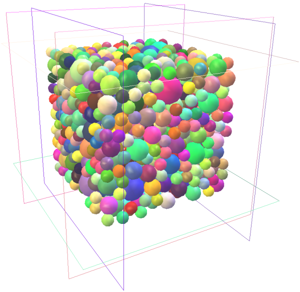


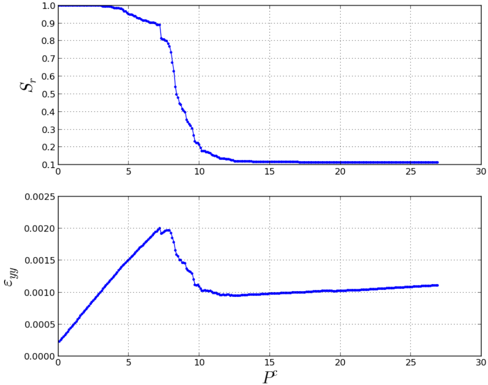
Capillary force and deformation



Capillary force and deformation

Shrinking/Swelling during drainage (oedometer conditions)





Conclusions

- Define pore geometry and determine P_e^c based on PFV scheme.
- Implement different drainage modes based on different assumptions.
- Application and comparison.(Side boundary, REV, residual saturation, P^c...)
- Simulate the deformation of spheres packing. (In progress)

Questions

Thank you for your attention.

Bibliography

B. Chareyre, A. Cortis, E. Catalano, and E. Barthélemy, "Pore-scale modeling of viscous flow and induced forces in dense sphere packings," *Transport in Porous Media*, vol. 92, pp. 473–493, 2012.

S. Ma, G. Mason, and N. Morrow, "Effect of contact angle on drainage and imbibition in regular polygonal tubes," *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, vol. 117, no. 3, pp. 273 – 291, 1996.

R. Mayer and R. Stowe, "Mercury porosimetrybreakthrough pressure for penetration between packed spheres," *Journal of Colloid Science*, vol. 20, no. 8, pp. 893 – 911, 1965.

H. Princen, "Capillary phenomena in assemblies of parallel cylinders: Ii. capillary rise in systems with more than two cylinders," *Journal of Colloid and Interface Science*, vol. 30, no. 3, pp. 359 – 371, 1969.

V. Joekar-Niasar, M. Prodanović, D. Wildenschild, and S. Hassanizadeh, "Network model investigation of interfacial area, capillary pressure and saturation relationships in granular porous media," *Water Resource Research*, vol. 46, 2010. [Online]. Available: http://dx.doi.org/10.1029/2009WR008585