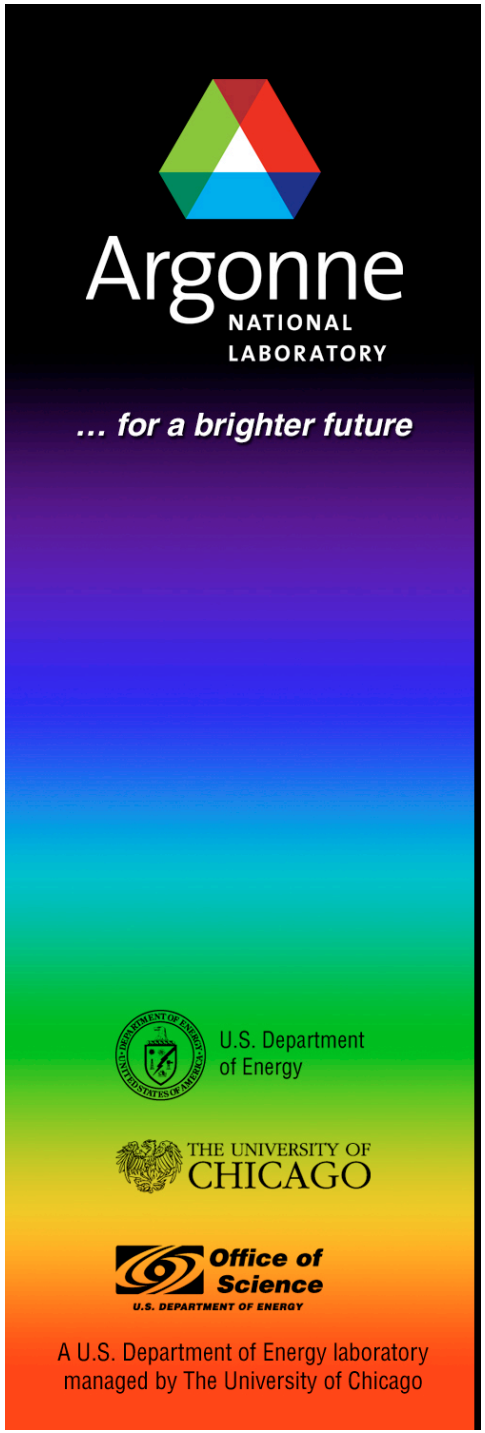



# An iterative solver for cone complementarity problems of nonsmooth multibody dynamics—and other DVI

*Mihai ANITESCU ,*

*Argonne National Laboratory*

*With D. Negrut (Wisconsin) and A. Tasora (Parma)*

*SIAM CSE , 2009*

*MIAMI*

# Nonsmooth contact dynamics—what is it?

- Differential problem with variational inequality constraints – DVI

*Newton Equations*

*Non-Penetration Constraints*

$$M\frac{dv}{dt} = \sum_{j=1,2,\ldots,p}\left(c_n^{(j)}n^{(j)} + \beta_1^{(j)}t_1^{(j)} + \beta_2^{(j)}t_2^{(j)}\right) + f_c(q,v) + k(t,q,v)$$

$$\frac{dq}{dt} = \Gamma(q)v$$

*Generalized Velocities*

$$c_n^{(j)} \geq 0 \quad \perp \quad \Phi^{(j)}(q) \geq 0, \qquad j = 1,2,\ldots,p$$

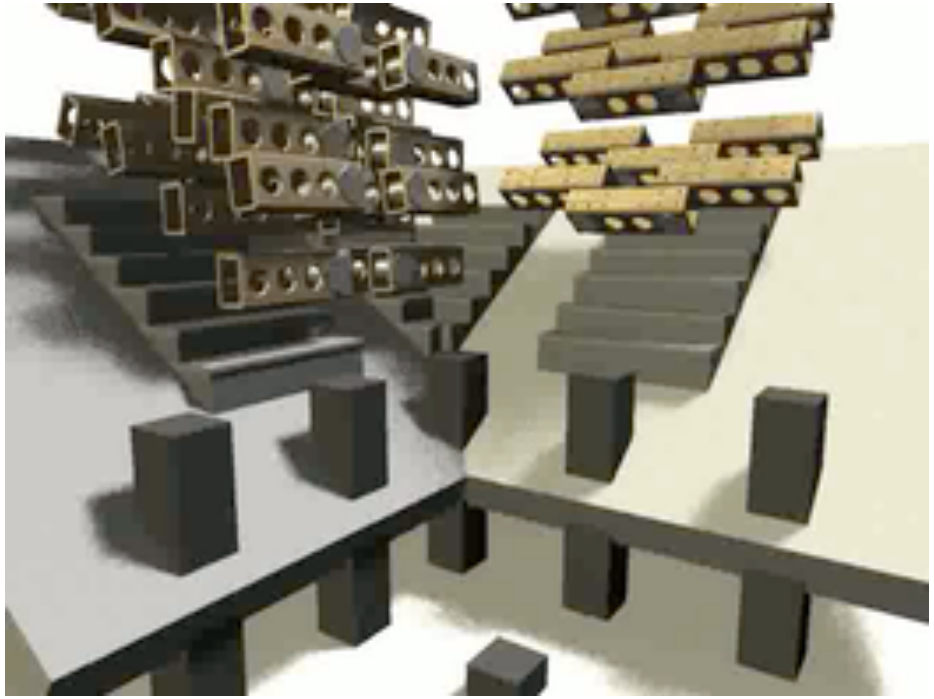
$$\left(\beta_1^{(j)}, \beta_2^{(j)}\right) = \operatorname{argmin}_{\mu^{(j)}c_n^{(j)} \geq \sqrt{\left(\beta_1^{(j)}+\beta_2^{(j)}\right)^2}} \left[\left(v^T t_1^{(j)}\right)\beta_1 + \left(v^T t_2^{(j)}\right)\beta_2\right]$$

*Friction Model*

- Truly, a Differential Problem with Equilibrium Constraints

# *Contact dynamics applications*

- Granular Flow,
- Masonry Stability,
- Rock Dynamics.
- Agent-Based Modeling (Pedestrian Evacuation Dynamics).
- Physics-based graphics simulations.
- Most common approaches by far are smoothing approaches (DEM)



# Question 1: Should we do smoothing?

- Recall, DVI (for C=R+)

$$\dot{x} = f\left(t, x(t), u(t)\right);$$
$$u \geq 0 \perp F\left(t, x(t), u(t)\right) \geq 0$$

- Smoothing

$$\dot{x} = f\left(t, x(t), u(t)\right);$$
$$u_i \; F_i\left(t, x(t), u(t)\right) = \varepsilon, \quad i = 1, 2, \ldots n_u$$

- Followed by forward Euler. Easy to implement!!

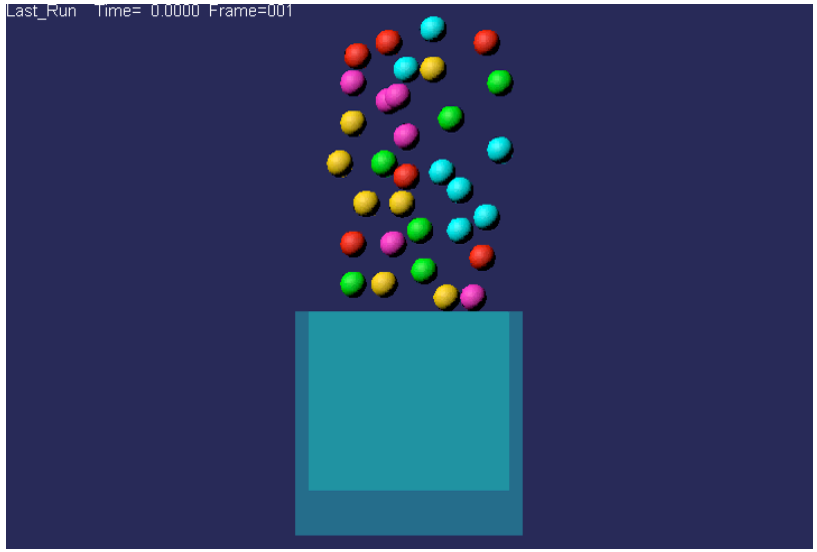
$$u_i^n \; F_i\left(t^{n-1}, x^{n-1}, u^{n-1}\right) = \varepsilon, \quad i = 1, 2, \ldots n_u$$
$$x^{n+1} = x^n + hf\left(t^n, x^n, u^n\right);$$

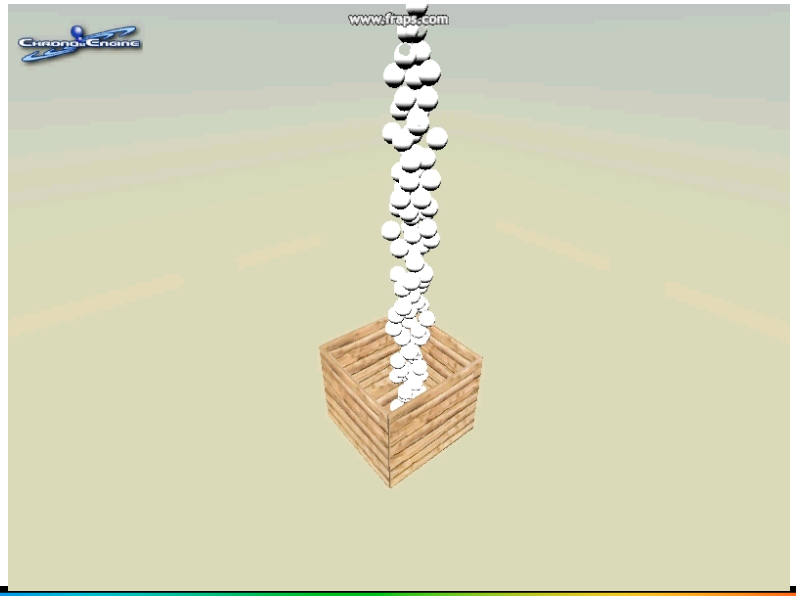
- Compare with the complexity of time-stepping
- But does it give good results?

$$x^{n+1} = x^n + hf\left(t^{n+1}, x^{n+1}, u^{n+1}\right);$$
$$u^{n+1} \geq 0 \perp F\left(t^{n+1}, x^{n+1}, u^{n+1}\right) \geq 0$$

# *Applying ADAMS to granular flow*

- ADAMS is the workhorse of engineering dynamics.
- ADAMS/View Procedure for simulating.
- Spheres: diameter of 60 mm and a weight of 0.882 kg.
- Forces:smoothing with stiffness of 1E5, force exponent of 2.2, damping coefficient of 10.0, and a penetration depth of 0.1



# ADAMS versus ChronoEngine

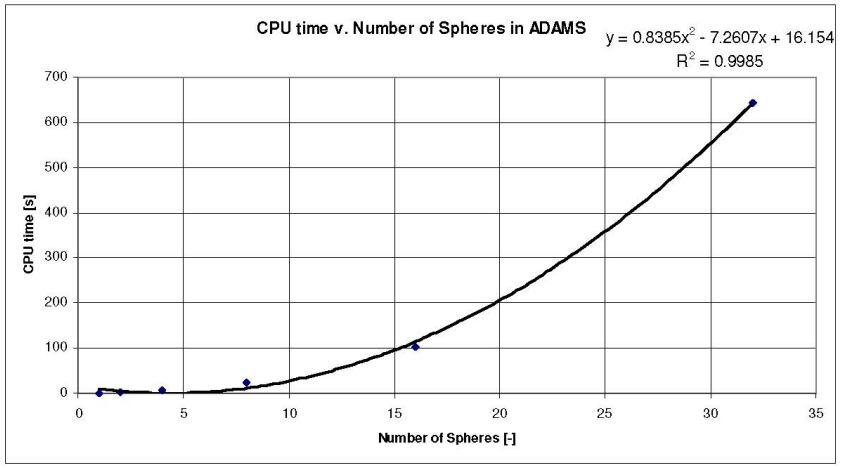
Table 1: Number of rigid bodies v. CPU time in ADAMS

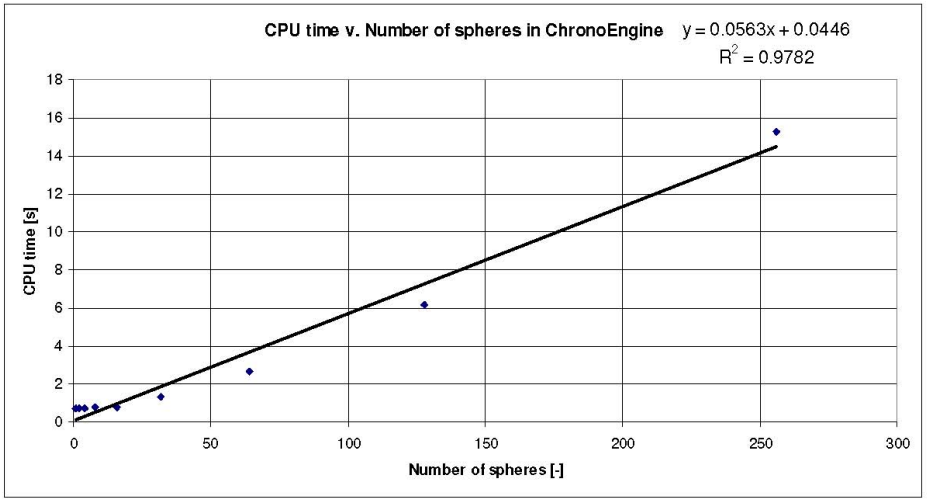
| Number of Spheres | Max Number of Mutual Contacts [-] | CPU time (seconds) |
|---|---|---|
| 1 | 1 | 0.41 |
| 2 | 3 | 3.3 |
| 4 | 14 | 7.75 |
| 8 | 44 | 25.36 |
| 16 | 152 | 102.78 |
| 32 | 560 | 644.4 |

The following graph shows the nonlinear increase in the CPU time as the number of colliding bodies increases.

Table 2: Number of rigid bodies v. CPU time in ChronoEngine

| Number of Spheres | Max Number of Mutual Contacts [-] | CPU time (seconds) |
|---|---|---|
| 1 | 1 | 0.70 |
| 2 | 3 | 0.73 |
| 4 | 14 | 0.73 |
| 8 | 44 | 0.76 |
| 16 | 152 | 0.82 |
| 32 | 560 | 1.32 |
| 64 | 2144 | 2.65 |
| 128 | 8384 | 6.17 |
| 256 | 33152 | 15.30 |

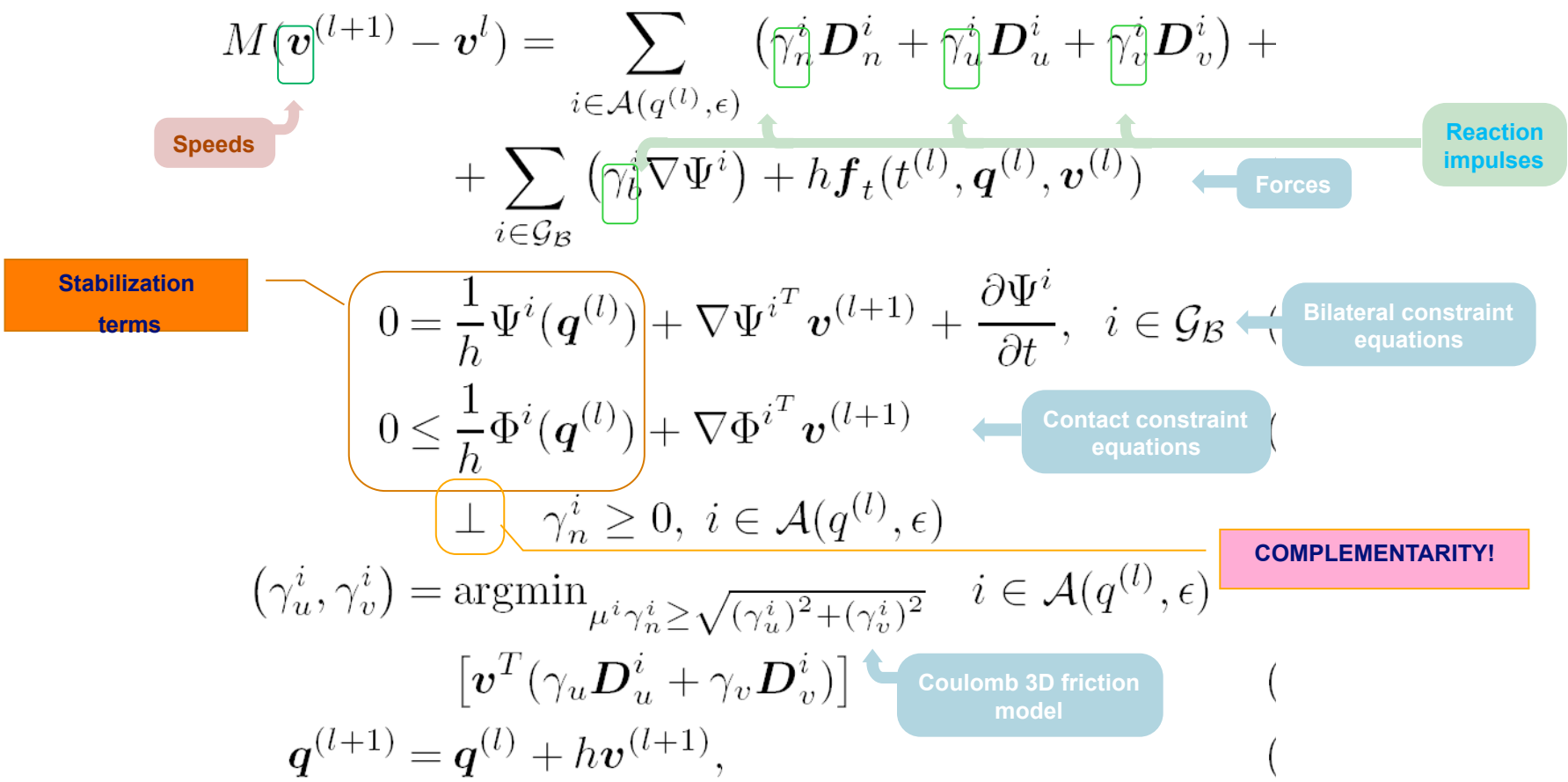



*Conclusion 1: Often, time stepping is more promising,*

# Time stepping scheme -- original

- A measure differential inclusion solution can be obtained by time-stepping (Stewart, 1998, Anitescu 2006)




$$M(\boldsymbol{v}^{(l+1)} - \boldsymbol{v}^{l}) = \sum_{i \in \mathcal{A}(q^{(l)}, \epsilon)} (\gamma_n^i \boldsymbol{D}_n^i + \gamma_u^i \boldsymbol{D}_u^i + \gamma_v^i \boldsymbol{D}_v^i) +$$

$$+ \sum_{i \in \mathcal{G}_\mathcal{B}} (\gamma_b^i \nabla \Psi^i) + h \boldsymbol{f}_t(t^{(l)}, \boldsymbol{q}^{(l)}, \boldsymbol{v}^{(l)})$$

$$0 = \frac{1}{h}\Psi^i(\boldsymbol{q}^{(l)}) + \nabla \Psi^{i^T} \boldsymbol{v}^{(l+1)} + \frac{\partial \Psi^i}{\partial t}, \quad i \in \mathcal{G}_\mathcal{B}$$

$$0 \leq \frac{1}{h}\Phi^i(\boldsymbol{q}^{(l)}) + \nabla \Phi^{i^T} \boldsymbol{v}^{(l+1)}$$

$$\perp \quad \gamma_n^i \geq 0, \ i \in \mathcal{A}(q^{(l)}, \epsilon)$$

$$(\gamma_u^i, \gamma_v^i) = \operatorname{argmin}_{\mu^i \gamma_n^i \geq \sqrt{(\gamma_u^i)^2 + (\gamma_v^i)^2}} \quad i \in \mathcal{A}(q^{(l)}, \epsilon)$$

$$\left[\boldsymbol{v}^T(\gamma_u \boldsymbol{D}_u^i + \gamma_v \boldsymbol{D}_v^i)\right]$$

$$\boldsymbol{q}^{(l+1)} = \boldsymbol{q}^{(l)} + h\boldsymbol{v}^{(l+1)},$$

# Time Stepping -- Convex Relaxation

- A modification (relaxation, to get convex QP with conic constraints):

$$M(\boldsymbol{v}^{(l+1)} - \boldsymbol{v}^l) = \sum_{i \in \mathcal{A}(q^{(l)}, \epsilon)} \left(\gamma_n^i \boldsymbol{D}_n^i + \gamma_u^i \boldsymbol{D}_u^i + \gamma_v^i \boldsymbol{D}_v^i\right) + \sum_{i \in \mathcal{G}_\mathcal{B}} \left(\gamma_b^i \nabla \Psi^i\right) + h \boldsymbol{f}_t(t^{(l)}, \boldsymbol{q}^{(l)}, \boldsymbol{v}^{(l)})$$

$$0 = \frac{1}{h}\Psi^i(\boldsymbol{q}^{(l)}) + \nabla \Psi^{i^T} \boldsymbol{v}^{(l+1)} + \frac{\partial \Psi^i}{\partial t}, \quad i \in \mathcal{G}_\mathcal{B}$$

$$0 \leq \frac{1}{h}\Phi^i(\boldsymbol{q}^{(l)}) + \nabla \Phi^{i^T} \boldsymbol{v}^{(l+1)} - \mu^i \sqrt{(\boldsymbol{D}_u^{i,T}\boldsymbol{v})^2 + (\boldsymbol{D}_v^{i,T}\boldsymbol{v})^2}$$

$$\perp \quad \gamma_n^i \geq 0, \; i \in \mathcal{A}(q^{(l)}, \epsilon)$$

$$(\gamma_u^i, \gamma_v^i) = \operatorname{argmin}_{\mu^i \gamma_n^i \geq \sqrt{(\gamma_u^i)^2 + (\gamma_v^i)^2}} \quad i \in \mathcal{A}(q^{(l)}, \epsilon)$$
$$\left[\boldsymbol{v}^T(\gamma_u \boldsymbol{D}_u^i + \gamma_v \boldsymbol{D}_v^i)\right]$$

$$\boldsymbol{q}^{(l+1)} = \boldsymbol{q}^{(l)} + h\boldsymbol{v}^{(l+1)},$$

*(For small $\mu$ and/or small speeds, almost no one-step differences from the Coulomb theory)*

*But In any case, converges to same MDI as unrelaxed scheme.*

*[ see M.Anitescu, "Optimization Based Simulation of Nonsmooth Rigid Body Dynamics" ]*

# Cone complementarity

- Aiming at a more compact formulation:

$$\boldsymbol{b}_{\mathcal{A}} = \left\{ \frac{1}{h}\Phi^{i_1}, 0, 0, \frac{1}{h}\Phi^{i_2}, 0, 0, \ldots, \frac{1}{h}\Phi^{i_{n_{\mathcal{A}}}}, 0, 0 \right\}$$

$$\boldsymbol{\gamma}_{\mathcal{A}} = \left\{ \gamma_n^{i_1}, \gamma_u^{i_1}, \gamma_v^{i_1}, \gamma_n^{i_2}, \gamma_u^{i_2}, \gamma_v^{i_2}, \ldots, \gamma_n^{i_{n_{\mathcal{A}}}}, \gamma_u^{i_{n_{\mathcal{A}}}}, \gamma_v^{i_{n_{\mathcal{A}}}} \right\}$$

$$\boldsymbol{b}_{\mathcal{B}} = \left\{ \frac{1}{h}\Psi^1 + \frac{\partial \Psi^1}{\partial t}, \frac{1}{h}\Psi^2 + \frac{\partial \Psi^2}{\partial t}, \ldots, \frac{1}{h}\Psi^{n_{\mathcal{B}}} + \frac{\partial \Psi^{n_{\mathcal{B}}}}{\partial t} \right\}$$

$$\boldsymbol{\gamma}_{\mathcal{B}} = \left\{ \gamma_b^1, \gamma_b^2, \ldots, \gamma_b^{n_{\mathcal{B}}} \right\}$$

$$D_{\mathcal{A}} = \left[ D^{i_1} | D^{i_2} | \ldots | D^{i_{n_{\mathcal{A}}}} \right], \quad i \in \mathcal{A}(\boldsymbol{q}^l, \epsilon) \qquad D^i = \left[ \boldsymbol{D}_n^i | \boldsymbol{D}_u^i | \boldsymbol{D}_v^i \right]$$

- $D_{\mathcal{B}} = \left[ \nabla\Psi^{i_1} | \nabla\Psi^{i_2} | \ldots | \nabla\Psi^{i_{n_{\mathcal{B}}}} \right], \quad i \in \mathcal{G}_{\mathcal{B}}$

$$\boldsymbol{b}_{\mathcal{E}} \in \mathbb{R}^{n_{\mathcal{E}}} = \{ \boldsymbol{b}_{\mathcal{A}}, \boldsymbol{b}_{\mathcal{B}} \}$$

$$\boldsymbol{\gamma}_{\mathcal{E}} \in \mathbb{R}^{n_{\mathcal{E}}} = \{ \boldsymbol{\gamma}_{\mathcal{A}}, \boldsymbol{\gamma}_{\mathcal{B}} \}$$

$$D_{\mathcal{E}} = [ D_{\mathcal{A}} | D_{\mathcal{B}} ]$$

# Cone complementarity problem at each step

- Also define:

$$\tilde{\boldsymbol{k}}^{(l)} = M\boldsymbol{v}^{(l)} + h\boldsymbol{f}_t(t^{(l)}, \boldsymbol{q}^{(l)}, \boldsymbol{v}^{(l)})$$
$$N = D_{\mathcal{E}}^T M^{-1} D_{\mathcal{E}}$$
$$\boldsymbol{r} = D_{\mathcal{E}}^T M^{-1} \tilde{\boldsymbol{k}} + \boldsymbol{b}_{\mathcal{E}}$$

- Then:

$$\begin{aligned}
M(\boldsymbol{v}^{(l+1)} - \boldsymbol{v}^l) = & \sum_{i \in \mathcal{A}(q^{(l)}, \epsilon)} (\gamma_n^i \boldsymbol{D}_n^i + \gamma_u^i \boldsymbol{D}_u^i + \gamma_v^i \boldsymbol{D}_v^i) + \\
& + \sum_{i \in \mathcal{G}_{\mathcal{B}}} (\gamma_b^i \nabla \Psi^i) + h\boldsymbol{f}_t(t^{(l)}, \boldsymbol{q}^{(l)}, \boldsymbol{v}^{(l)}) \\
0 = & \frac{1}{h}\Psi^i(\boldsymbol{q}^{(l)}) + \nabla {\Psi^i}^T \boldsymbol{v}^{(l+1)} + \frac{\partial \Psi^i}{\partial t}, \quad i \in \mathcal{G}_{\mathcal{B}} \\
0 \leq & \frac{1}{h}\Phi^i(\boldsymbol{q}^{(l)}) + \nabla {\Phi^i}^T \boldsymbol{v}^{(l+1)} \\
& \perp \quad \gamma_n^i \geq 0, \; i \in \mathcal{A}(q^{(l)}, \epsilon) \\
(\gamma_u^i, \gamma_v^i) = & \operatorname{argmin}_{\mu^i \gamma_n^i \geq \sqrt{(\gamma_u^i)^2 + (\gamma_v^i)^2}} \quad i \in \mathcal{A}(q^{(l)}, \epsilon) \\
& [\boldsymbol{v}^T(\gamma_u \boldsymbol{D}_u^i + \gamma_v \boldsymbol{D}_v^i)]
\end{aligned}$$

becomes..

$$(N\boldsymbol{\gamma}_{\mathcal{E}} + \boldsymbol{r}) \in -\Upsilon^{\circ} \quad \perp \quad \boldsymbol{\gamma}_{\mathcal{E}} \in \Upsilon$$

This is a CCP,
**CONE COMPLEMENTARITY PROBLEM**

# Cone complementarity—Decomposable cones.

- Here we introduced the convex cone

$$\Upsilon = \left( \bigoplus_{i \in \mathcal{A}(\boldsymbol{q}^l, \epsilon)} \mathcal{FC}^i \right) \bigoplus \left( \bigoplus_{i \in \mathcal{G}_\mathcal{B}} \mathcal{BC}^i \right)$$

$\mathcal{FC}^i$ In R^3 is $i$-th friction cone

$\mathcal{BC}^i$ is R

- ..and its polar cone:

$$\Upsilon^\circ = \left( \bigoplus_{i \in \mathcal{A}(\boldsymbol{q}^l, \epsilon)} \mathcal{FC}^{i\circ} \right) \bigoplus \left( \bigoplus_{i \in \mathcal{G}_\mathcal{B}} \mathcal{BC}^{i\circ} \right)$$

CCP: $(N\boldsymbol{\gamma}_\mathcal{E} + \boldsymbol{r}) \in -\Upsilon^\circ \quad \perp \quad \boldsymbol{\gamma}_\mathcal{E} \in \Upsilon$

# General: The iterative method

- Question 3: How to efficiently solve the Cone Complementarity Problem for large-scale systems?

$$(N\boldsymbol{\gamma}_{\mathcal{E}} + \boldsymbol{r}) \in -\Upsilon^{\circ} \quad \perp \quad \boldsymbol{\gamma}_{\mathcal{E}} \in \Upsilon$$

- Our method: use a fixed-point iteration

$$\boldsymbol{\gamma}^{r+1} = \lambda \Pi_{\Upsilon} \left( \boldsymbol{\gamma}^{r} - \omega B^{r} \left( N\boldsymbol{\gamma}^{r} + \boldsymbol{r} + K^{r} \left( \boldsymbol{\gamma}^{r+1} - \boldsymbol{\gamma}^{r} \right) \right) \right) + (1 - \lambda)\, \boldsymbol{\gamma}^{r}$$

- with matrices:
- ..and a non-extensive orthogonal projection operator onto feasible set

$$B^{r} = \begin{bmatrix} \eta_1 I_{n_1} & 0 & \cdots & 0 \\ 0 & \eta_2 I_{n_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \eta_{n_k} I_{n_{n_k}} \end{bmatrix}$$

$$N^{T} = \begin{bmatrix} 0 & K_{12} & K_{13} & \cdots & K_{1n_k} \\ 0 & 0 & K_{23} & \cdots & K_{2n_k} \\ 0 & 0 & 0 & \cdots & K_{3n_k} \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ 0 & 0 & 0 & & 0 \end{bmatrix}$$

$$\Pi_{\Upsilon} : \mathbb{R}^{n\varepsilon} \rightarrow \mathbb{R}^{n\varepsilon}$$

# General: The iterative method

- ASSUMPTIONS

A1 The matrix $N$ of the problem (CCP) is symmetric and positive semi-definite.

A2 There exists a positive number, $\alpha > 0$ such that, at any iteration $r$, $r = 0, 1, 2, \ldots$, we have that $B^r \succ \alpha I$

A3 There exists a positive number, $\beta > 0$ such that, at any iteration $r$, $r = 0, 1, 2, \ldots$, we have that $(x^{r+1} - x^r)^T \left( (\lambda \omega B^r)^{-1} + K^r - \frac{N}{2} \right) (x^{r+1} - x^r) \geq \beta \left\| x^{r+1} - x^r \right\|^2$.

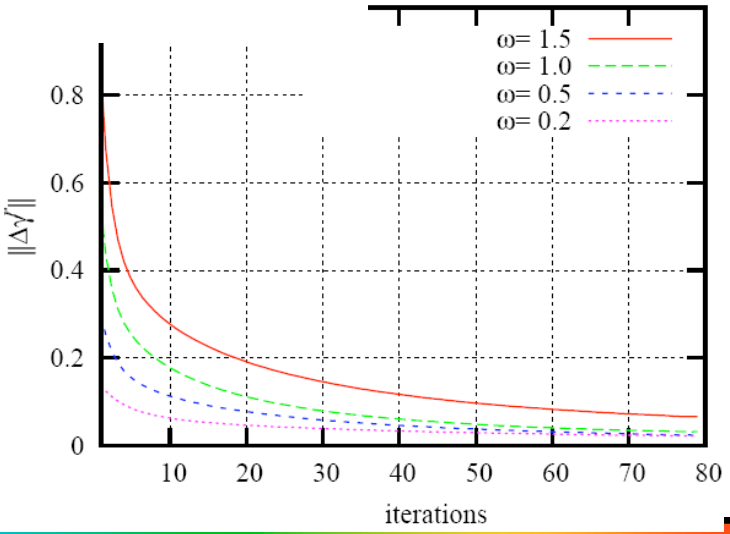
Always satisfied in multibody systems

Essentially free choice, we use identity blocks

Use ω overrelaxation factor to adjust this

- Under the above assumptions, we can prove THEOREMS about convergence.

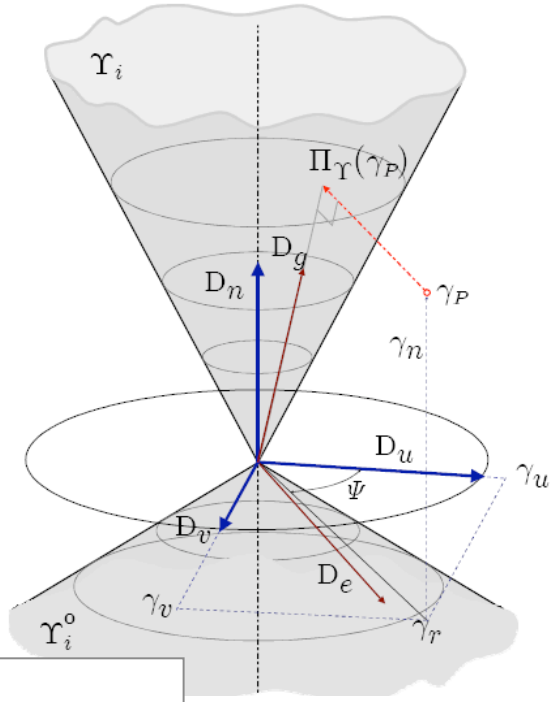
- The method produces a bounded sequence with an unique accumulation point.



# The projection operator is easy and separable

- *For each frictional contact constraint:*

$$\Pi_\Upsilon = \left\{ \Pi_{\Upsilon_1}(\boldsymbol{\gamma}_1)^T, \ldots \Pi_{\Upsilon_{n_\mathcal{A}}}(\boldsymbol{\gamma}^{n_\mathcal{A}})^T, \Pi_b^1(\gamma_b^1), \ldots, \Pi_b^{n_\mathcal{B}}(\gamma_b^{n_\mathcal{B}}) \right\}^T$$



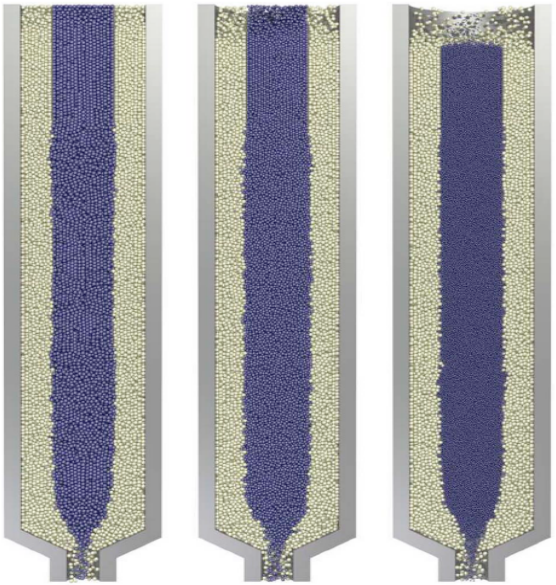

- *For each bilateral constraint, simply do nothing.*
- The complete operator:

$$\forall i \in \mathcal{A}(\boldsymbol{q}^{(l)}, \epsilon)$$

$$\begin{aligned}
&\gamma_r < \mu_i \gamma_n && \Pi_i = \boldsymbol{\gamma}_i \\
&\gamma_r < -\frac{1}{\mu_i}\gamma_n && \Pi_i = \{0,0,0\} \\
&\gamma_r > \mu_i \gamma_n \wedge \gamma_r > -\frac{1}{\mu_i}\gamma_n && \Pi_{i,n} = \frac{\gamma_r \mu_i + \gamma_n}{\mu_i^2 + 1} \\
& && \Pi_{i,u} = \gamma_u \frac{\mu_i \Pi_{i,n}}{\gamma_r} \\
& && \Pi_{i,v} = \gamma_v \frac{\mu_i \Pi_{i,n}}{\gamma_r}
\end{aligned}$$

# Simulating the PBR nuclear reactor

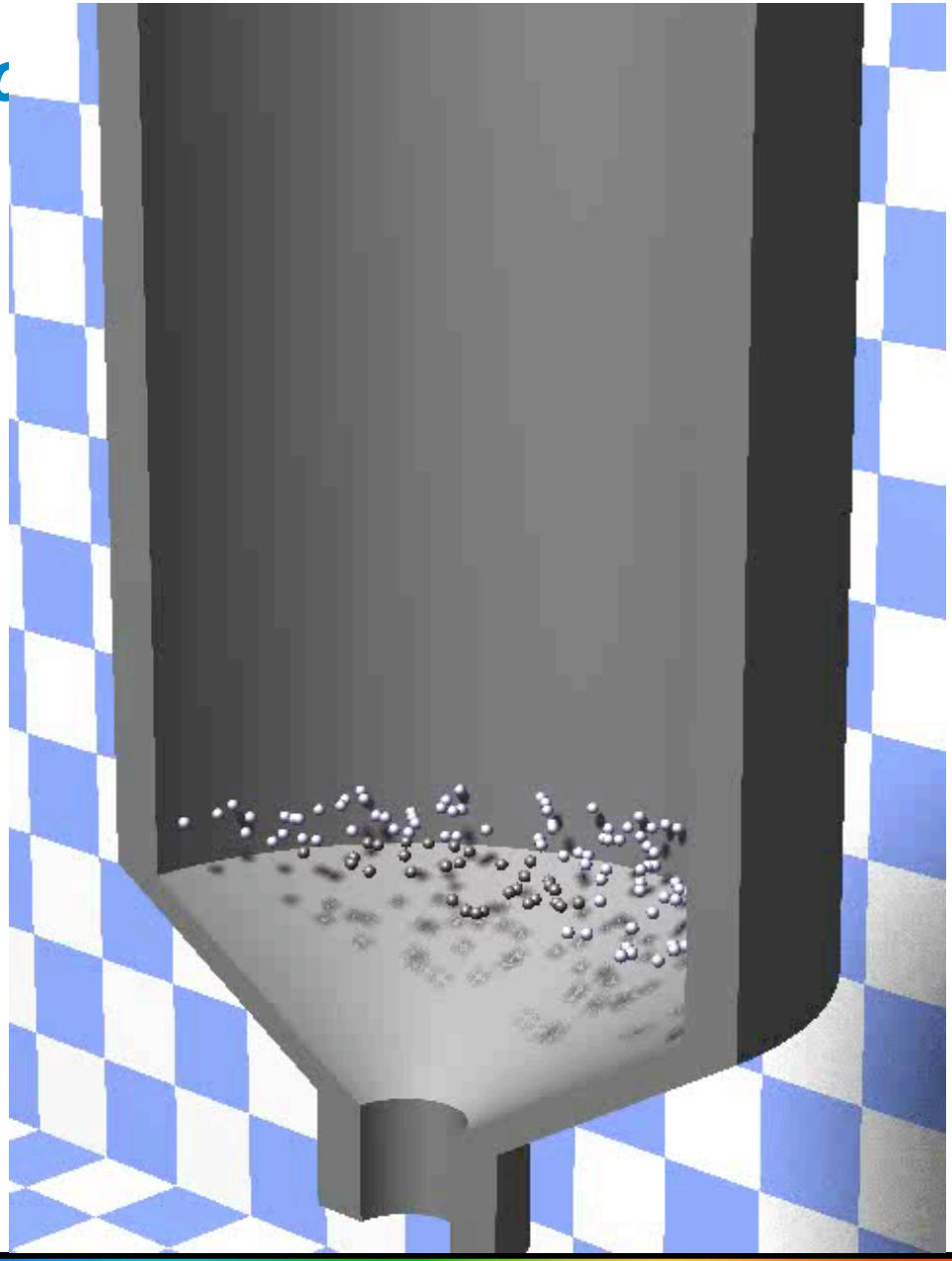
- Problem of bidisperse granular flow with dense packing.
- Previous attempts: DEM methods on supercomputers at Sandia Labs regularization)
- 40 seconds of simulation for 440,000 pebbles needs 1 week on 64 processors dedicated cluster (Rycroft et al.)

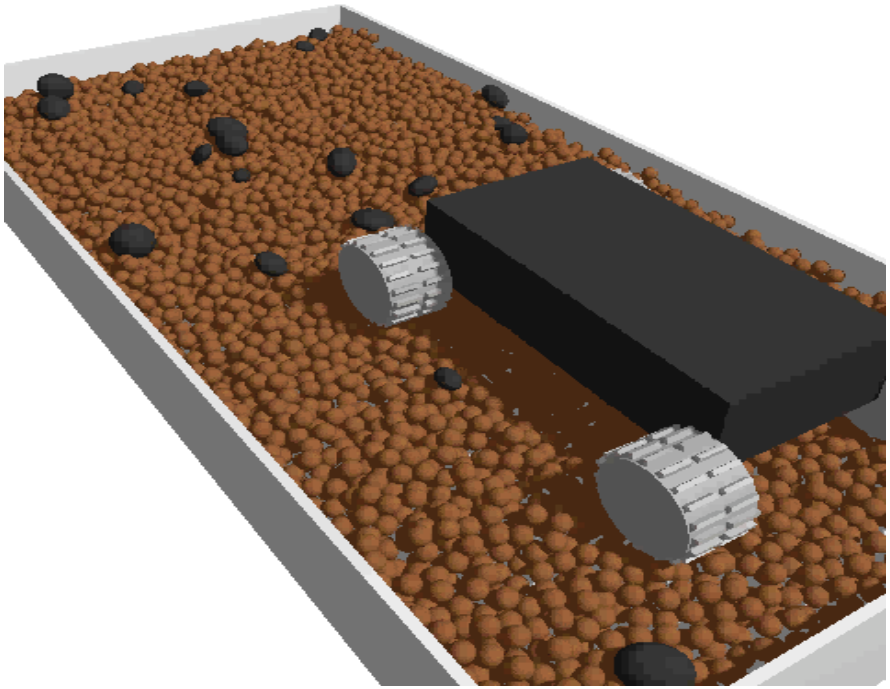
model a frictionless wall, $\mu_w = 0.0$. For the current simulations we set $k_t = \frac{2}{7} k_n$ and choose $k_n = 2 \times 10^5$ $gm/d$. While this is significantly less than would be realistic for graphite pebbles, where we expect $k_n > 10^{10}$ $gm/d$, such a spring constant would be prohibitively computationally expensive, as the time step scales as $\delta t \propto k_n^{-1/2}$ for collisions to be modeled effectively. Previous simulations have shown that

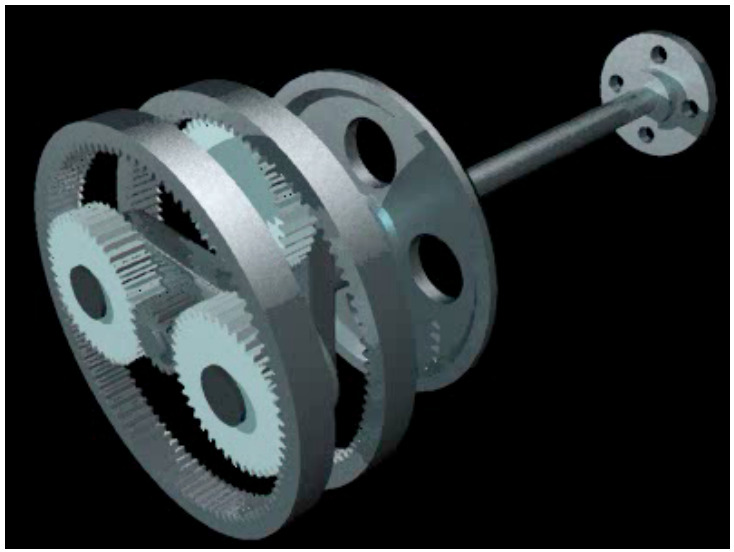
# *Simulating the PBR nuc*

- 160’000 Uranium-Graphite spheres, 600’000 contacts on average
- Two millions of primal variables, six millions of dual variables
- *1 day on a Windows station…*
- But we are limited by the 2GB user mode limit, 64 bit port in progress—but linear scaling..
- We estimate 3CPU days, compare with 450 CPU days for an incomplete solution in 2006 !!!
- Answer 3: Our approach is efficient for large scale!!



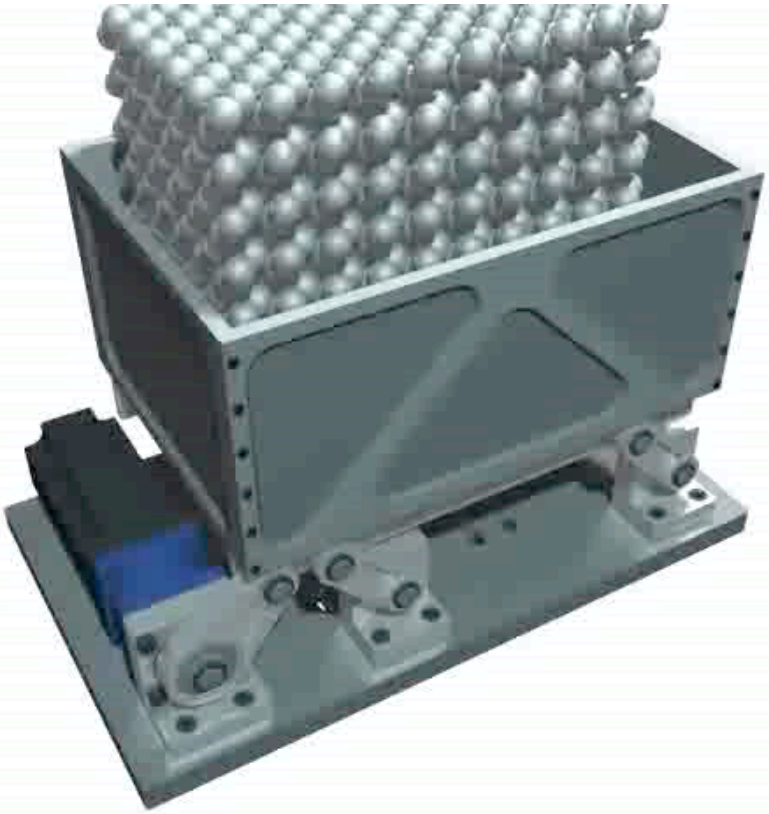
## *In addition, we can approach efficiently approach many engineering problems (see website for papers)*





# *Scaling/constraint accuracy test:*

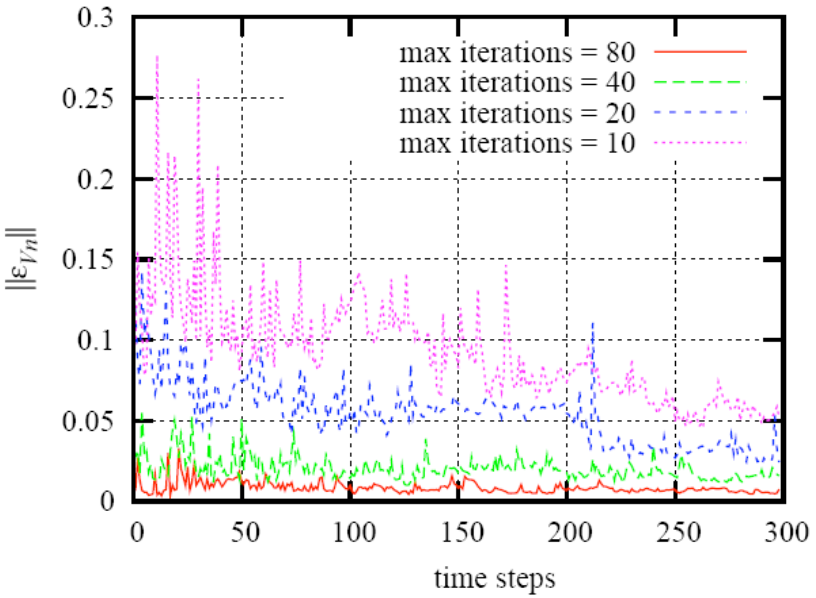
- Size-segregation in shaker, with thousands of steel spheres



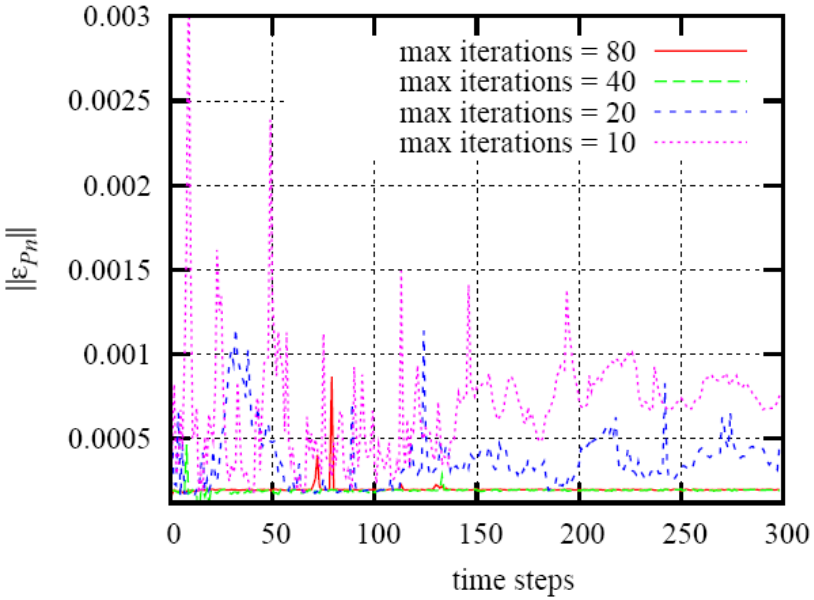
Note: solution beyond reach of Lemke-type LCP solvers!

# Tests

- Feasibility accuracy increases with number of iterations:




Speed violation in constraints

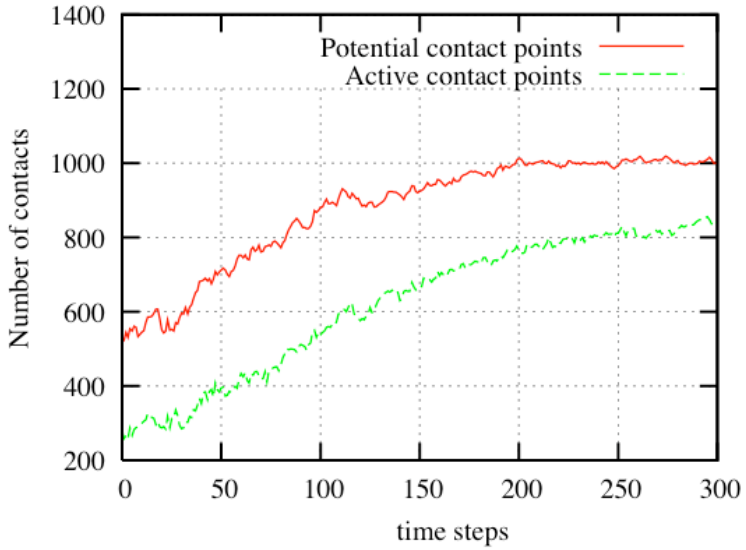



Position error in constraints (penetration)

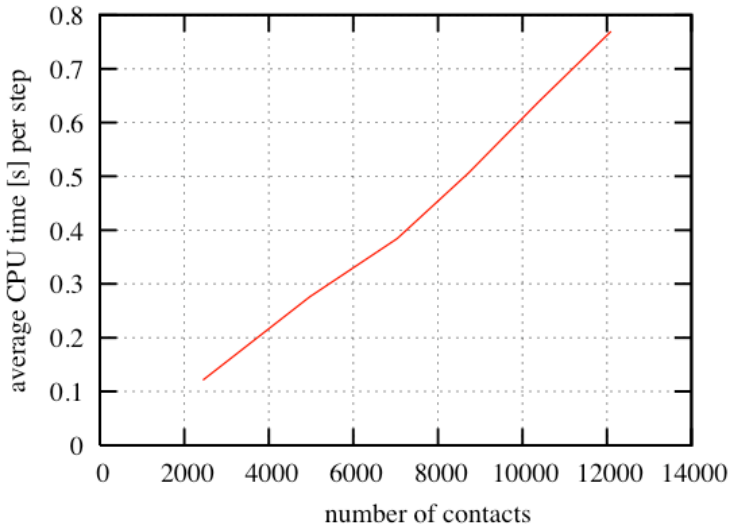
*(with example of 300 spheres in shaker)*

# Tests: Scalability

- CPU effort per contact, since our contacts are the problem variables.
- Penetration error was uniformly no larger than 0.2% of diameter.

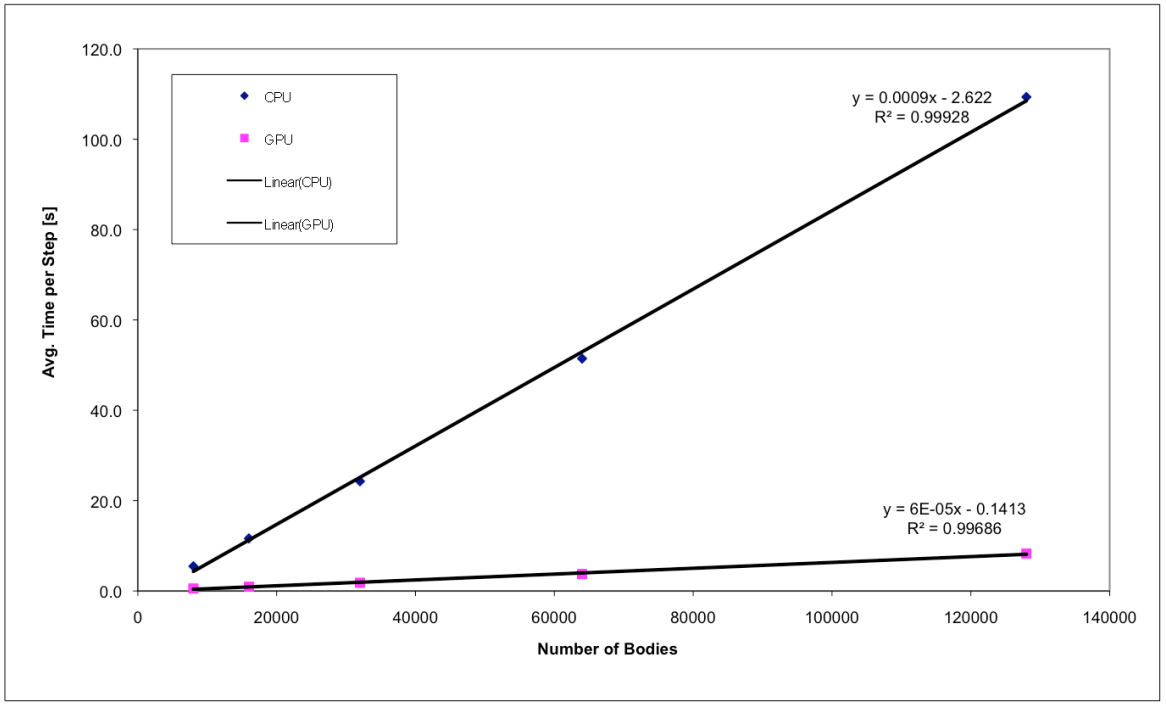



Number of contacts in time, 300 spheres




CPU time per step for 300-1500 spheres

# Preliminary Results for GS on large problems on GPU

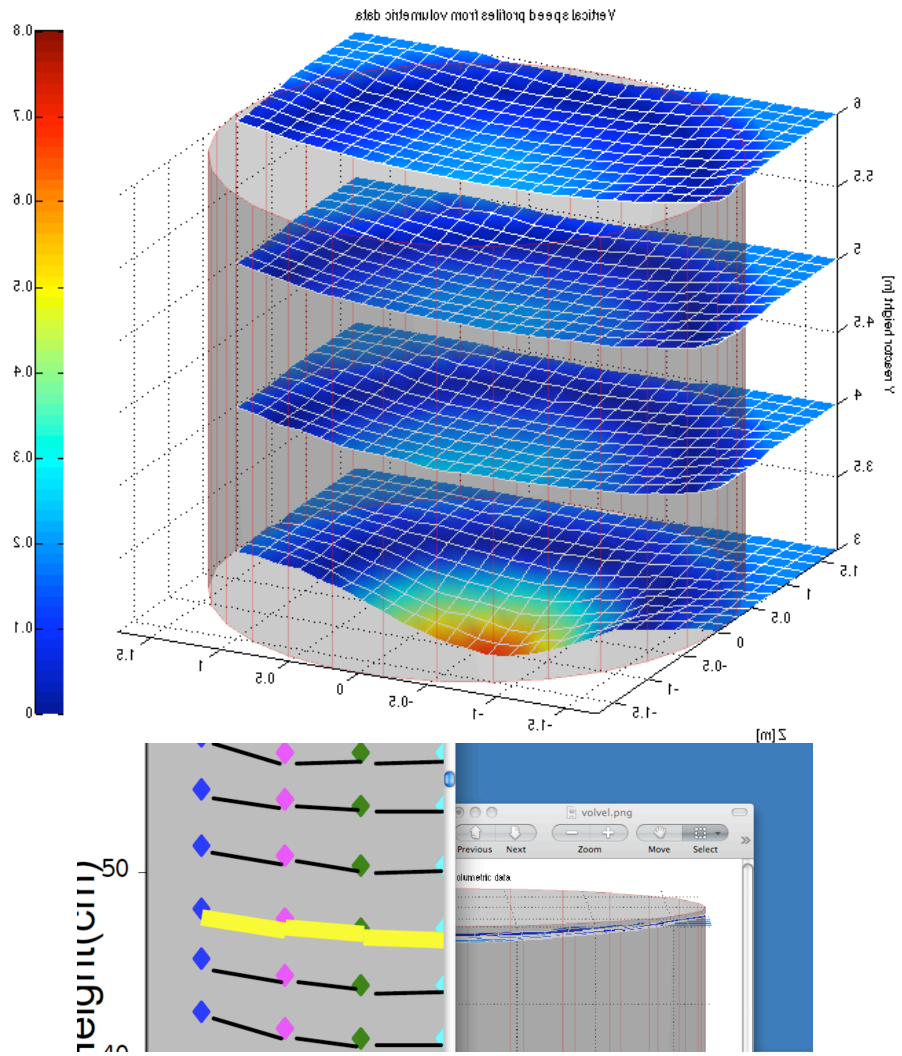


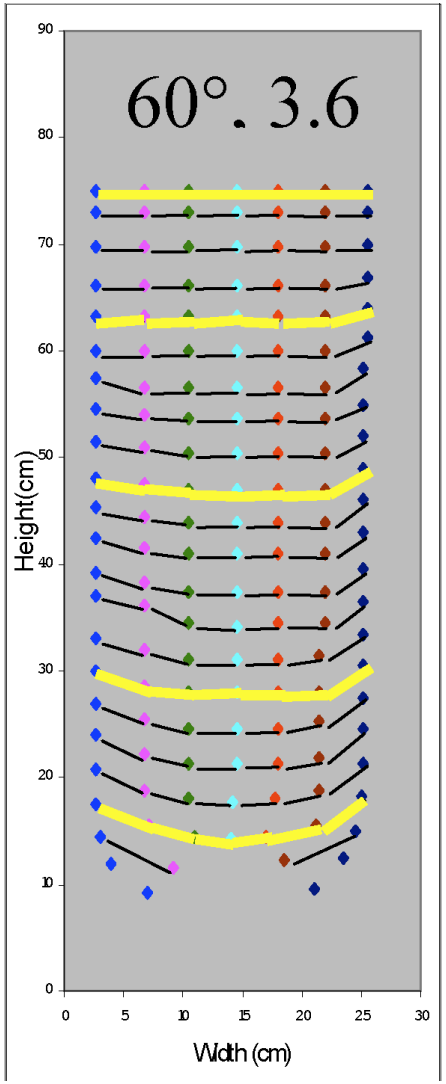

| | CPU | | | GPU | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Number of Bodies | CCP Time | CD Time | Step Time | CCP Time | CD Time | Step Time | CCP Speedup | CD Speedup | Step Speedup |
| 8000 | 5.26725 | 0.10655 | 5.47370 | 0.39600 | 0.12288 | 0.54658 | 13.3010 | 0.8671 | 10.0144 |
| 16000 | 11.18999 | 0.23601 | 11.63754 | 0.74096 | 0.15337 | 0.95315 | 15.1020 | 1.5388 | 12.2095 |
| 32000 | 23.28647 | 0.55861 | 24.28257 | 1.46889 | 0.21752 | 1.80878 | 15.8531 | 2.5680 | 13.4249 |
| 64000 | 49.16970 | 1.36559 | 51.44442 | 3.12925 | 0.36476 | 3.75152 | 15.7130 | 3.7438 | 13.7130 |
| 128000 | 103.66658 | 3.80176 | 109.34287 | 6.97682 | 0.74488 | 8.27050 | 14.8587 | 5.1038 | 13.2208 |

# *Future work*

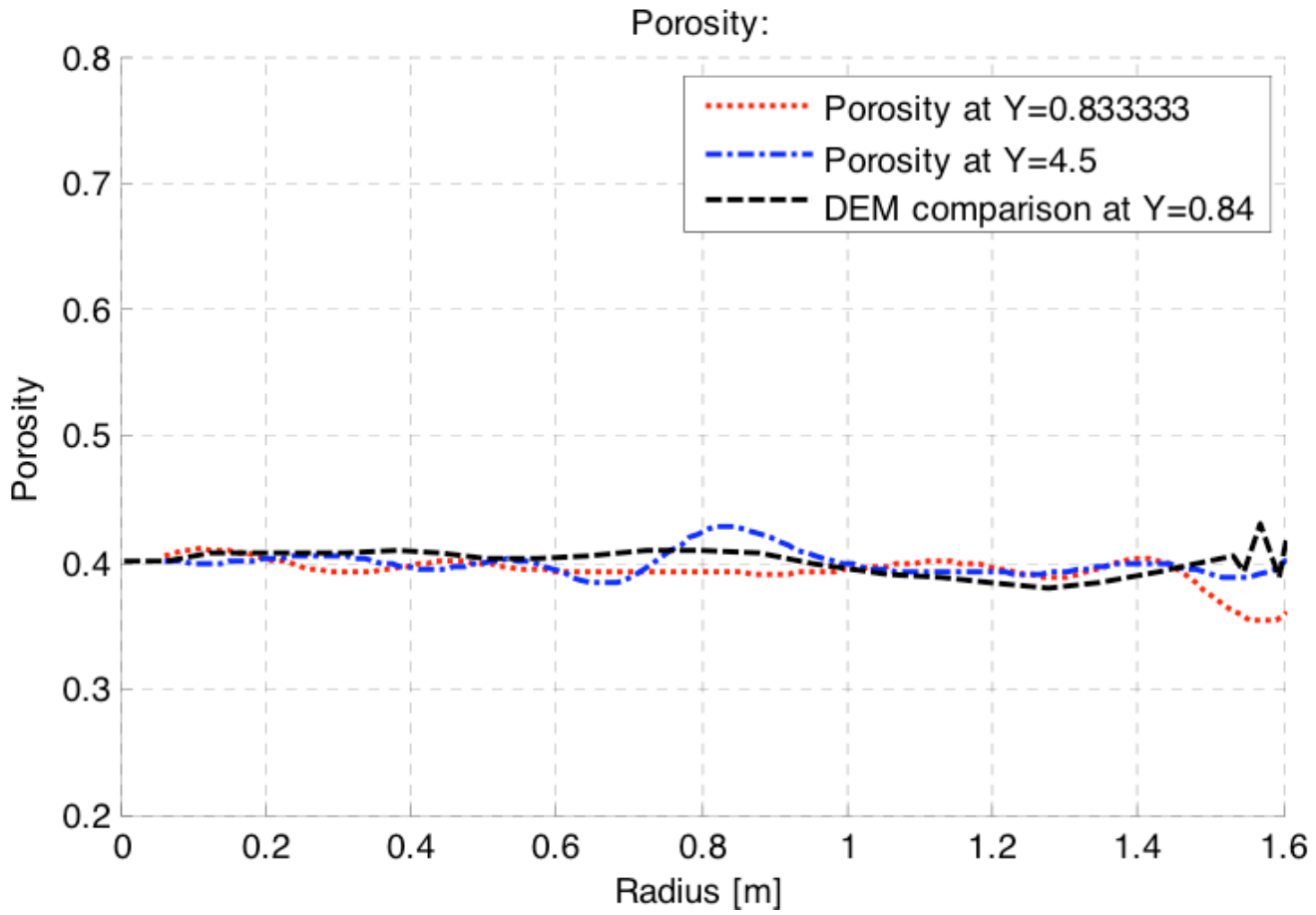
- N non symmetric, but positive semidefinite.
- Parallelizing the algorithms: block Jacobi with Gauss Seidel blocks, or coloring GS (50% there).
- Huge scale simulation. Multigrid for rigid multibody dynamics?
- Including a good collision model– here we are at a loss with rigid body theory – may need some measure of deformability – convolution complementarity.
- Involving other physics … fluid flow.
- Compare with experimental data.

# Comparison with experimental data PBR



# Packing statistics



# *Conclusions*

- We have defined a new algorithm for complementarity problems with conic constraints.
- We have shown that it can solve  very large problems in granular flow far faster than DEM.
- It is the first iterative algorithm that provably converges for nonsmooth rigid body dynamics.
- Its scalability is decent.
- We have created a multithreaded implementation and GPU port increases computational speed by a factor comparable to the number of stream processors.