Argonne NATIONAL LABORATORY

... for a brighter future

U.S. Department of Energy

THE UNIVERSITY OF CHICAGO

Office of Science U.S. DEPARTMENT OF ENERGY

A U.S. Department of Energy laboratory managed by The University of Chicago

An iterative solver for cone complementarity problems of nonsmooth multibody dynamics

Mihai ANITESCU [1],

Alessandro TASORA[2],

[1] Argonne National Laboratories

[2] Università degli Studi di Parma

2nd International Conference in Continuous Optimization

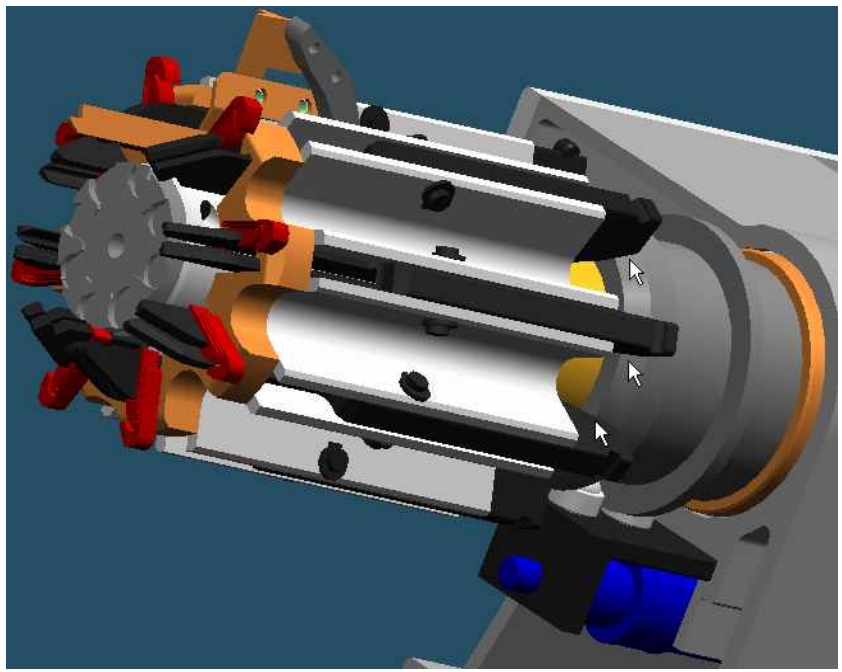
Introduction

- Unilateral constraints, friction, impacts are frequent in mechanisms
- They result in discontinuities in velocity → non-smooth dynamics

Traditional ODE / DAE solvers require **smoothness**!

Example: packaging device

Example: radial gripper with 3D cams

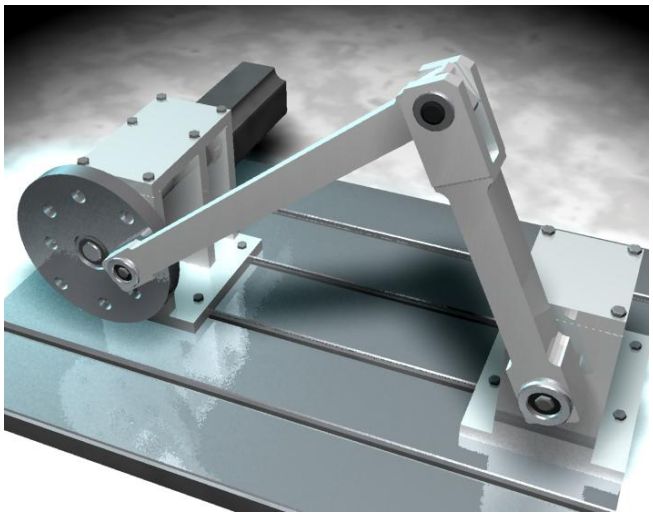


Introduction

- Multi rigid body systems, n degrees of freedom

Only bilateral constraints:

DAE / ODE:
Solve for unknown accelerations at each time step using **linear systems**



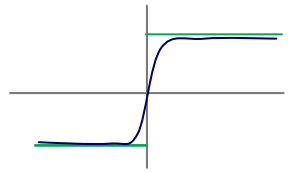
$O(n^3)$ computational complexity [Gauss]

$O(n)$ computational complexity with recent methods [A.Tasora, D.Baraff]

Introduction

.. Adding also non-smooth constraints (es.friction):

NO



ODE or DAE + regularization methods
(trick: approximate with stiff force fields)

Stiffness: too small time steps!

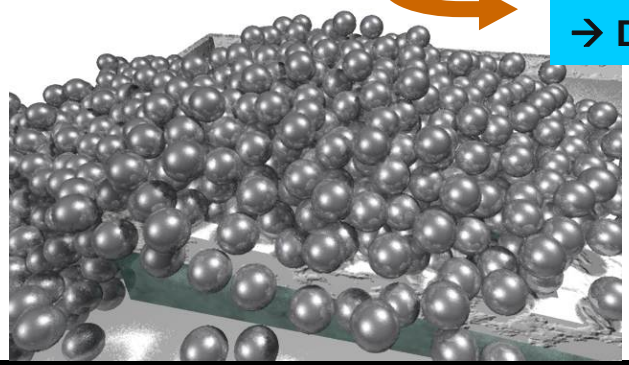
ODE or DAE + "stop-and-restart" methods NO

Impracticable for complex systems!

"Time Stepping Methods" ,
"Vector Measure Differential Inclusions"

OK

→ **DCP**: **D**ifferential-**C**omplementarity **P**roblem



Handle large steps, multiple discontinuities

Embed complementarity problem, each step

How to solve complementarity problems?

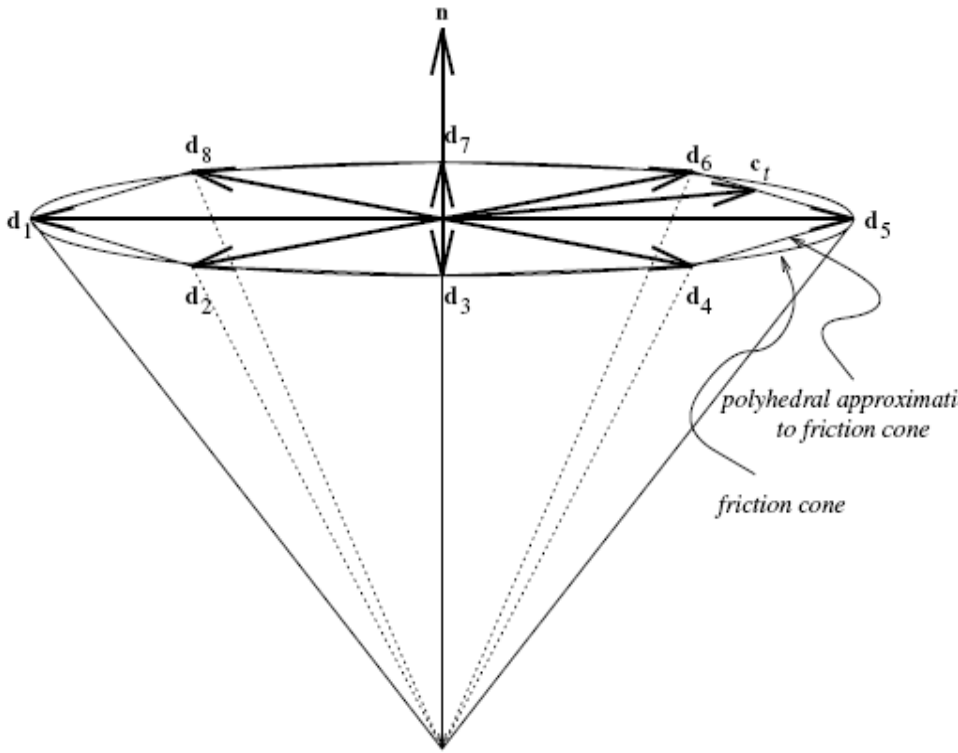
Goal

- Develop a solver for complementarity problems in time stepping methods with full cone formulation
- Must be able to handle thousands or millions of constraints
- Iterative solver, matrix-less, linear-time, linear-space
- Robust: must handle ill-posed and redundant constraints
- Efficient, fast, real-time if possible

Goal

- Why not use off-the-shelf LCP (Linear Complementarity) solvers (PATH)?
- Most LCP solvers (Lemke, Dantzig) are based on exact simplex methods with NP-hard complexity class. Risk of combinatorial explosion!
- *We prefer an approximate O(n) iterativem method*
- LCP require finitely-generated approximations of NL convex sets, see the friction cones:
- *We want no polyhedral approximations, within a matrix-free method for very large scal ecomputation – not available*

Need for a custom iterative **non-linear complementarity solver**



Example (D.Stewart): Polyhedral approximation of friction cones, to feed into typical LCP solve

Nonsmooth dynamics

- Contact, dynamics, friction for rigid bodies. Applicable to granular media, structural analysis, robotics …
- Differential problem with equilibrium constraints – DPEC.

$$
\begin{aligned}
M\frac{dv}{dt} &= \sum_{j=1,2,..,p}\left(c_n^{(j)}n^{(j)}+\beta_1^{(j)}t_1^{(j)}+\beta_2^{(j)}t_2^{(j)}\right)+f_c(q,v)+k(t,q,v)\\
\frac{dq}{dt} &= v\\
c_n^{(j)} &\geq 0 \quad \perp \quad \Phi^{(j)}(q)\geq 0,\ j=1,2,\ldots,p\\
\left(\beta_1^{(j)},\beta_2^{(j)}\right) &= \operatorname{argmin}_{\mu^{(j)}c_n^{(j)}\geq\sqrt{\left(\beta_1^{(j)}+\beta_2^{(j)}\right)^2}} \qquad j=1,2,\ldots,p\\
&\left[\left(v^Tt_1^{(j)}\right)t_1^{(j)}+\left(v^Tt_2^{(j)}\right)t_2^{(j)}\right]^T\left(\beta_1t_1^{(j)}+\beta_2t_2^{(j)}\right).
\end{aligned}
$$

  - May not have a continuous velocity solution even without impact (Painleve, 1896) – need to go to weaker forms- measure differential inclusions.

Model

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

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

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

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

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

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

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

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

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

Speeds

Reaction impulses

Forces

Stabilization terms

Bilateral constraint equations

Contact constraint equations

COMPLEMENTARITY!

Coulomb 3D friction model

Convex Relaxation

- A modification (relaxation, to get a convex problem):

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

(For small μ and/or small speeds, almost no differences from the Coulomb theory)

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

- 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

- Here we introduced the convex cone

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

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

The iterative method

- How to practically solve the Cone Complementarity Problem?

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

- 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 pr operator onto feasible set

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

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

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

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 $(x^{r+1} - x^r)^T \left((\lambda \omega B^r)^{-1} + K^r - \frac{N}{2} \right) (x^{r+1} - x^r) \geq \beta \left\| x^{r+1} - x^r \right\|^2$.

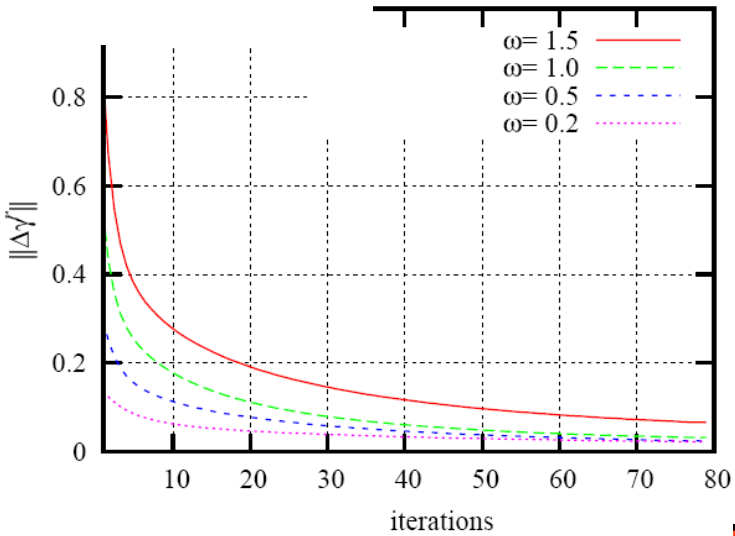
Always satisfied in multibody systems

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.



Theory

$$(OC)\quad \begin{array}{ll} \min & f(x) = \frac{1}{2}x^T N x + r^T x \\ s.t. & x_i \in \Upsilon^i, \qquad\qquad\qquad\qquad i = 1, 2, \ldots, n_k. \end{array}$$

Theorem Assume that $x^0 \in \Upsilon$ and that the sequences of matrices B^r and K^r are bounded. Then we have that

$$f(x^{r+1}) - f(x^r) \leq -\beta \left\| x^{r+1} - x^r \right\|^2$$

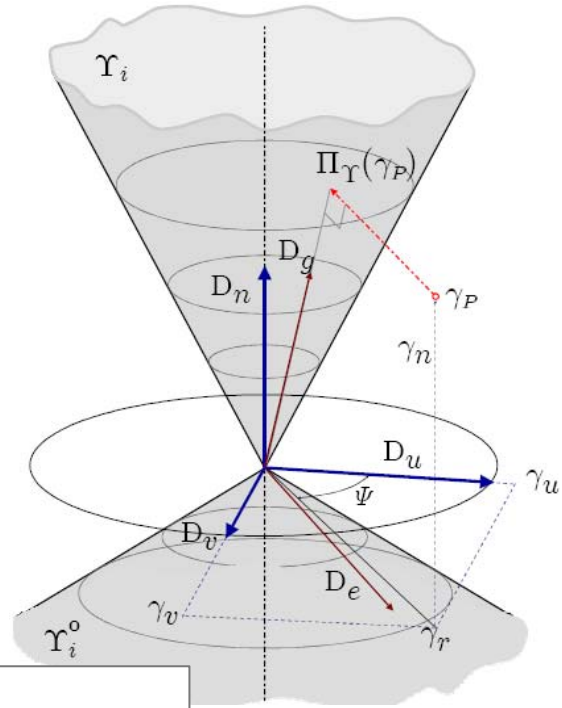
for any iteration index r, and any accumulation point of the sequence x^r is a solution of (CCP).

Corollary Assume that the friction cone of the configuration is pointed (that is, there does not exist a choice of reaction forces whose net effect is zero). If the relevant parameters satisfy assumptions A2 and A3, then the algorithm produces a bounded sequence, and any accumulation point results in the same velocity solution

The projection operator

- *For each frictional contact constraint:*

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



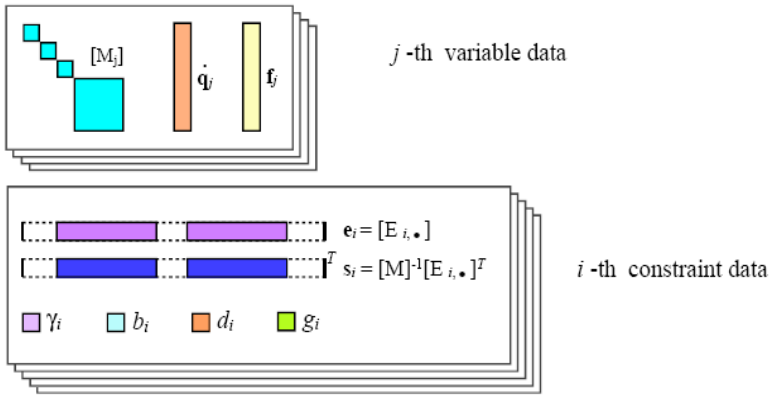
- *For each bilateral constraint, simply do nothing.*
- The complete operator:

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

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

The algorithm

- Development of an efficient algorithm for fixed point iteration:



- *avoid temporary data, exploit sparsity. Never compute explicitly the N matrix!*
- *implemented in incremental form. Compute only deltas of multipliers.*
- *O(n) space requirements and supports premature termination*
- *for real-time purposes: O(n) time*

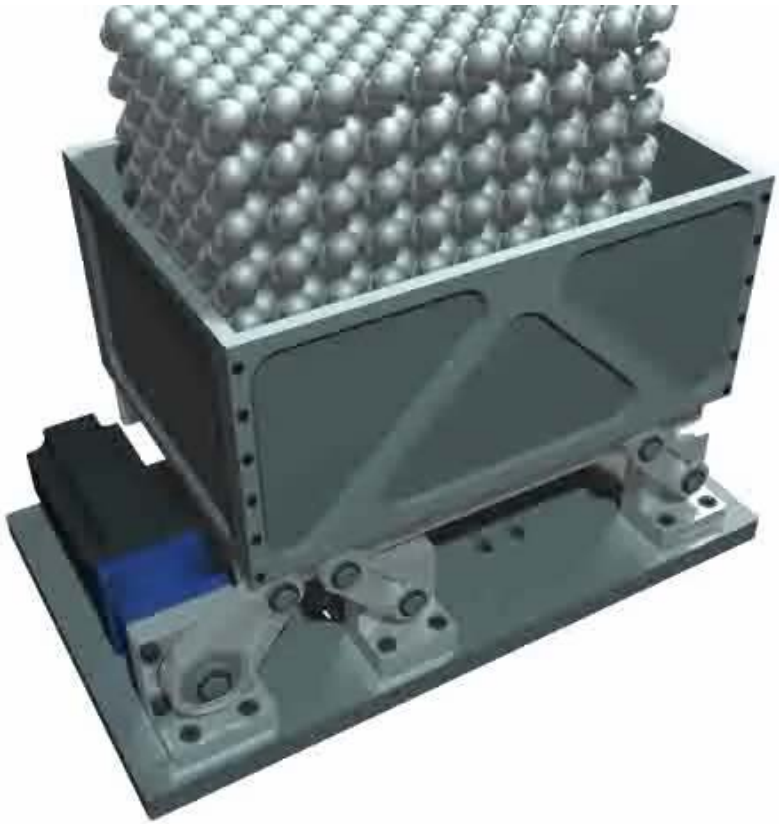
The algorithm

(1) *// Pre-compute some data for friction constraints*
(2) **for** $i := 1$ **to** $n_{\mathcal{A}}$
(3) $\quad \boldsymbol{s}_a^i = M^{-1} D^i$
(4) $\quad g_a^i = D^{i,T} \boldsymbol{s}_a^i$
(5) $\quad \eta_a^i = \frac{3}{\text{Trace}(g_a^i)}$
(6) *// Pre-compute some data for bilateral constraints*
(7) **for** $i := 1$ **to** $n_{\mathcal{B}}$
(8) $\quad s_b^i = M^{-1} \boldsymbol{\nabla \Psi}^i$
(9) $\quad g_b^i = \boldsymbol{\nabla \Psi}^{i,T} s_b^i$
(10) $\quad \eta_b^i = \frac{1}{g_b^i}$
(11)
(12) *// Initialize impulses*
(13) **if** warm start with initial guess $\boldsymbol{\gamma}_{\mathcal{E}}^*$
(14) $\quad \boldsymbol{\gamma}_{\mathcal{E}}^0 = \boldsymbol{\gamma}_{\mathcal{E}}^*$
(15) **else**
(16) $\quad \boldsymbol{\gamma}_{\mathcal{E}}^0 = 0$
(17)
(18) *// Initialize speeds*
(19) $\boldsymbol{v} = \sum_{i=1}^{n_{\mathcal{A}}} \boldsymbol{s}_a^i \boldsymbol{\gamma}_a^{i,0} + \sum_{i=1}^{n_{\mathcal{B}}} s_b^i \gamma_b^{i,0} + M^{-1} \tilde{\boldsymbol{k}}$

(21) *// Main iteration loop*
(22) **for** $r := 0$ **to** r_{max}
(23) $\quad$ *// Loop on frictional constraints*
(24) $\quad$ **for** $i := 1$ **to** $n_{\mathcal{A}}$
(25) $\quad\quad \boldsymbol{\delta}_a^{i,r} = \left(\boldsymbol{\gamma}_a^{i,r} - \omega \eta_a^i \left(D^{i,T} \boldsymbol{v}^r + \boldsymbol{b}_a^i \right) \right);$
(26) $\quad\quad \boldsymbol{\gamma}_a^{i,r+1} = \lambda \Pi_{\Upsilon} \left(\boldsymbol{\delta}_a^{i,r} \right) + (1 - \lambda) \boldsymbol{\gamma}_a^{i,r} \ ;$
(27) $\quad\quad \Delta \boldsymbol{\gamma}_a^{i,r+1} = \boldsymbol{\gamma}_a^{i,r+1} - \boldsymbol{\gamma}_a^{i,r} \ ;$
(28) $\quad\quad \boldsymbol{v} := \boldsymbol{v} + \boldsymbol{s}_a^{i^T} \Delta \boldsymbol{\gamma}_a^{i,r+1}.$
(29) $\quad$ *// Loop on bilateral constraints*
(30) $\quad$ **for** $i := 1$ **to** $n_{\mathcal{B}}$
(31) $\quad\quad \delta_b^{i,r} = \left(\gamma_b^{i,r} - \omega \eta_b^i \left(\nabla \boldsymbol{\Psi}^{i,T} \boldsymbol{v}^r + b_b^i \right) \right);$
(32) $\quad\quad \gamma_b^{i,r+1} = \lambda \Pi_{\Upsilon} \left(\delta_b^{i,r} \right) + (1 - \lambda) \gamma_b^{i,r} \ ;$
(33) $\quad\quad \Delta \gamma_b^{i,r+1} = \gamma_b^{i,r+1} - \gamma_b^{i,r} \ ;$
(34) $\quad\quad \boldsymbol{v} := \boldsymbol{v} + s_b^{i^T} \Delta \gamma_b^{i,r+1}.$
(35)
(36) **return** $\boldsymbol{\gamma}_{\mathcal{E}}$, $\boldsymbol{v}$

Examples

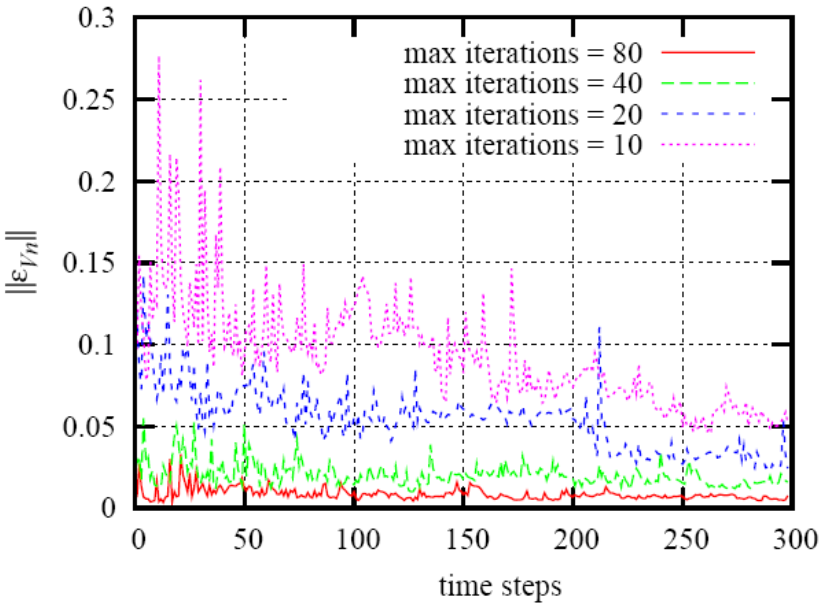
- Example: size-segregation in shaker, with thousands of steel spheres



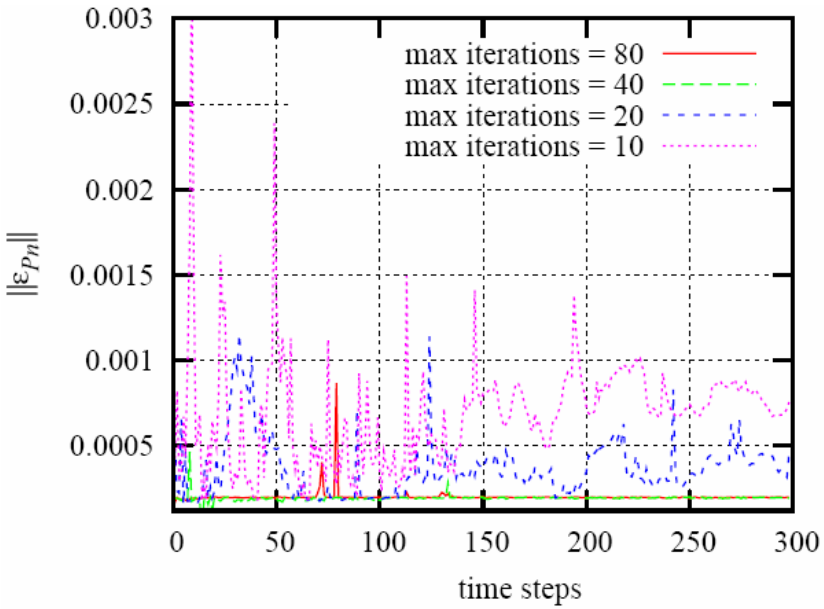
Note: solution beyond reach of Lemke-type LCP solvers!

Tests

- Feasibility accuracy increases with number of iterations:



Speed violation in constraints

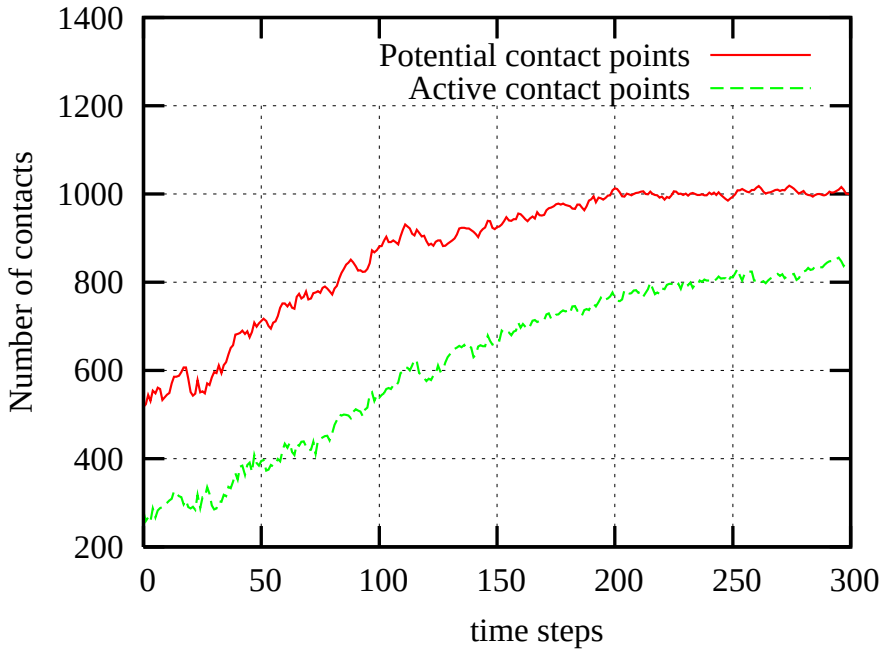


Position error in constraints (penetration)

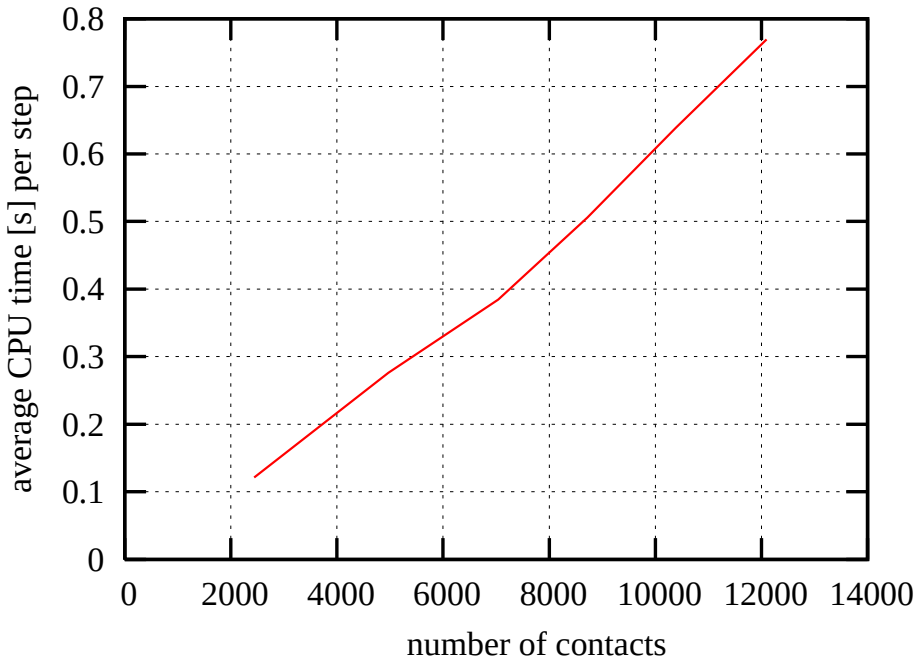
(with example of 300 spheres in shaker)

Tests: Scalability

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



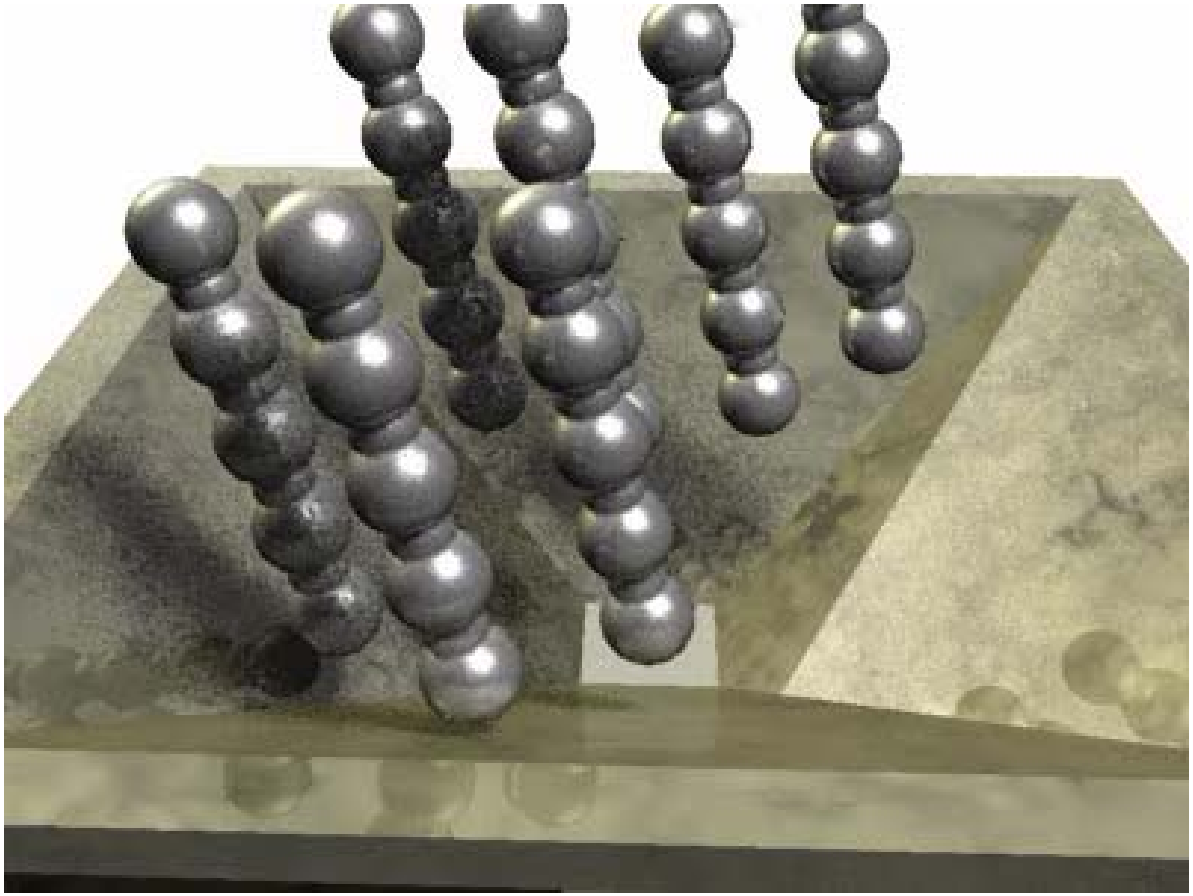
Number of contacts in time, 300 spheres



CPU time per step for 300-1500 spheres

Examples

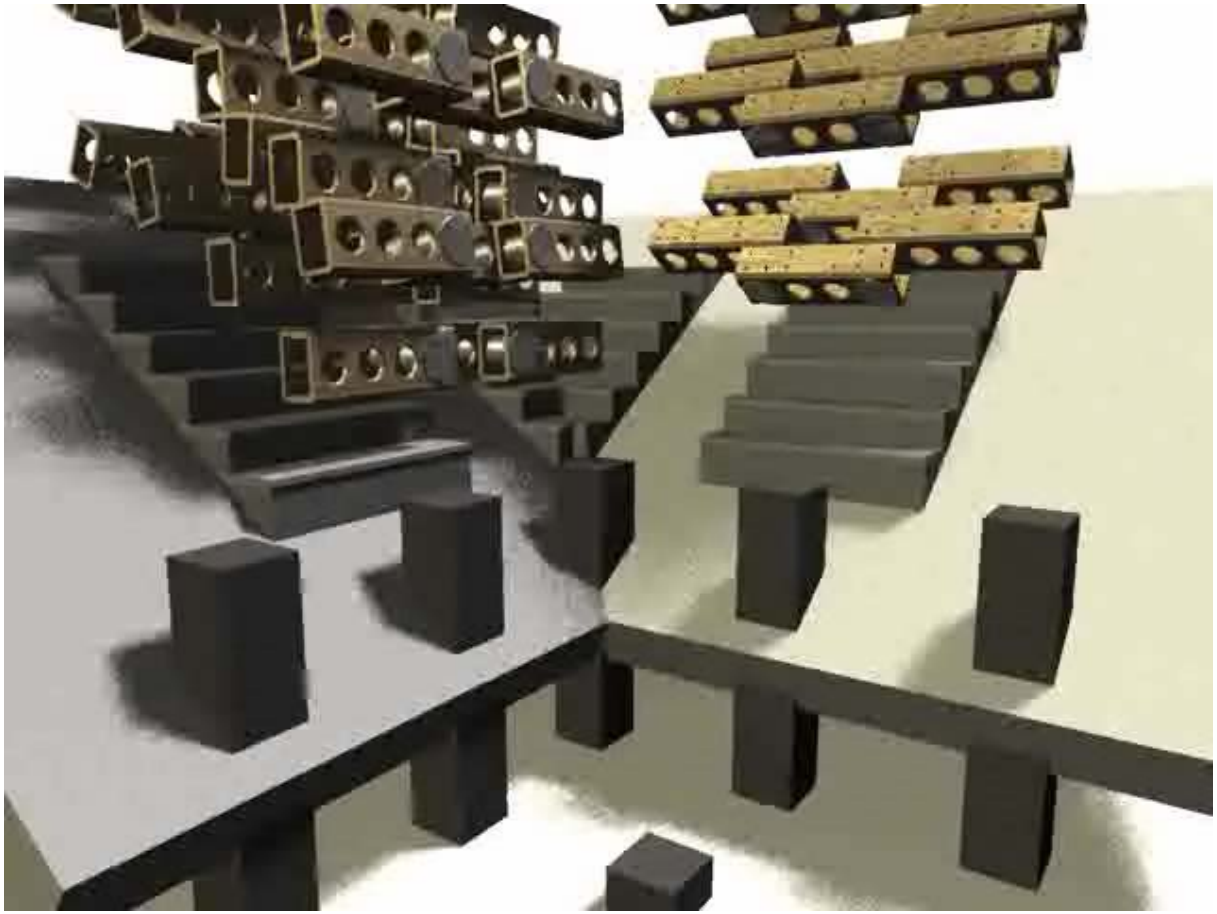
- Benchmark: contacts between articulated objects (contacts + bilateral joints)



Note: real-time simulation

Examples

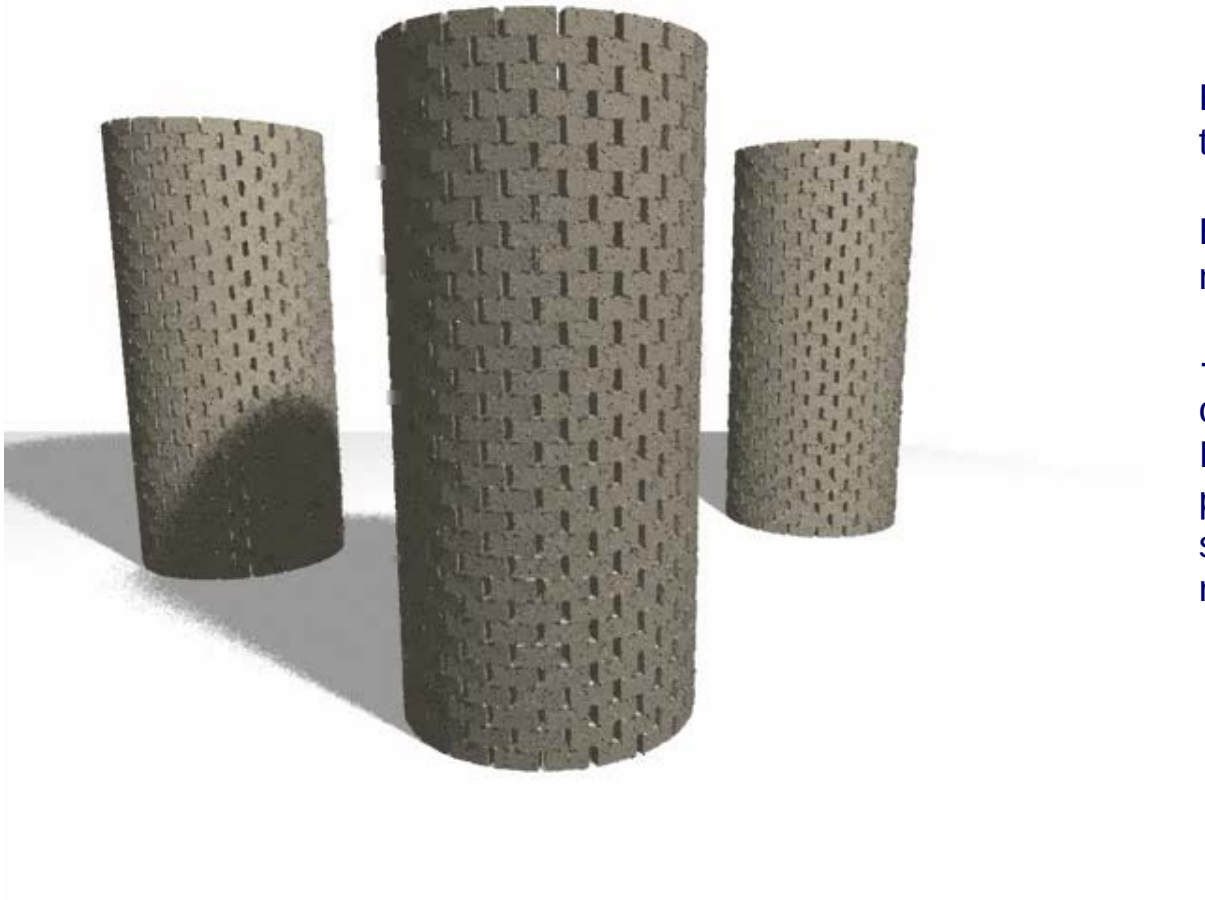
- Benchmark: contacts between non-convex shapes



Note: real-time simulation

Examples

- Benchmark: contacts between many polytopes and high friction coeff.



High stacks of objects: a typical difficult benchmark.

Even smallest CCP errors may compromise stability.

→ Many iterations needed, otherwise unstable and falls. Indicates need for preconditioning or warm starting; subject of future research.

100 iterations: STABLE

Few iterations: UNSTABLE

Examples

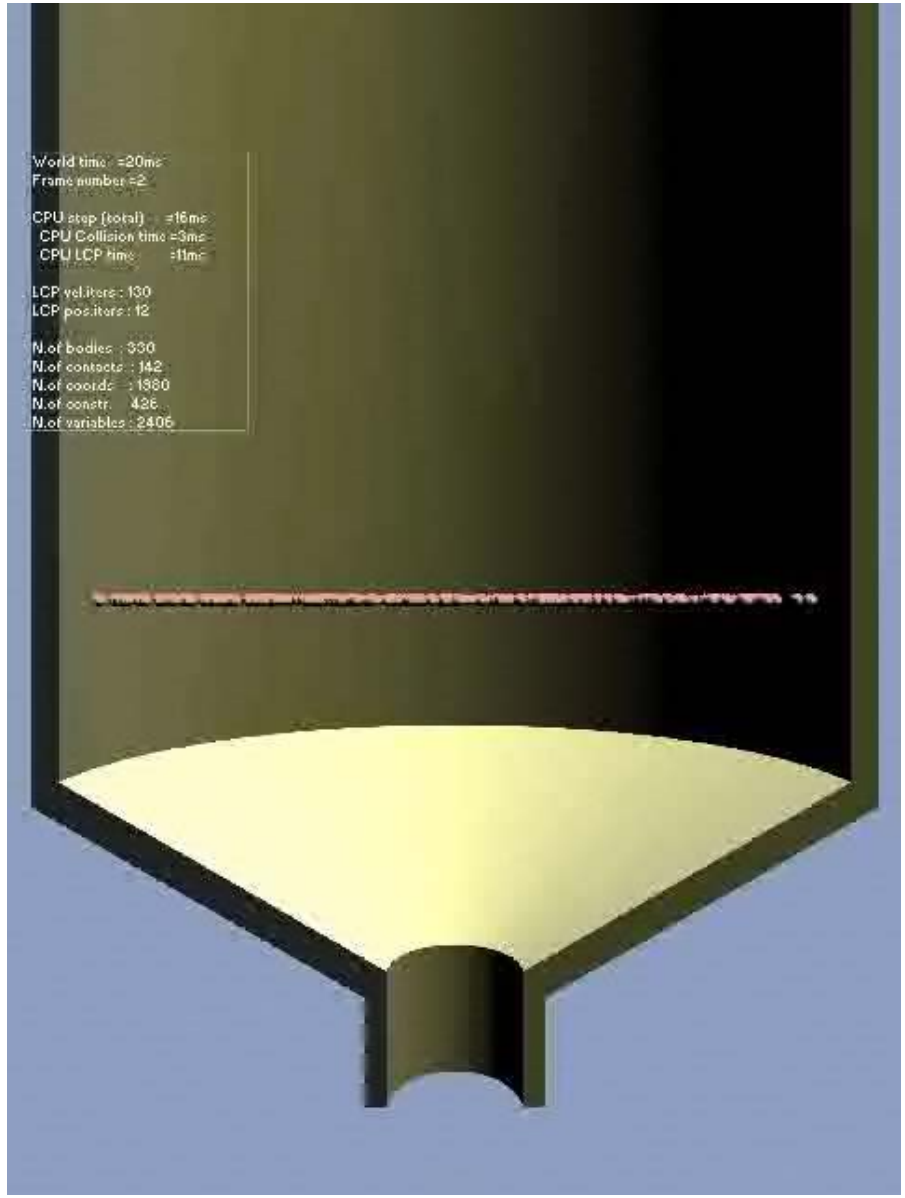
- Refueling in a PBR nuclear reactor

- Up to 300’000 uranium-graphite pebbles

- Dense bidisperse granular flow problem: hard to simulate

- Our method requires hours on 1CPU where other approach (DEM) requires weeks on 64 .
- Currently, we are doing 30,000 pebbles in about 2 hrs CPU on windows Laptop, maximum 120,000 contacts and +500,000 impulse variables.

For details on PBR reactors, see Bazant et al. (MIT)



Some open problems

- Slow convergence for high mass ratios (many iterations needed for reasonable precisions).
- How to introduce a more advanced friction model?
  Ex: different static and kinematic friction coefficients, etc.
  Note: anisotropic friction would be easy.
- How to precondition (to make the number of iterations constant for same penetration error and vastly increasing number of bodies?) though warm starting may contribute a lot here.
- Collisions with restitution: not discussed here..

Future work

- Optimization of C++ source, aiming at simulations with 500’000 bodies on a single PC.
- Implementation on parallel machines. Multithreading.
- Persistent contact manifold, for warm starting the method
- Recent GP-GPU processors should be exploited for massive low-cost parallelism (SIMD stream-kernel architecture)

Conclusions

- Approach based on Time Stepping Methods
- Solution by means of fixed point iteration, with no polygonal cone approximation.
- Fast, robust, matrix-less iterative scheme
- Fits well in real-time scenarios, since can be stopped early with a good solution.
- Tested with more than 500’000 constraint multipliers, scales well.
- Implemented in a C++ multibody simulation middleware Chrono::Engine (developer by Tasora) `(http://www.deltaknowledge.com/chronoengine)`

Introduction

- Example: a walking robot, simulated using our approach